Highlights

A Comprehensive Survey of Federated Intrusion Detection Systems: Techniques, Challenges and Solutions

Ioannis Makris,Aikaterini Karampasi,Panagiotis Radoglou-Grammatikis,Nikolaos Episkopos,Eider Iturbe,Erkuden Rios,Nikos Piperigkos,Aris Lalos,Christos Xenakis,Thomas Lagkas,Vasileios Argyriou,Panagiotis Sarigiannidis

- **Comprehensive Analysis of Federated Intrusion Detection Systems**: A holistic analysis of Intrusion Detection Systems (IDS) leveraging Federated Learning (FL) is carried out. It is worth mentioning that the paper discusses Federated IDS (FIDS) used in different domains, such as the Industrial Internet of Things (IIoT) and the Internet of Vehicles (IoV), demonstrating the versatility and importance of FL in enhancing cybersecurity across diverse domains.
- **Analysis of Aggregation Methods and Challenges**: Different aggregation methods are analysed, paying special attention to the challenges associated with each method, including data privacy, model robustness and efficiency.
- **Trends and Research Directions**: Based on the previous analysis, particular research directions for future work in the context of FL-driven IDS are provided.

A Comprehensive Survey of Federated Intrusion Detection Systems: Techniques, Challenges and Solutions[⋆,⋆⋆]

Ioannis Makris[a,1], Aikaterini Karampasi[b], Panagiotis Radoglou-Grammatikis[b,*], Nikolaos Episkopos[a], Eider Iturbe[c], Erkuden Rios[c], Nikos Piperigkos[d], Aris Lalos[d], Christos Xenakis[e], Thomas Lagkas[f], Vasileios Argyriou[g] and Panagiotis Sarigiannidis[b]

[a]MetaMind Innovations P.C., Kila, Kozani, 50100, Greece
[b]Department of Electrical and Computer Engineering, University of Western Macedonia, Campus ZEP Kozani, Kozani, 50100, Greece
[c]TECNALIA, Basque Research and Technology Alliance (BRTA), Parque Científico y Tecnológico de Gipuzkoa, Mikeletegi Pasealekua, 2, 20009 Donostia, SS, Derio, 48160, Spain
[d]Industrial Systems Institute / Research Center "ATHENA", Patras Science Park building Platani, Patras, 26504, Greece
[e]Secure Systems Laboratory, Department of Digital Systems, University of Piraeus, 80 Karaoli & Dimitriou, Piraeus, 18534, Greece
[f]Department of Computer Science, Democritus University of Thrace, Kavala Campus, Kavala, 65404, Greece
[g]Department of Networks and Digital Media, Kingston University London, Penrhyn Road, Kingston upon Thames, Surrey, London, KT1 2EE, UK

ARTICLE INFO

Keywords:
Cybersecurity, Federated Learning, Intrusion Detection, Intrusion Prevention

ABSTRACT

Cyberattacks have increased radically over the last years, while the exploitation of Artificial Intelligence (AI) leads to the implementation of even smarter attacks which subsequently require solutions that will efficiently confront them. This need is indulged by incorporating Federated Intrusion Detection Systems (FIDS), which have been widely employed in multiple scenarios involving communication in cyber-physical systems. These include, but are not limited to, the Internet of Things (IoT) devices, Industrial IoT (IIoT), healthcare systems (Internet of Medical Things / IoMT), Internet of Vehicles (IoV), Smart Manufacturing (SM), Supervisory Control and Data Acquisition (SCADA) systems, Multi-access Edge Computing (MEC) devices, among others. Tackling the challenge of cyberthreats in all the aforementioned scenarios is of utmost importance for assuring the safety and continuous functionality of the operations, crucial for maintaining proper procedures in all Critical Infrastructures (CIs). For this purpose, pertinent knowledge of the current status in state-of-the-art (SOTA) federated intrusion detection methods is mandatory, towards encompassing while simultaneously evolving them in order to timely detect and mitigate cyberattack incidents. In this study, we address this challenge and provide the readers with an overview of FL implementations regarding Intrusion Detection in several CIs. Additionally, the distinct communication protocols, attack types and datasets utilized are thoroughly discussed. Finally, the latest Machine Learning (ML) and Deep Learning (DL) frameworks and libraries to implement such methods are also provided.

1. Introduction

In the digital era of the Internet of Things (IoT) and Artificial Intelligence (AI), smart ecosystems have emerged as a cornerstone of innovation, thus seamlessly integrating advanced technologies into the fabric of daily life and business operations. Their interconnected nature and reliance on vast data exchanges have revolutionised industries, offering unprecedented efficiency and multiple benefits. However, this digital transformation opens up a Pandora's box of security issues that may lead to catastrophic effects. In particular, the integration of smart technologies in Critical Infrastructures (CIs), introduces the risk of cyber-physical attacks capable of causing substantial damages beyond the digital realm. Moreover, the proliferation of smart devices can increase the risk of both typical single-step cyberattacks and multi-step cyberattacks. This is because their functions rely on insecure communication protocols and heterogeneous technologies that may be characterised by potential vulnerabilities. For

⋆This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101070450 (AI4CYBER). Disclaimer: Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the European Commission can be held responsible for them.

makris@metamind.gr (I. Makris); a.karampasi@uowm.gr (A. Karampasi); pradoglou@uowm.gr (P. Radoglou-Grammatikis); nepisko@gmail.com (N. Episkopos); Eider.Iturbe@tecnalia.com (E. Iturbe); Erkuden.Rios@tecnalia.com (E. Rios); piperigkos@ceid.upatras.gr (N. Piperigkos); lalos@isi.gr (A. Lalos); xenakis@unipi.gr (C. Xenakis); tlagkas@cs.duth.gr (T. Lagkas); vasileios.argyriou@kingston.ac.uk (V. Argyriou); psarigiannidis@uowm.gr (P. Sarigiannidis)

https://metamind.gr/, makris@metamind.gr (I. Makris); https://ithaca.ece.uowm.gr/, a.karampasi@uowm.gr (A. Karampasi); ttps://ithaca.ece.uowm.gr/, pradoglou@uowm.gr (P. Radoglou-Grammatikis); https://metamind.gr/, nepisko@gmail.com (N. Episkopos); https://www.tecnalia.com/, Eider.Iturbe@tecnalia.com (E. Iturbe); https://www.tecnalia.com/, Erkuden.Rios@tecnalia.com (E. Rios); https://www.isi.gr/, piperigkos@ceid.upatras.gr (N. Piperigkos); https://www.isi.gr/, lalos@isi.gr (A. Lalos); https://www.unipi.gr/en/xenakis-2/, xenakis@unipi.gr (C. Xenakis); https://cs.duth.gr/, tlagkas@cs.duth.gr (T. Lagkas); https://www.kingston.ac.uk/staff/profile/professor-vasilis-argyriou-332/, vasileios.argyriou@kingston.ac.uk (V. Argyriou); https://ithaca.ece.uowm.gr/, psarigiannidis@uowm.gr (P. Sarigiannidis)

ORCID(s): 0000-0002-6574-8525 (I. Makris); 0000-0001-5993-3233 (A. Karampasi); 0000-0003-1605-9413 (P. Radoglou-Grammatikis); 0009-0004-7130-3874 (N. Episkopos); 0000-0002-5458-6049 (E. Iturbe); 0000-0001-5541-1091 (E. Rios); 0000-0003-0262-7619 (N. Piperigkos); 0000-0003-0511-9302 (A. Lalos); 0000-0001-6718-122X (C. Xenakis); 0000-0002-0749-9794 (T. Lagkas); 0000-0003-4679-8049 (V. Argyriou); 0000-0001-6042-0355 (P. Sarigiannidis)

instance, several industrial communication protocols, such as Modbus/Transmission Control Protocol (TCP), Distribution Network Protocol 3 (DNP3) and IEC 61850, are prone to unauthorised attacks. On the other hand, it is worth mentioning that the autonomous and automatic ability of smart devices to communicate with each other and with external environments may raise security and privacy concerns. Finally, cyber attacks are continuously evolving by leveraging Machine Learning (ML) and Deep Learning (DL) methods to automate and optimise malicious activities, making them more efficient and difficult to predict or counteract. In contrast to conventional cyberattacks that rely on predefined scripts and methods, ML & DL approaches are able to adjust in real-time and imitate human behaviour. Therefore, in light of the aforementioned remarks, the need for robust security measures that can ensure confidentiality, integrity and availability principles is evident.

The role of Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) is indispensable for safeguarding interconnected networks against a myriad of cyberattacks. Such systems can effectively monitor various kinds of data, such as network traffic, system logs and operational data, in order to recognise suspicious activities and potential security breaches in real-time. For this purpose, they can leverage both signature-based and anomaly-detection methods. On the one hand, signature-based detection relies on predefined patterns or signatures that characterize particular malicious activities. On the other hand, anomaly-based detection leverages statistical analysis and AI techniques in order to detect and discriminate malicious actions. While the use of AI for intrusion detection bears significant advantages compared to traditional methods, even though these two approaches act in a complementary manner, it also raises several concerns. First, AI requires a vast amount of data for training, thus creating privacy concerns, especially when dealing with sensitive information. Moreover, usually, this kind of data is not available and differs from environment to environment. In addition, given the evolving nature of cyberattacks, the AI models should continuously be updated and re-trained towards recognising new attack patterns. Finally, the use of AI in the context of critical operations creates ethical considerations. Hence, despite the benefits of AI, specific countermeasures are required in order to address the aforementioned challenges.

Federated Learning (FL) is an ML approach that allows multiple entities and environments to collaboratively learn a common AI model while keeping all the training data localised. FL can resolve various concerns about the utilisation of AI in cybersecurity, especially in the context of intrusion detection. In particular, FL may unravel those challenges that are related to data privacy and model robustness. On the one hand, FL enables decentralised training without sharing sensitive data with a centralised server or third parties (Figure 1). Thus, given that the data remains within local entities and environments, the risk of data breaches is reduced. Additionally, FL enhances the robustness and resilience of AI models since they leverage multiple data sources. This diversity results in generalised models that can encounter different attack vectors. Based on the aforementioned remarks, this paper aims to investigate the impact of FL on intrusion detection and prevention mechanisms. Therefore, based on the aforementioned remarks, the contributions of this paper are summarised as follows:

- **Comprehensive Analysis of Federated Intrusion Detection Systems**: A holistic analysis of Intrusion Detection Systems (IDS) leveraging Federated Learning (FL) is carried out. It is worth mentioning that the paper discusses Federated IDS (FIDS) used in different domains, such as the Industrial Internet of Things (IIoT) and the Internet of Vehicles (IoV), demonstrating the versatility and importance of FL in enhancing cybersecurity across diverse domains.
- **Analysis of Aggregation Methods and Challenges**: Different aggregation methods are analysed, paying special attention to the challenges associated with each method, including data privacy, model robustness and efficiency.
- **Trends and Research Directions**: Based on the previous analysis, particular research directions for future work in the context of FL-driven IDS are provided.

The rest of this paper is structured as follows. Section 2 discusses similar survey papers and summarises the differences and contributions of this work. Section 3 provides an overview of intrusion detection and prevention. Similarly, a background on FL is provided in section 4. Next, section 5 provides a comprehensive analysis of FIDS. Subsequently, based on this analysis, lessons learned and directions for future work in this research are provided in section 6. Finally, section 7 concludes this paper.

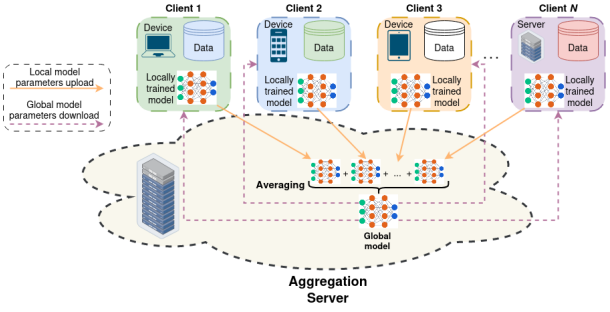

Figure 1: Federated Learning overview Shaheen et al. (2022).

2. Motivation, Relevant Work and Contributions

Federated Intrusion Detection Systems (FIDS) have been extensively studied to enhance the gained knowledge regarding the safety of interconnected systems. The complexity that is present in various environments, concerning the multiple

end-devices which must communicate with each other or with a main entity/server poses new challenges in effectively addressing the highly trained attacks that are continuously created. More specifically, in the digital era where the endless battle between attackers and defenders evolves, the latter should always be ahead in order to preserve the smooth operation of the systems.

Towards fortifying the entities involved against cyberthreats, a variety of surveys have been provided to the research community. More precisely, the authors in Ali, Li and Yousafzai (2024) elaborate on the usage of blockchain, as well as FL on Industrial Internet of Things (IIoT) in extensive research regarding IDS and IPS on such an ecosystem. In Alsamiri and Khalid (2023) the authors provide us with a comprehensive survey for identifying the applications of FL for IDS and IPS in the Internet of Vehicles (IoV) environments. Moreover, in Mourad, Otrok and Guizani (2023) the authors make an effort of not only identifying IDS and IPS systems for a variety of environments, yet they are additionally proposing a cybersecurity framework encompassing explainability of the techniques implemented against cyberattacks. In Girdhar, Singh and Kumar (2023) the authors elaborated on comparing various attack detection techniques focusing on AI and blockchain techniques, while identifying any limitations and future proposals regarding cybersecurity issues. Additionally, in Lavaur, Pahl, Busnel and Autrel (2022) the authors try to determine FL-based IDS systems from the creation of FL in 2016 up to 2021.

Having said this, we intend to provide a comprehensive survey that investigates FIDS in terms of data types and sources, FL aggregation techniques, attacks under consideration, detection performance, technologies and datasets. More notably, the importance of this study lies in the aggregation of the relevant literature, while incorporating various circumstances, and are, additionally, evaluated on similar datasets making them comparable in terms of performance in distinct scenarios, as they are posed in the current State of the Art (SOTA) techniques.

3. Overview of Intrusion Detection and Prevention

In recent years, it is obvious that even though smart technologies offer many advantages, the corresponding services and applications are vulnerable to a significant number of intrusions and cyberattacks. With the main goal being the prevention of such attacks in a timely manner, unexpected security events and zero-day vulnerabilities make this intention unrealistic. However, the prompt detection of cyberthreats and anomalies, without the need to affect legitimate services, can be treated as a realistic solution. Therefore, the development and implementation of IDSs is necessary. To this end, an overview of Intrusion Detection and Prevention Systems (IDPSs) is given in this section. According to the Request For Comments (RFC) 2828 (Internet Security Glossary) Shirey (2000), intrusion detection is related to regularly checking, evaluating, and monitoring security-related events aiming to immediately detect any malicious behaviour or anomaly in the system. In 1978, D. Denning built and defined the first solid intrusion detection model, Denning (1987). Based on this model, numerous engineers started building and designing the first IDSs. In 1980, the term "IDS" was coined, and it was directly connected to a hardware and/or software system that automatically executed the aforementioned activities. In that year, James Anderson discovered that log files could be used as an efficient approach to monitor the health of a computing system and the way in which individuals interact with it, Anderson (1980). The following section defines (a) the goals and specifications of IDPS, (b) an IDPS reference architecture, (c) intrusion detection methods and (d) intrusion prevention mechanisms.

3.1. Objectives and Requirements of Intrusion Detection and Prevention Systems

According to the `RFC 2828 (Internet Security Glossary)`, a system intrusion is defined as: "A security event or combination of multiple security events, that constitutes a security incident in which an intruder gains, or attempts to gain, access to a system (or system resources) without being given the authorisation to do so". On the other hand, intrusion detection is described as: "A security service that monitors and analyses system events for the purpose of finding and providing real-time or near real-time warning or attempts to access system resources in an unauthorised manner". Even though attackers aim to exploit new vulnerabilities and evade potential defence mechanisms, they follow a common attack methodology consisting of ten steps, as presented by MITRE ATT&CK: a) Initial access, b) Execution, c) Persistence, d) Privilege escalation, e) Defense evasion, f) Credential access, g) Discovery, h) Lateral movement, i) Collection and exfiltration, and j) Command and control Roy, Panaousis, Noakes, Laszka, Panda and Loukas (2023); Stallings, Brown, Bauer and Howard (2012). Regardless of the primary goal of IDPS, which is the instantaneous detection and mitigation of potential attacks, the constrained characteristics of IoT devices and applications have led to new requirements regarding their role in IoT environments Zarpelão, Miani, Kawakani and de Alvarenga (2017). According to P. Radoglou-Grammatikis et al. in Radoglou-Grammatikis and Sarigiannidis (2019), these requirements are illustrated below:

- **Detection of various cyber threats and anomalies**: Based on previous studies, an IDPS should be capable of detecting and classifying a wide range of cyberattacks and anomalies, Heidari and Jabraeil Jamali (2022); Arisdakessian, Wahab, Mourad, Otrok and Guizani (2022); Yang, Liu, Li, Wu, Wang, Zhao and Han (2022); Yi, Bo, Ji, Saltzman, Jaehnig, Lei, Gao and Zhang (2023); Thakkar and Lohiya (2023).
- **Intrusion Detection in a Timely Manner**: Based on how critical each IoT application is, the corresponding cyberattacks and anomalies should be detected as close to real-time as possible.

- **High Detection Accuracy**: Achieving high accuracy is the most important challenge of an IDPS in terms of detecting several cyberattacks and anomalies.
- **Lightweight Resource Scaling**: Due to the constrained characteristics of IoT environments, an IDPS should be capable of operating in the best possible way, particularly in terms of high accuracy and punctual detection, without consuming lots of computing resources and influencing the core operation of the IoT devices and applications.
- **Scalability**: Based on the size of the IoT applications consisting of multiple IoT devices, a relevant IDPS should be capable of monitoring and controlling them efficiently.
- **Resiliency against Cyberattacks**: An IDPS should be capable of detecting and countering threats that aim at harming the system.
- **Friendly User Interface**: Similarly, due to the large amount of data and security events in IoT environments, the corresponding IDPS should be eligible for visualizing and finding the correlation of the different security events and alerts in a clear and understandable way.

3.2. Reference Architecture of Intrusion Detection and Prevention Systems (IDPS)

As illustrated in Figure 2, a classic IDPS consists of three main components, namely the Agent(s), the Analysis Engine, and the Response Module. The agents' responsibilities are monitoring the actions of the various entities and collecting as well as pre-processing the necessary data. It is important to note that depending on the position of the agent, an IDPS can be categorized as Host-based IDPS (HIDPS), Network-based IDPS (NIDPS) and Hybrid IDPS.

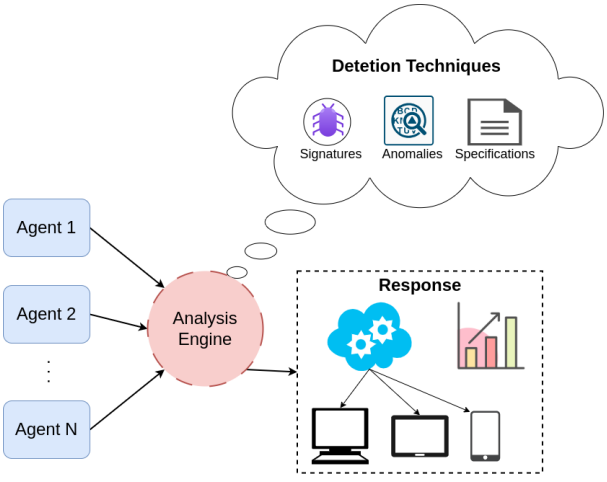

Figure 2: Typical IDPS Architecture Radoglou-Grammatikis and Sarigiannidis (2019).

3.3. Intrusion Detection Techniques

The detection procedures within the Analysis Engine are based on the assumption that the way an intruder behaves differs from the behaviour of a normal/legitimate user, and this difference can be measured using a variety of methods. However, the two behavioural profiles are highly correlated. Therefore, a non-tight interpretation of an intruder's activities and actions will result in the detection of more attackers but, at the same time, it will also lead to more False Positives (FP). On the other hand, a stricter analysis of the intruder's actions will be accompanied by more False Negatives (FN). Figure 3 illustrates the correlation between the behavioural profile of a normal/legitimate user compared to an intruder's. Based on this, it is obvious that there is a practical element of settlement with respect to finding intrusions and anomalies.

Regarding the IDPS, it can be classified into one of two groups based on the methods used by the Analysis Engine: (a) signature & specification-based detection, and (b) anomaly-based detection. Each method is further elaborated in the following subsections.

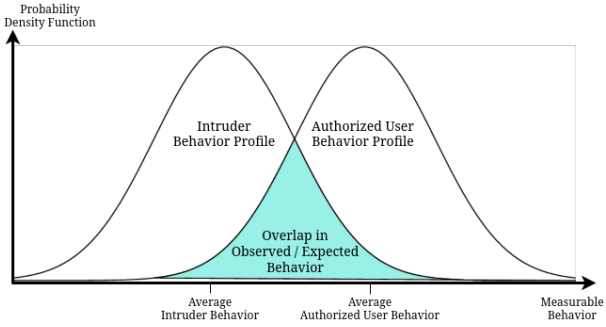

Figure 3: Behaviour Profiles of a Normal/Legitimate User and an Intruder Stallings et al. (2012).

3.3.1. Signature & Specification-based Detection

According to the syntax of the patterns/rules, the IDPS is classified as signature-based or specification-based. Signature-based methods, also known as misuse detection, follow a number of widely known malicious patterns or attack rules (i.e., signatures) without being capable of detecting unknown anomalies and zero-day attacks. On the contrary, specification-based techniques utilize a set of rules (i.e., specifications) that define normal behaviours and, therefore, are capable of detecting unknown anomalies, however, they are inadequate in classifying different types of attacks. Snort, Chakrabarti, Chakraborty and Mukhopadhyay (2010), Suricata, Wong, Dillabaugh, Seddigh and Nandy (2017), and Bro, Udd, Asplund, Nadjm-Tehrani, Kazemtabrizi and Ekstedt (2016), are popular NIDPS of this category. Similarly, OSSEC is a HIDPS of this category, Teixeira, Assunção, Pereira, Malta and Pinto (2019).

3.3.2. Anomaly-based Detection

The anomaly-based IDPS employs a model that distinguishes between normal and malicious behaviour patterns by implementing statistics and AI. More precisely, supervised

ML and DL methods are implemented and trained by utilising previous data from various agents. The training procedure can be established at distinct times or in a continuous way, thus updating the model by feeding it with information regarding new attacks and malicious behaviours. Such methods can identify unknown anomalies and zero-day attacks, since they are able to identify deviations from normal behaviour, but also produce a large number of false alarms. Moreover, this method requires a dataset that includes both malicious and normal data samples, which are rarely publicly available Radoglou-Grammatikis, Sarigiannidis, Efstathopoulos, Lagkas, Fragulis and Sarigiannidis (2021).

Although many ML/DL algorithms exist, the strong majority of them follow the steps outlined below:

- **Pre-processing Phase**: Given an available dataset (labelled or not), in this stage, every data point/instance is appropriately pre-processed according to the feature space and the tunable hyperparameters of the ML/DL methods, which will be used in the next phase. Usually, pre-processing methods like standardization, min-max scaling, normalization, maximum absolute scaling, and robust scaling are implemented, García, Luengo and Herrera (2015).

- **Training Phase**: In this phase, the selected ML/DL model is implemented and trained using the pre-processed data of the first stage. As mentioned, there are various ML/DL models for this purpose. In general, they can be split into four main groups: (a) supervised detection, Cunningham, Cord and Delany (2008), (b) unsupervised/outlier detection, Barlow (1989), (c) semi-supervised/novelty detection, Van Engelen and Hoos (2020) and (d) Reinforcement Learning (RL) detection, Arulkumaran, Deisenroth, Brundage and Bharath (2017). The first category needs a labelled dataset, including a particular label like "*Anomaly*", "*Normal*", or "*Unauthorised Activity*" for each data point/instance. ML/DL models widely recognised in this category include Naive Bayes (NB), Jiang, Zhang and Cai (2008), Linear Discriminant Analysis (LDA), Tharwat, Gaber, Ibrahim and Hassanien (2017), Quadratic Discriminant Analysis (QDA), Tharwat (2016), Decision Trees (DTs) Lomax and Vadera (2013), Random Forest, Resende and Drummond (2018), Logistic Regression (LR) DeMaris (1995), AdaBoost, Sagi and Rokach (2018) and Neural Networks (NNs), Gümüşbaş, Yıldırım, Genovese and Scotti (2020), among others. On the contrary, the second group utilizes clustering mechanisms for unlabelled datasets, Saxena, Prasad, Gupta, Bharill, Patel, Tiwari, Er, Ding and Lin (2017). However, outliers may be present in the training data. Paradigms of methods for detecting outliers include K-Nearest Neighbor (KNN), Cunningham and Delany (2021), Principal Component Analysis (PCA), Ringnér (2008), Angle-Based Outlier Detection (ABOD), Kriegel, Schubert and Zimek (2008), Minimum Covariance Determinant (MCD), Hubert and Debruyne (2010), Stochastic Outlier Selection (SOS), Janssens, Huszár, Postma and van den Herik (2012), Isolation Forest, Hariri, Kind and Brunner (2019), and Local Outlier Factor, Alghushairy, Alsini, Soule and Ma (2020). In the third group, namely the semi-supervised models, the training dataset does not contain any outliers, and the goal is to determine whether a new instance is an outlier or not. OneClassSVM is a typical example in this category, Li, Huang, Tian and Xu (2003). Concerning RL implementations, the goal is to train an agent that interacts with the environment in order to identify the most efficient policy with regard to particular states Lopez-Martin, Carro and Sanchez-Esguevillas (2020a).

- **Inference**: After completing the training procedure, the model is ready to be used by the Analysis Engine. According to the decision made by the model, the relevant alert can be either triggered or not by the Response Module.

3.4. Intrusion Prevention Techniques

After the detection processes, mitigation and prevention actions can be implemented by the response module. A known example is the activation of some firewall rules that aim to isolate cyberattackers. An alternative example is the implementation of honeypots in order to mislead and capture future malicious activities. Finally, the response module can implement the Software-Defined Networking (SDN) technology in order to reduce the impact of cyberattacks or anomalies in near real-time, Xie, Yu, Huang, Xie, Liu, Wang and Liu (2018). In this study, SDN and Honeypots are used by the proposed Security Information and Event Management (SIEM) system in order to mitigate the various threats.

4. Overview of Federated Learning

Federated Learning is an emerging approach in AI/ML that has gained high popularity due to its significantly high performance in diverse and distinct tasks (i.e., intrusion detection, image classification, and object detection) while maintaining user privacy and data confidentiality. Compared to traditional machine learning models that require the centralization of data from various sources, raising privacy concerns and data leakage risks, FL allows models to be trained directly on decentralized devices, such as mobile phones or IoT devices, without the need to exchange or move raw and potentially sensitive data to a centralized entity. This technique aims to advance AI utilization while preserving privacy, enabling a diverse range of applications across different domains. Federated Learning operation is based on a decentralized principle. In particular, a global machine learning model is initially distributed by a central device, like a server, to all participating devices/nodes, each with its local dataset. Since multiple terms are utilized throughout the literature regarding these entities, we will maintain this

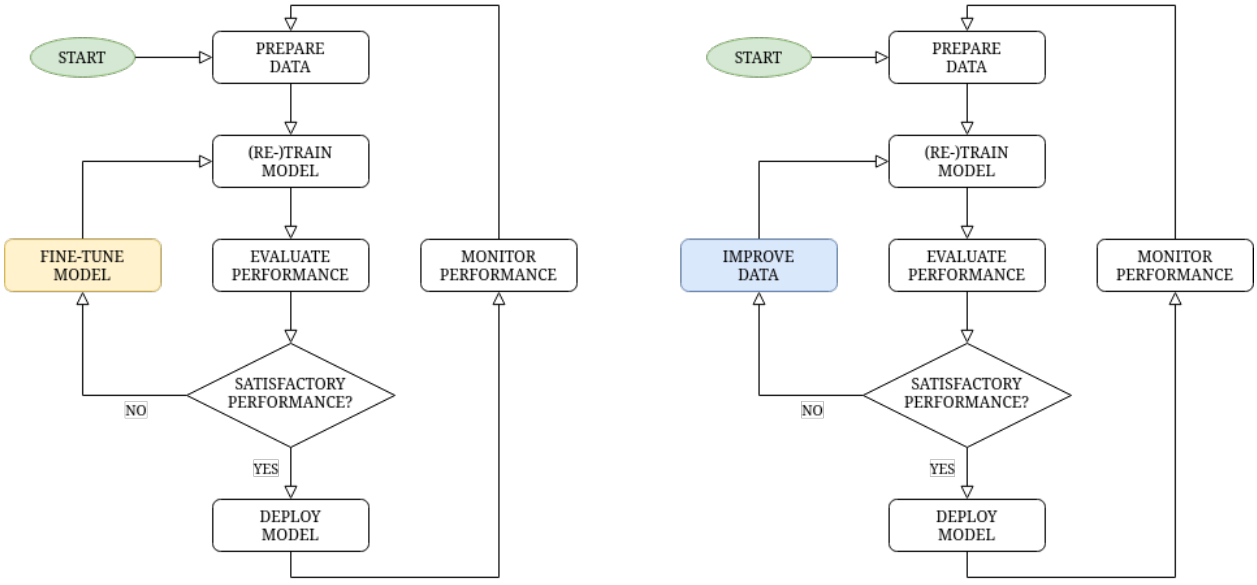

Figure 4: Model-centric vs Data-centric Federated Learning Prendki (2022).

diversity in order to familiarize the reader with them. These devices train the aforementioned transmitted model using their local data and then broadcast the updated models back to the central server. Following, the server is responsible for the aggregation of the transmitted models from the nodes, in order to produce a new global machine learning model that is going to be re-transmitted to the nodes. This continuous procedure is executed for a specific number of training rounds, a pre-defined period of time, or when the model converges to a satisfactory state, such as a given accuracy threshold. In general, the nature of FL reduces data transfer overhead, counters privacy risks, and allows collaborative training of AI/ML models without the disclosure of sensitive or confidential information.

4.1. Model-centric vs Data-centric

The following categories of FL approaches can be identified:

1. **Model-centric** FL: In this technique, a preconfigured and existing NN model is hosted in the cloud. Then, periodically, a number of individual workers spawn, receive a copy of the said model, continue training and improve the received model using their local, isolated, and private data, and finally send their updated model back to the cloud, which performs the local models' aggregation. An example of a model-centric federated learning algorithm is what Google utilizes to improve their Gboard next word prediction model using their customers' Android-powered smartphones, Hard, Kiddon, Ramage, Beaufays, Eichner, Rao, Mathews and Augenstein (2018).
2. **Data-centric** FL: In this case, a preconfigured and existing data source is hosted in the cloud. Then, spontaneously, a number of individual workers spawn, with each one having its own custom model, and train their corresponding models individually on the data offered by the centralized data source in an ad-hoc manner. In this approach, data can be constantly improved and renewed, and the architectures of models can be reevaluated and readjusted based on the data attributes as well as on other useful qualitative and quantitative characteristics of the data.

Currently, the majority of AI applications are model-centric due to limited data quantity and data access restrictions that are enforced on data sharing and distribution, both for preserving privacy and for enforcing intellectual property protection, which makes it hard to create datasets that are extensive, representative and generally recognized as public standards. Furthermore, model-centric FL focuses on fine-tuning an NN on a dataset with a fixed set of attributes for solving a specific problem. However, data-centric FL is more suitable for obtaining more accurate and continuously evolving AI models since data are the most valuable assets in data science workflows. Thus, a centralized and rich data source can lead to a more accurate reflection of the real world, especially for evidence that changes over time. Additionally, with the data-centric approach, model architectures can change over time, adapting to new circumstances and providing more accurate and up-to-date results. An abstract comparison between the two aforementioned FL approaches is presented in Figure 4.

4.2. Cross-device vs Cross-silo

Model-centric FL can be further distinguished into two different federated learning-based settings based on the scale of the federation, using the following distinctive terms:

1. **Cross-device** FL: Cross-device FL refers to model training through employing a highly scalable number of computing devices, achieving a massively parallel training procedure, Kairouz, McMahan, Avent, Bellet, Bennis, Bhagoji, Bonawitz, Charles, Cormode, Cummings et al. (2021). Each of these devices has limited computational and storage resources and its own private dataset, which is only locally accessible, making the data most likely Independent and Identically Distributed (IID) with a fixed partitioning. During the federating training phase, computational nodes can be added or removed at any point, Kholod, Yanaki, Fomichev, Shalugin, Novikova, Filippov and Nordlund (2020a). Eventually, the only central entity is the model aggregation server, which orchestrates the overall federated learning procedure. These computing devices are usually edge devices, e.g. Smartphones, and/or IoT devices. However, with this approach, even though the number of available training devices can be millions, only a fraction of them are randomly employed as actual workers due to limited availability.
2. **Cross-silo** FL: Cross-silo FL refers to model training through employing a small and mostly fixed number of clients (e.g. less than a hundred), including organizations and/or companies or other geographically distributed data centres, where each client participates in all phases of the training procedure, Kairouz et al. (2021); Huang, Huang and Liu (2022). Each of these devices has its own computational and storage resources, either limited or high-performance, as well as its own local and private dataset with a fixed partitioning, making the data most likely non-independent and identically distributed (non-IID) since different participating organizations tend to collect different types of data and/or perform different data engineering before storing the collected data.

The main drawback of Cross-silo FL is its significantly smaller scaling capabilities since large organizations can not scale as well as cross-device FL, which consists of plenty of edge devices. Also, when a participating computing device loses its connection to the network, an appropriate handling mechanism must exist to resolve any synchronization and exchange issues. Moreover, the fact that data is characterized by non-IID introduces additional challenges in data handling, Li, Wen, Wu, Hu, Wang, Li, Liu and He (2021c). On the other hand, each client is almost always available and can continuously perform both computation and communication tasks during the whole training procedure. This guarantees higher overall training stability. An abstract comparison between the two aforementioned model-centric FL approaches is presented in Figure 5.

4.3. Horizontal FL vs Vertical FL

Depending on the difference between datasets of distinct parties participating in the FL procedure, classification can be extended in two different FL settings using the following distinctive terms:

1. **Horizontal** FL (HFL): HFL or sample-based FL, describes cases in which different datasets use the same number of features but differ in the number and value of samples, Yang, Liu, Chen and Tong (2019). For instance, two different banks in the same area operate the same business, so the feature spaces are the same, although each one has its own clients, some of which may be common. This FL setting mostly applies to cross-device FL.

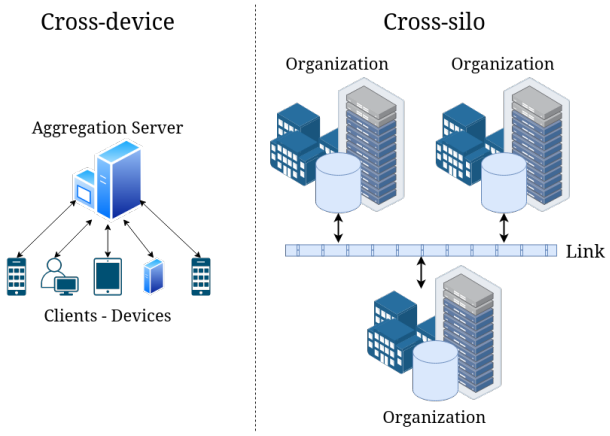

Figure 5: Cross-device vs Cross-silo Federated Learning Kholod et al. (2020b).

2. **Vertical** FL (VFL): VFL or feature-based FL, describes cases in which distinct datasets share the same sample IDs (e.g. users) but differ in the number of features that they use, Yang et al. (2019). For instance, a bank and a hospital in the same local area are likely to have most of the area's residents as clients, thus they have a large user space intersection. However, since their business models are completely dissimilar, the data of the bank's business model are most likely different from the hospital's data, which means that their feature spaces are distinguishable. Vertical FL collaboratively performs the aggregation of different features from all parties, and preserves privacy during the computation of each model's gradients.

In cases where separate datasets differ both in the number of features and the number of instances/samples, **Federated Transfer Learning (FTL)** is used, Pan and Yang (2010). In this scenario, a model is trained on the limited common representation between the two feature spaces, followed by transfer learning to the training result, to obtain predictions for samples with features from each feature side separately. FTL is an important addition to the techniques of FL because it deals with problems that exceed the scope of current FL techniques. An abstract comparison among the three aforementioned federated learning settings is presented in Figure 6.

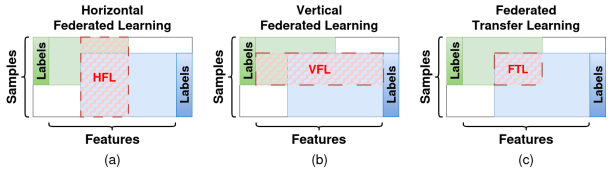

Figure 6: Horizontal Federated Learning vs Vertical Federated Learning vs Federated Transfer Learning Asad et al. (2020b).

4.4. Security and Privacy

Two additional major issues FL has to deal with are security during gradients/parameters transmission between participating devices and data leaks which can occur from applying reverse engineering to a model. However, since FL can take place even in IoT environments and networks, where devices have limited computational resources and battery capacity matters, all these need to be accompanied by techniques that are able to reduce the number of bytes that are being transmitted as well as the total time required for the whole federated training to be completed. To that end, numerous studies have been carried out and a variety of techniques have been proposed.

Updated model parameters are transferred from the participating worker to the aggregation server. These model parameters may contain private information, such that even individual data points that the model was trained on can be reconstructed. In order to overcome concerns about models memorizing sensitive user data, leading to data leaks, the implementation of a **Differential Privacy** (DP) mechanism into FL has been proposed, Dwork, McSherry, Nissim and Smith (2006b); Dwork, Kenthapadi, McSherry, Mironov and Naor (2006a).

Moreover, due to the updated model parameters being transferred between participants, an additional concern that arises is the parameters' security, meaning that a third party can not access these local models. This is where parameter encryption comes into play. However, since the training procedure may consist of hundreds of low-resource participating devices, an efficient encryption algorithm is mandatory. Towards this goal, a variety of techniques have been proposed, with the most famous being the **Additively Homomorphic Encryption**, as presented in Figure 7, Phong, Aono, Hayashi, Wang and Moriai (2017).

Finally, since modern NN models consist of a huge number of parameters, which significantly increases the model sizes to hundreds of MBs or even GBs, to avoid encrypting and transmitting all of these parameters, different **data compression** techniques have been proposed, like those presented in Deng, Chen, Zhang, Gong and Zhu (2019). In any event, all of the aforementioned complications are still under active research.

4.5. Federated Learning Aggregation Strategies

An aggregation strategy in federated learning refers to the method used to combine the local models from various clients to update the global model. This strategy is crucial for ensuring that the global model benefits from the diverse data distributions of the clients, while an effective aggregation can significantly improve the performance and robustness of the global model.

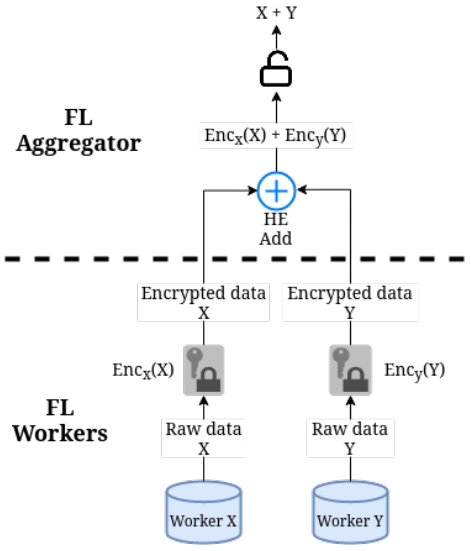

Figure 7: Additively Homomorphic Encryption Roth et al. (2021).

4.5.1. *Federated Averaging - FedAvg & FedSGD*

Federated Averaging (FedAvg) is a generalization of Federated Stochastic Gradient Descend (FedSGD), which is a parallel/federated version of the classic SGD, McMahan, Moore, Ramage, Hampson and y Arcas (2017); Robbins and Monro (1951). The main differences between these two baseline fusion techniques are situated in the number of SGD steps performed locally in each client and in the type of data that is collected on the aggregation server. Regarding FedSGD, every participating worker performs a single SGD step in each federated training round, Stich (2018). On the other hand, in FedAvg, every participating worker performs more than one SGD steps in each training round. After completing all the steps of SGD, each client transmits its updated model's parameters (weights and biases) to the aggregation server. Figures 8 and 9 provide a schematic overview of these two techniques and their differences.

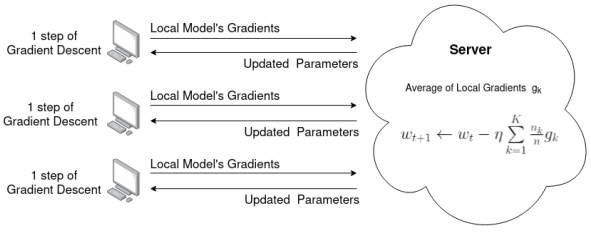

Figure 8: FedSGD Conceptual Figure.

With regard to each method's advantages, FedAvg is characterized by a low communication cost as most of the computation is performed locally, additionally to its robustness to imbalanced and non-IID data. On the other hand, convergence is not guaranteed when FedAvg is used. FedSGD superiority is identified in guaranteeing convergence and efficiency in terms of computation. However, the numerous communication rounds that are demanded to reach convergence significantly increase the communication cost.

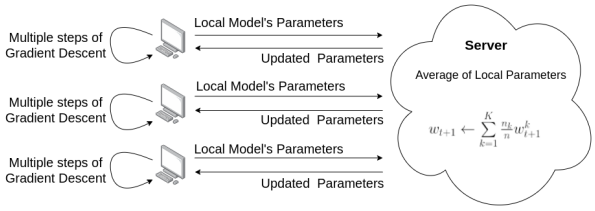

Figure 9: FedAvg Conceptual Figure.

4.5.2. FedProx

FedProx is thought to be a generalization of FedAvg, Li, Sahu, Zaheer, Sanjabi, Talwalkar and Smith (2020b). Its goal is to use all available clients/worker devices, whereas FedAvg selects a subset of these, while guaranteeing convergence. Distinct worker devices in FL systems often have different constraints related to their limited resources in terms of the capabilities of the available hardware, network connection reliability, and current battery status. FedProx tolerates different amounts of work to be held locally across devices based on their available system resources and then averages the solutions sent from each client (worker). However, a high number of local updates may still cause these solutions to diverge due to the heterogeneity of the data.

Towards avoiding divergence, FedProx adds a proximal term μ to effectively limit the impact of variable local updates. This proximal term has the two following advantages:

- It deals with the issue of statistical heterogeneity by setting certain restrictions on the local updates in order to keep them closer to the initial (global) model without the need to manually tune the number of local training epochs.
- It tolerates the incorporation of variable amounts of local work originating from systems' heterogeneity.

Figure 10 provides a schematic overview of the FedProx technique.

With regard to the parameter μ it may be considered as a constant penalty which affects convergence. Apparently, when $\mu = 0$ FedProx has the same behaviour as FedAvg.

The convergence rate is calculated using the statistical heterogeneity/device dissimilarity in the network, so convergence is achieved under a bounded assumption on dissimilarity between the local functions.

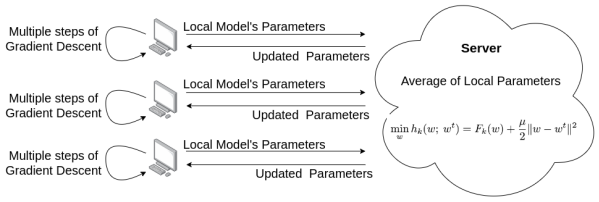

Figure 10: FedProx Conceptual Figure.

4.5.3. FedDANE

FedDANE adopts a similar approach to FedProx, while drawing inspiration from inexact-DANE, Shamir, Srebro and Zhang (2014), a variant of DANE, Shamir et al. (2014); Reddi, Konečnỳ, Richtárik, Póczós and Smola (2016), that allows for local updating and is useful when the device communication is characterized by a bottleneck, Li, Sahu, Zaheer, Sanjabi, Talwalkar and Smithy (2019). Compared to FedAvg, DANE, and inexact-DANE algorithms try to solve a different local problem which uses two additional terms - a gradient correction term and a proximal term. Due to data being characterized by statistical heterogeneity in federated environments, the convergence can be potentially improved by forcing each client to train and update its model while keeping it as close as possible to the current global model, increasing the stability and the amenability of the method to theoretical analysis. Taking advantage of the gradient correction term, DANE allows the model update to become an approximate Newton-type method, leading to improved convergence, which can be proven when it has to deal with well-behaved objectives, Reddi et al. (2016). Identical to inexact DANE, FedDANE inexactly solves an approximate Newton-type subproblem, with the difference that it only aggregates updates from a specific number of devices at each round (Figure 11).

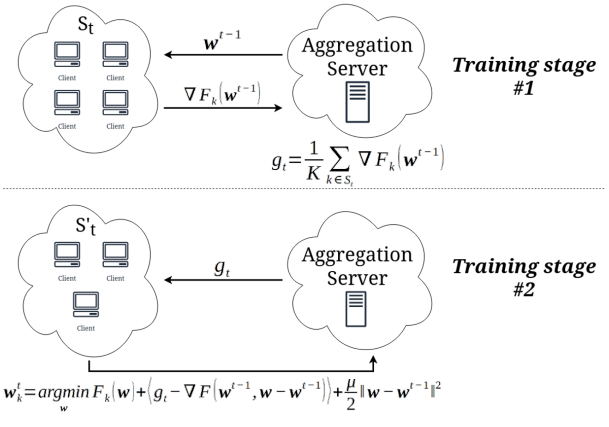

Figure 11: FedDANE Conceptual Figure.

4.5.4. FedOpt

Each one of the aforementioned federated learning techniques suffers from certain drawbacks which need to be addressed. Towards this goal, FedOpt, Asad, Moustafa and Ito (2020a), was proposed which aims at the following:

- Communication overhead reduction,
- Efficient and secure gradient aggregation,
- Privacy preservation.

Since hundreds of MBs need to be transferred among devices for model parameter updates, tackling communication overhead should be a priority for achieving an efficient federated training procedure. FedOpt proposes an optimization based on Distributed Stochastic Gradient Descent (DSGD) by introducing temporal sparsity into DSGD to reduce the total communication delay, Dean, Corrado, Monga, Chen, Devin, Mao, Ranzato, Senior, Tucker, Yang et al. (2012). This temporal sparsity leads to the design of a Sparse Compression Algorithm (SCA) for FedOpt, which reduces the number of transferred parameter bits during the federated training by up to 3 times, meaning that an x3 better communication performance can be achieved. Not only that, but SCA, also, allows each participating device to perform multiple epochs of SGD to locally train its model in each round.

Additionally, as previously mentioned, there exists the need for FedOpt to identify malicious nodes in the network as well as a dishonest centralized cloud aggregator which may be interested in inferring private user data from the received local models. To enforce model and data security, locally trained models' parameters should be encrypted prior to transmission. Simultaneously, the cloud server, after receiving the encrypted parameters, should be the only one able to aggregate them without being able to decrypt any of the received models, to avoid private and sensitive information leaks. This can be achieved through the use of Additively Homomorphic Encryption. Ultimately, for additional data privacy protection, Differential Privacy is employed through the use of a Laplacian mechanism, Dwork, McSherry, Nissim and Smith (2006c). This mechanism adds random noise to the parameters of the local models' gradients.

Another advantage of FedOpt is its tolerance against devices unexpectedly dropping out of the federated training procedure while suffering negligible accuracy loss. Furthermore, because of the use of gradient compression, the desired level of convergence is achieved in much fewer training epochs than it would normally require. However, the authors do not mention any convergence guarantees of FedOpt, although studies like Meng, Chen, Wang, Ma and Liu (2017) and Swenson, Murray, Kar and Poor (2020) have showcased the conditions under which DSGD converges.

A more technical FedOpt implementation is presented in Reddi, Charles, Zaheer, Garrett, Rush, Konečnỳ, Kumar and McMahan (2020) where the authors utilized known and effective optimization algorithms, such as ADAM, ADAGRAD, and YOGI for the federated aggregation strategy. In particular, they used a set of parameters that control various aspects of the optimization of model parameters, and they introduced the following strategies:

1. **FedAdam** that incorporates a pair of decay parameters, which regulate the significance that the aggregation gives to past updates and the significance assigned to updates of the current model within the algorithm,
2. **FedAdagrad** that aggregates client models based on the distance between each client's model and the global model held by the server, and
3. **FedYogi** that utilizes both the distance of nodes' models from the global model, the direction of it, and, also, the aforementioned decay parameters.

In general, all of the above aggregation techniques aim to minimize the communication overhead, while providing a high level of adaptability in cases where the data are non-IID, partial node participation is observed, and data sizes significantly differ across the participating nodes.

5. Analysis of Federated Intrusion Detection and Prevention Systems

Toward gathering existing knowledge regarding the IDS techniques encompassing the FL approach, a thorough analysis of SOTA implementations is presented in this survey paper. More specifically, the literature that will be discussed henceforth is incorporated in Table 1. For each study that we will elaborate on, all the distinct characteristics that are required for a comprehensive demonstration are provided. In that manner, the reader is equipped with an easy-to-compare representation of this study.

First and foremost, due to the fact that Deep Neural Networks (DNNs) require a huge amount of data in order to have a high detection accuracy of intrusion attacks in a network, Z. Tang et al. proposed an FL approach, in which different Internet Service Providers (ISPs) need to share their collected network traffic data, but they will also collaborate with each other, in order to train a global model, Tang, Hu and Xu (2022). More precisely, each ISP (federated client) performs a small number of training iterations for its Gated Recurrent Unit (GRU) network and sends the resulting network parameters back to the server. Subsequently, the server will compute the average of the clients' parameters and will send the result back to the ISPs, Ansari, Bartoš and Lee (2022). The dataset that the proposed solution was evaluated on, was CIC-IDS2017, Gharib, Sharafaldin, Lashkari and Ghorbani (2016); Sharafaldin, Gharib, Lashkari and Ghorbani (2018a,b), in which Accuracy, Precision, Recall, and F1-Score were measured, when identifying a variety of attacks. The simulation took place using PySyft, Ziller, Trask, Lopardo, Szymkow, Wagner, Bluemke, Nounahon, Passerat-Palmbach, Prakash, Rose et al. (2021a), for the Federated Learning part, and PyTorch, Paszke, Gross, Chintala, Chanan, Yang, DeVito, Lin, Desmaison, Antiga and Lerer (2017), for the GRU implementation, while FedAvg was utilized as the fusion method when exploiting the Internet Protocol (IP) and Transmission Control Protocol (TCP). Regarding

GRU, the hidden layer consisted of 256 units, and the output layer was a fully connected layer of 15-dimensional tensors (equal to the number of different network traffic types). The results indicated that the accuracy of each ISP's detection mechanism was higher when they joined the federation learning, compared to when they trained their models using only local data, reaching an accuracy level of 97.2% on the task that was developed. An overview of the GRU architecture is displayed in Figure 12.

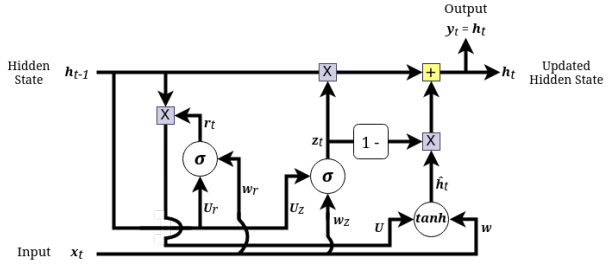

Figure 12: Gated recurrent unit Vasilev et al. (2019).

R. Zhao et al. in Zhao, Yin, Shi and Xue (2020) proposed an intelligent IDS using Long Short-Term Memory (LSTM) running through an FL approach in order to preserve privacy, Kundu, Yu, Wynter and Lim (2022); Althubiti, Jones and Roy (2018); Imrana, Xiang, Ali and Abdul-Rauf (2021); Alaeddine (2020). An overview of the LSTM architecture is presented in Figure 13. More specifically, their proposed model focuses on the identification of high-risk malicious behaviour, such as directory traversal attacks, a large number of reads, taking access and erasing files in bulk, and getting rid of software in bulk, among others. The target system of this study is Unix-like OS, whereas the protocol that is exploited is Shell commands which are stored in the bash history file. In terms of implementation, a Bidirectional LSTM (BiLSTM) is transmitted to all participating users by the server. BiLSTM is a two-way LSTM (it consists of a forward LSTM and a backward LSTM) and it is utilized for modeling the bi-directional connection between command input before and after preprocessing. This model was trained on the open-source SEA dataset, which is the most popular intrusion detection command record dataset produced by the AT&T Shannon Lab. Each local model receives user commands as input, and uses a tokenizer for the preprocessing, in order for the results to be fed into the forward and the backward LSTM. Following, there exists a dropout layer which randomly ignores a fraction of neurons during training in order to avoid overfitting. Each user sends its results to the server, which aggregates them using a weighted average method in order to update the parameters of the global model and send them back to the users. For this implementation the authors employed the TensorFlow framework, Abadi, Agarwal, Barham, Brevdo, Chen, Citro, Corrado, Davis, Dean, Devin, Ghemawat, Goodfellow, Harp, Irving, Isard, Jia, Jozefowicz, Kaiser, Kudlur, Levenberg, Mané, Monga, Moore, Murray, Olah, Schuster, Shlens, Steiner, Sutskever, Talwar, Tucker, Vanhoucke, Vasudevan, Viégas, Vinyals, Warden, Wattenberg, Wicke, Yu and Zheng (2015), along with the scikit-learn library, while the fusion technique that was utilized was FedAvg. At first, the Bi-LSTM was compared with a Convolutional Neural Network (CNN) and the results indicated that the former produced higher accuracy and lower loss. Finally, a comparison between the Federated Learning Bi-LSTM (FL-LSTM) with a Centralized Bi-LSTM (CL-LSTM) showcased that the accuracy of the FL-LSTM model was 99.21%, the recall was 99.23%, and the F1 value was 99.21%. Regarding the CL-LSTM, its accuracy was 99.51%, having a 0.3% difference compared to the FL-LSTM.

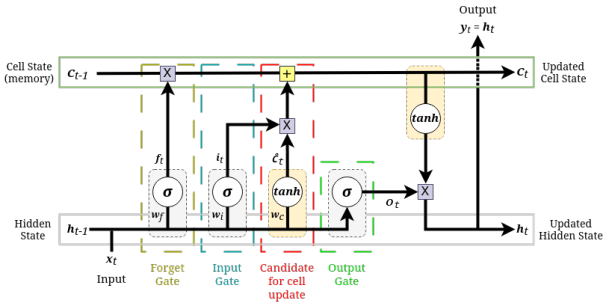

Figure 13: Long short-term memory neural networks architecture Dobilas (2022).

Y. Zhao et al. proposed a multi-task DNN which simultaneously performs network anomaly detection, VPN traffic recognition tasks, and traffic classification tasks, and is implemented using Federated Learning to reduce training time and achieve better results, Zhao, Chen, Wu, Teng and Yu (2019). The DNN model which was developed consists of fully connected layers using Leaky Rectified Linear Unit (LeakyReLU) as their activation function for the implementation of which the PyTorch library was utilized. In terms of the FL architecture, there exists an input layer, certain layers that are shared, and task layers. The external layers are connected with the input layers of the NN. The shared layers' goal is the extraction of features from the input layer. Last but not least, the task layers consist of smaller networks connected with the shared layers, and every subnetwork is responsible for a specific job. It is worth noting that this architecture is more efficient because shared layers can decrease the number of parameters in the network. The fusion technique that was employed was FedAvg, whereas the protocols on which it focused were IP, TCP, Virtual Private Network (VPN) and Tor. Three datasets were employed for the evaluation of the proposed model. Firstly, the CIC-IDS2017 dataset was used, Sharafaldin et al. (2018b). This dataset provides real-world labelled network traffic, including benign and malicious traffic in PCAP format. Next, the ISCXVPN2016 dataset was used, which contains 7 categories of encrypted network traffic in browsing, chat, streaming, mail, VoIP, P2P and File Transfer, Draper-Gil, Lashkari, Mamun and Ghorbani (2016). For each instance, the regular and the VPN sessions are stored in PCAP format. The last dataset used is the ISCXtor2016,

which expresses instances in Tor forms, Lashkari, Draper-Gil, Mamun, Ghorbani et al. (2017). The evaluation was performed using Accuracy, Precision, and Recall metrics. MT-DNN-FL was compared to certain centralized methods, like K-NN, DTs, and RF, and the results indicated that the proposed method achieved an accuracy of 98.14%, which is higher than the aforementioned centralized methods.

In an IoT environment, the ML models that are used for the detection of anomalies and attacks cannot be trained using a centralized dataset due to privacy and security reasons, and, thus, their training takes place using only local data collected on the devices, making them less accurate. In order to increase anomaly detection accuracy, V. Mothukuri et al. in Mothukuri, Khare, Parizi, Pouriyeh, Dehghantanha and Srivastava (2021) proposed an FL approach, in which each IoT device, placed in an IIoT environment or not, will train a model using only its local data. Then, the parameters of all trained models are sent to the FL server, which aggregates them and sends back the updated weights to each IoT device. By adopting an FL approach, the data of each IoT device are kept private, and the accuracy of each model is expected to increase. In terms of architecture, four GRU models are employed which are trained in a Federated Learning scheme, followed by a Random Forest Decision Tree Ensembler (presented in Figure 14). The fusion technique that is applied in this paper is FedAvg, while the attack scenarios on to which it elaborates are Man In The Middle (MITM), Ping (ICMP) flood Distributed Denial of Service (DDoS) Flood, Modbus Query Flood and SYN flood DDoS. The protocols which were employed for the evaluation procedure include Modbus, Remote Terminal Unit (RTU), IP, TCP and Message Queuing Telemetry Transport (MQTT). Even though the datasets that it was validated on are not explicitly mentioned, the libraries that the authors utilized were PySyft and PyTorch. Regarding the performance of the proposed approach, it was found that it outperformed the accuracy of a centralized approach in detecting attacks, while, simultaneously, securing the privacy of users' data and reaching 90.25%.

Along with the rapid growth of infrastructures implementing intelligent networking and computing technologies, such as 5G, an increase in the number of attacks against Cyber-Physical Systems (CPSs) is observed. A CPS consists of multiple and different systems, which can be described as a physical system controlled in combination with embedded software, as displayed in Figure 15. Due to the fact that there are no sufficient high-quality examples to train ML models in order to detect intrusions, B. Li et al. in Li, Wu, Song, Lu, Li and Zhao (2021a) proposed a novel approach called DeepFed mainly concentrating on Industrial CPS and Supervisory Control and Data Acquisition (SCADA) systems. The protocol that the authors take into account is Modbus, while they consider numerous attacks to be identified by their model, namely reconnaissance, response injection, command injection, Denial of Service (DoS) and eavesdropping of data resources and/or model parameters. With regard to their model, DeepFed, as shown in Figure 16, consists of a CNN, O'Shea and Nash (2015); Albawi, Mohammed and Al-Zawi (2017); Li, Liu, Yang, Peng and Zhou (2022); Vinayakumar, Soman and Poornachandran (2017); Mohammadpour, Ling, Liew and Chong (2018), in parallel with a GRU, the outputs of which are given to an MLP, and finally to a softmax layer for the classification the attacks. Furthermore, they adopted an FL approach, so that different CPSs can collaboratively build a global intrusion detection model without sharing their private data. Additionally, a Paillier cryptosystem was implemented for the secure transmission of model parameters of each CPS during the training phase, Paillier (1999). Regarding the performance of DeepFed, a high effectiveness in detecting attacks was observed, with the accuracy levels exceeding 99%, along with overall better performance compared to SOTA schemes. For their implementation, authors employed the Keras backend as well as the Flask framework.

B. Cetin et al. tried to address the problem of vulnerabilities that exist in wireless networks by exploiting the 802.11 protocol, Nicopolitidis, Obaidat, Papadimitriou and Pomportsis (2003); Kavitha and Sridharan (2010); Mitchell and Chen (2014); Alrajeh, Khan and Shams (2013), and by proposing ML and DL methods for the identification and mitigation of various types of intrusion attacks (injection, impersonation and flood), Cetin, Lazar, Kim, Sim and Wu (2019). The major problem with training a global centralized model is the limited amount of data available due to security reasons. In this sense, this paper adopts an FL approach, which preserves privacy and addresses some security concerns while employing the FedAvg fusion technique. In terms of implementation, edge devices are trained locally, and then a global model is constructed by aggregating each model's parameters and averaging them using LEAF, Caldas, Duddu, Wu, Li, Konečnỳ, McMahan, Smith and Talwalkar (2018). More precisely, Stacked AutoEncoders (SAEs), Farahnakian and Heikkonen (2018); Kingma and Welling (2013a), are used for the detection of anomalies/attacks, which are designed to provide a compressed illustration of anomalous observations. A schematic overview of an Autoencoder (AE) is presented in Figure 17 and of an SAE in Figure 18. Regarding the performance of the proposed model, it was evaluated on the AWID Dataset, Kolias, Kambourakis, Stavrou and Gritzalis (2015), and it was found that it increased the classification accuracy, reaching approximately 83%, while decreasing both computation and communication costs.

For implementing IoT in the transportation sector, the Internet of Vehicles (IoV) plays an important role in the design of Smart Transportation Systems (STSs), Fangchun, Wang, Li, Liu and Sun (2014); Contreras-Castillo, Zeadally and Guerrero-Ibañez (2018). The interconnected vehicles and the transportation infrastructures, which comprise an STS, are vulnerable to a variety of cyber intrusions. M. Abdel-Basset et al. in Abdel-Basset, Moustafa, Hawash, Razzak, Sallam and Elkomy (2021) proposed a Federated Deep Learning (FDL) intrusion detection framework called FED-IDS. This framework can accurately identify attacks by implementing a context-aware transformer network to learn how traffic flow is represented in vehicles, and identify attacks in IoT, IoV and STS systems. The protocol which the authors exploited

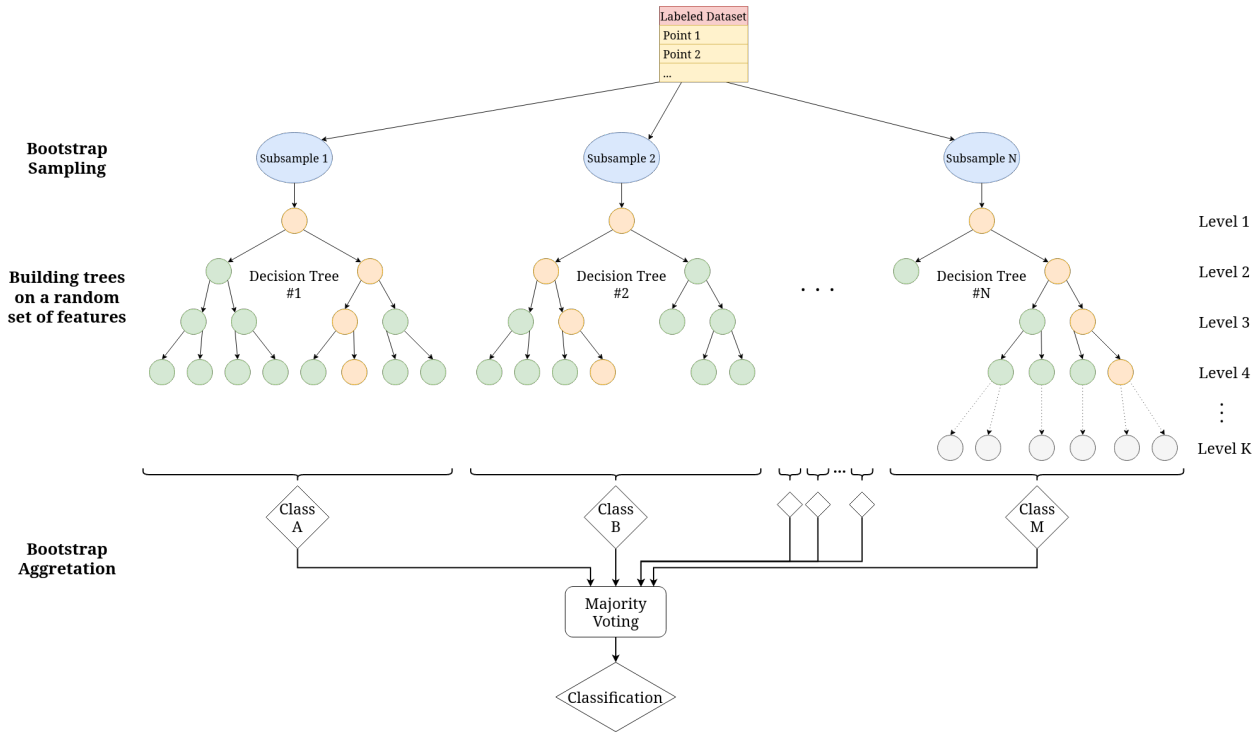

Figure 14: Random Forest Decision Tree Ensembler architecture Gao et al. (2021).

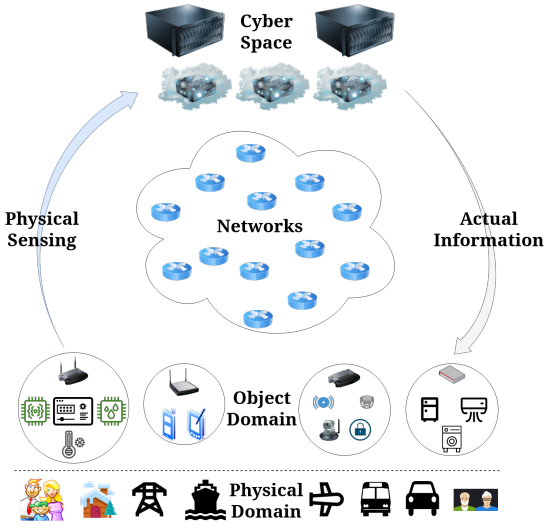

Figure 15: Cyber-physical Systems Ali et al. (2015).

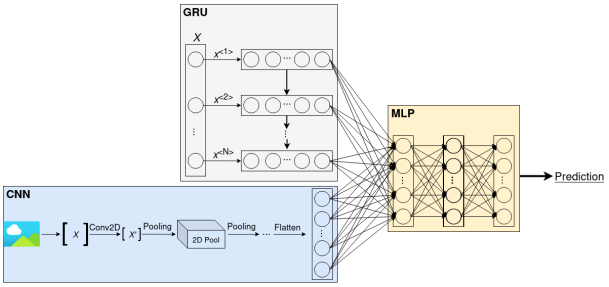

Figure 16: CNN-GRU architecture, which consists of a CNN in parallel with a GRU, followed by an MLP Wu et al. (2020).

was MQTT. In this manner, FED-IDS was designed using three main modules: the Encoder, the Decoder, and the Classification module. Furthermore, a blockchain approach is adopted in the training part in order to let the different edge nodes be part of the training in a secure and reliable manner, Monrat, Schelén and Andersson (2019); Nakamoto (2009); Meng, Tischhauser, Wang, Wang and Han (2018). For this FDL implementation, the PySyft library was utilized. FED-IDS was evaluated using two different datasets, namely the Car-Hacking dataset, Song, Woo and Kim (2020); Seo, Song and Kim (2018), for which the attacks that were taken into account were IoT traffic including normal, scanning, DoS, DDoS, ransomware, backdoor, injection, Cross-Site Scripting (XSS), Password Cracking Attack (PWA), MITM attack and the TON_IoT dataset, Moustafa (2021a), which includes spoofing of the drive gear and spoofing the RPM gauge, fuzzy attack and DoS attack, Moustafa (2021a); Booij, Chiscop, Meeuwissen, Moustafa and den Hartog (2021); Alsaedi, Moustafa, Tari, Mahmood and Anwar (2020); Moustafa, Keshky, Debiez and Janicke (2020b); Moustafa, Ahmed and Ahmed (2020a); Moustafa (2019, 2021b); Ashraf, Keshk, Moustafa, Abdel-Basset, Khurshid, Bakhshi and Mostafa (2021). The results indicated its superiority over existing SOTA approaches, reaching a mean accuracy of 92.5%

and 97.2% in the TON_IoT and the Car-Hacking dataset, respectively.

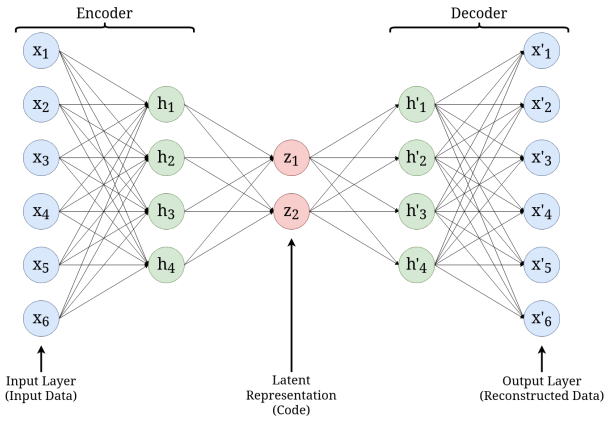

Figure 17: AutoEncoder Song et al. (2021a).

In order to ensure security in IoT, as well as in Wireless Edge Networks (WENs), Z. Chen et al. in Chen, Lv, Liu, Fang, Chen and Pan (2020) developed an intrusion detection algorithm called FedAGRU. FedAGRU stands for Federated Learning-based Attention GRU, and its main difference from centralized learning methods is that it updates universal learning models without the need to share raw data between edge devices and a server. The Attention mechanism is used as a penalizing method for devices that have poor performance in order to avoid unnecessary parameters transfer to the server, Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser and Polosukhin (2017); Niu, Zhong and Yu (2021). In other words, Attention can be treated as a rewarding mechanism for the devices, improving the model's accuracy. When evaluating FedAGRU on the KDDCup99, University of California, CIC-IDS2017 and WSN-DS, Almomani, Al-Kasasbeh and Al-Akhras (2016), datasets, it was found that its detection accuracy was increased by almost 8%, while simultaneously decreasing communication cost by almost 70%, compared to centralized methods. More specifically, the accuracy levels that FedAGRU achieved were 99.28% and 98.82% on IID and non-IID, respectively. More importantly, the accuracy that was reported encompassed the distinct and various data that were included in the three datasets that the authors utilized. Additionally, It was found that FedAGRU is robust against poisoning attacks. Finally, for this implementation, the PySyft library was employed.

Due to the increasing popularity of IoT architectures, many IoT-specific, as well as IIoT-specific attacks have been developed, making interconnected devices vulnerable in terms of security and user data privacy. In this regard, D. C. Attota et al. in Attota, Mothukuri, Parizi and Pouriyeh (2021) tried to take advantage of the Multi-view learning, Xu, Tao and Xu (2013); Sun (2013), in which the training of an ML model takes place using different data views and combines it with a Federated Learning-based IDS which will be trained using different IoT devices in a decentralized approach, with its main goal being to detect, classify, and mitigate the strong majority of these attacks. More specifically, the authors of this study focused on scanning and brute force attacks by exploiting the MQTT protocol as it provides the session layer for communication among devices. With regard to Multi-view Federated Learning Intrusion Detection (MV-FLID), as it is called, the multi-view learning technique will be used in order to predict as accurately and efficiently as possible different types of attacks, while the FL aspect will keep each device's data private and will perform the aggregation taking advantage of peer learning. Concerning the evaluation of the proposed method, it was found that MV-FLID had a higher accuracy compared to centralized approaches, while reporting 98% accuracy on the MQTT-IoT-IDS2020 dataset, Hindy, Bayne, Bures, Atkinson, Tachtatzis and Bellekens (2021). The fusion technique that was employed included the FedAvg method while exploiting the PySyft and PyTorch libraries.

The technology used in Beyond 5G networks has significantly decreased the latency between different applications and devices. However, several cyber-security issues have been raised due to the fact that real-time applications transfer data continuously from edge devices to dedicated computing servers, increasing the risk of data leakage. To address this issue, K. S. Kumar et al. in Kumar, Nair, Roy, Rajalingam and Kumar (2021) proposed a federated ML mechanism, which implements Paillier Homomorphic Encryption and Differential Privacy, Dong, Roth and Su (2019). Additionally, they designed an Artificial Immune IDS in order to identify and classify any anomalies and attacks in the network flow, such as DoS attacks, User to Root (U2R), Root to Local (R2L) and probe attacks. Their implementation was based on the PySyft library, exploiting the MQTT protocol, while targeting Multi-access Edge Computing (MEC) devices. In terms of performance, the results indicated that the proposed system is better than existing edge approaches, while simultaneously providing more secure communication between edge nodes, which is depicted in an accuracy level of 92.7% derived on the CIFAR-10, Krizhevsky, Hinton et al. (2009), and KDDCup99 datasets.

The vulnerability of interconnected vehicles and transportation infrastructures to cyber intrusion attacks due to the wide usage of software and wireless interfaces, raises the need for high-performance IDSs. These IDSs need to, subsequently, be integrated into high computational network devices due to the continuous training and updating of models, yet in most cases, the resources in these devices are restricted. H. Liu et al. in Liu, Zhang, Zhang, Zhou, Shao, Pu and Zhang (2021) proposed a cooperative IDS between the edge devices, like vehicles, and roadside units (RSUs), targeting systems like IoV and Vehicle-to-Everything (V2X). This FL approach, encompassing the averaging method as the fusion technique, which was, also, based on the PySyft and PyTorch libraries, reduces the number of required resources while assuring the privacy and security of data. In order to avoid any data leakage and preserve security in the aggregation phase, blockchain mechanisms were used, such as the Ethereum protocol, for broadcasting the trained models, Alkadi, Moustafa, Turnbull and Choo (2021); Alexopoulos, Vasilomanolakis, Ivánkó

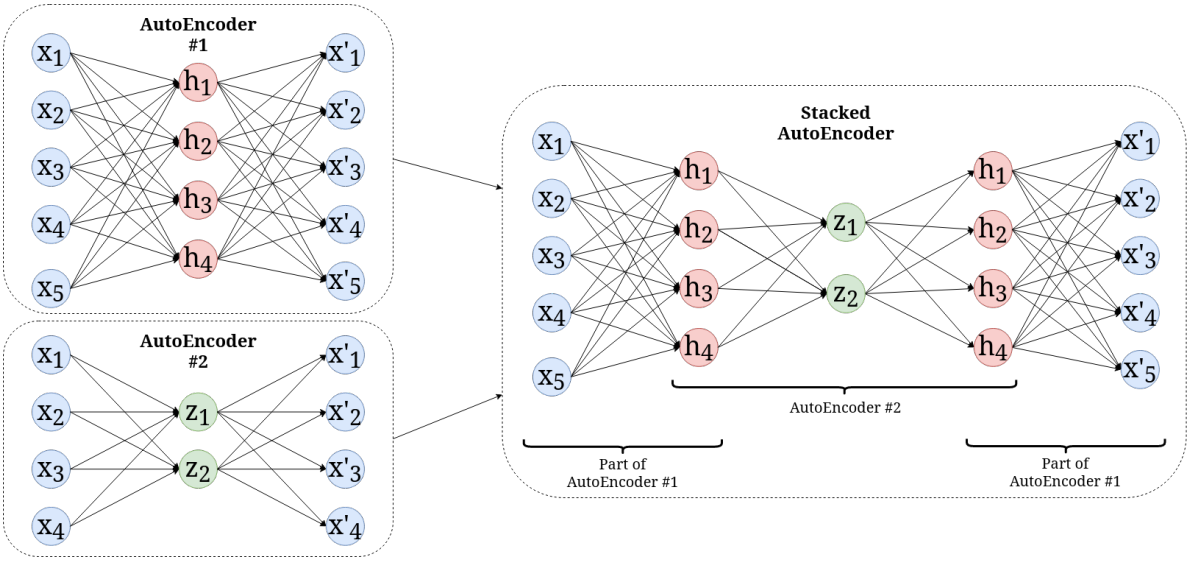

Figure 18: Stacked AutoEncoder Liu et al. (2018).

and Mühlhäuser (2018); Liang, Shanmugam, Azam, Karim, Islam, Zamani, Kavianpour and Idris (2020). As it was found, the proposed method achieved high accuracy, and more specifically higher than 90%, in known attacks included in the KDDCup99 dataset, such as DoS attacks, URL, R2L and probe attacks, while preserving privacy through blockchain mechanisms.

The resource-constrained IoT devices, which are widely used due to the development of 5G and MEC architectures, are highly vulnerable to attacks. As Y. Fan et al. in Fan, Li, Zhan, Cui and Zhang (2020) described, there are three major challenges that need to be addressed so that 5G technology can provide secure and private communication. The first one is focused on the level of difficulty that exists in designing and training a unified intrusion detection model due to how heterogeneous, diverse, and personalized IoT networks are. Secondly, there are many privacy issues which do not allow raw data to be shared. Finally, the amount of data that certain IoT networks produce is small, making it impossible to train an accurate model. To overcome these challenges, the authors proposed an IDS for 5G IoT, Li, Xu and Zhao (2018); Wang, Chen, Song, Guizani, Yu and Du (2018), based on FTL, West, Ventura and Warnick (2007); Zhuang, Qi, Duan, Xi, Zhu, Zhu, Xiong and He (2021); Pan and Yang (2010), called IoTDefender. This framework performs the aggregation using a federated learning mechanism, builds detection models using transfer learning, Wu, Guo and Buckland (2019); Mathew, Mathew, Govind and Mooppan (2017), and employs the averaging fusion method to allow all IoT devices to exchange information and preserve privacy by exploiting the 6LowPAN protocol. As a result, IoTDefender can detect a wide variety of unknown attacks included in the CIC-IDS2017, NSL-KDD, Tavallaee, Bagheri, Lu and Ghorbani (2009), and IoT Datasets, Mirsky, Doitshman, Elovici and Shabtai (2018a), due to its generalization ability. Regarding the evaluation of this framework, it was found that it achieved an accuracy of 92.81%, making it more effective than traditional methods. Finally, an important aspect of IoTDefender is that it produced a lower FP rate than a centralized model, proving its ability to generalize.

Despite the success of implementing FL mechanisms in detecting and identifying malicious traffic patterns in network systems, Y. Sun et al. in Sun, Esaki and Ochiai (2021) noted that no single global model exists which can detect all types of attacks, due to the fact that some networks have dissimilarities in terms of data distribution. For this specific reason, they propose Segmented-Federated Learning (Segmented-FL) which tries to group networks based on specific characteristics (segmentation). More precisely, the main goal of the proposed method is to build multiple global models (one per group) in order to cover as many distinct attacks as possible. The evaluation of the Segmented-FL with the averaging technique as the fusion method was performed on a custom dataset to detect malicious network events in Local Area Networks (LANs) using a variety of metrics such as weighted precision, recall, and F1-score. It was found that it achieved F1-scores up to 96.4%, 80.3%, and 91.2% when it was tested in three different types of intrusion detection tasks, namely server message block (SMB), TCP SYN flood and User Datagram Protocol (UDP) unicast attacks, while exploiting the following protocols: IP, Address Resolution Protocol (ARP), TCP, HTTP, HTTPS, UDP, multicast Domain Name System (mDNS), Dynamic Host Configuration Protocol (DHCP), among others. For each task, Segmented-FL improved the traditional FL system by 0.1%, 4.0%, and 1.1%, respectively.

While MEC can overcome the limitations of cloud computing in order to support IoT systems, careful attention must be paid to network instabilities and vulnerabilities to cyberattacks. For this reason, D. Man et al. in Man, Zeng,

Yang, Yu, Lv and Wang (2021) proposed an Intrusion Detection Federated-based Learning system, called FedACNN, which performs intrusion detection tasks using federated learning with CNNs on IoT, IIoT and MEC architectures, against attacks such as DoS, U2R, R2L and probing, O'Shea and Nash (2015); Albawi et al. (2017); Li et al. (2022); Vinayakumar et al. (2017); Mohammadpour et al. (2018). More specifically, the CNN, implemented with the PyTorch library, mainly consisted of Convolutional layers, Pooling layers, and a fully connected layer. Also, the Rectified Linear Unit (ReLU) activation function is used in order to increase the convergence speed during training. In addition, the communication delay is reduced using an attention mechanism. Regarding the performance of FedACNN, it was found that it outperformed traditional ML methods in terms of accuracy, reaching 99.76% on the NSL-KDD dataset, with a reduction of half in the number of required communication rounds.

Nowadays, patterns of cyberattacks tend to change frequently, making them more unpredictable. While centralized ML models improve the detection of attacks, they face some security and privacy issues, and, simultaneously, they do not generalize well enough to identify a variety of distinct types of attacks. On this rationale, Y. Sun et al. in Sun, Ochiai and Esaki (2020) proposed a Segmented federated learning system, which does not train a single global model, yet it holds multiple global models, each referring to a group of similar attacks on the LAN. The segmentation of participant networks is performed dynamically, while the training of each segment's model takes place using only the segment's participant. Furthermore, in order to increase the adaptability of the system, each global model interacts and communicates with all the other global models to update its parameters, while it exploits the averaging fusion technique. Additionally, the protocol information that is utilized in this study includes protocols such as ARP, IP, TCP, UDP, HTTPS, HTTPS, mDNS, DHCP, among others. The proposed method attained accuracies of 92.3%, 81.3%, and 87.7%, when employing a CNN on a custom dataset.

Due to the increased importance of network security, an accurate IDS has become an essential component of every modern network, with its main goal being the detection and identification of malicious attacks. Y. Cheng et al. in Cheng, Lu, Niyato, Lyu, Kang and Zhu (2022) proposed an FTL system that uses extreme learning to increase its performance in terms of identifying attacks in MEC architectures. FLTrELM, as it is called, initially builds a model through extreme learning to generate additional training samples, due to the insufficient number of samples in the original dataset, and then implements the federated learning mechanism to let the model learn how to preserve data privacy during the training phase. Finally, an intrusion detection model is produced by exploiting the PySyft and PyTorch libraries. Regarding the evaluation of FLTrELM, experiments on popular datasets, like NSL-KDD and UNSW-NB15, Moustafa and Slay (2015), proved that the proposed framework achieved high accuracy (73%) in predicting attacks, especially in cases where the number of samples was restricted, as well as when new types of attacks were introduced, while, simultaneously, protecting data privacy. To be more specific, their implementation proved its efficiency in detecting numerous attacks such as DoS, U2R, R2L, probing, fuzzers, analysis, backdoor, exploits, generic, reconnaissance, shellcode and worms attacks.

Even though modern Federated Learning systems using DNN are successful in detecting and identifying cyberattacks and intrusions, G. Shingi et al. in Shingi, Saglani and Jain (2021) note that due to the different nature of each network's data, a single global model cannot fit all cases. For this reason, they propose a Segmented-FL framework, in which similar networks are grouped by periodically evaluating local models. The global models aggregate the local models' parameters using a weighted average algorithm based on the size of the dataset each network holds. In terms of architecture, local models consist of an NN with 3 layers; an input layer, a hidden layer, and the output layer, while the federated learning mechanisms are implemented using a server which stores all global models (one global model per group). Regarding the evaluation of Segmented-FL, it was found that the proposed method outperformed both the centralized and the traditional FL approach, producing an F1-score of 92% in predicting attackers, and 92% in predicting victims, using the CIDDS-001, Ring, Wunderlich, Grüdl, Landes and Hotho (2017b), and CIDDS-002, Ring, Wunderlich, Grüdl, Landes and Hotho (2017a), datasets and encompassing DoS, brute force and PortScan attacks, while exploiting the Internet Control Message Protocol (ICMP), IP, TCP and UDP protocols.

In order to address the heterogeneity of networks, S. I. Popoola et al. in Popoola, Gui, Adebisi, Hammoudeh and Gacanin (2021) proposed an FDL system, in which each node trains a DNN using local network traffic data from IoT, IIoT and IoV devices. In addition, there exists a dedicated server that receives every model's resulting parameters, aggregates them using the Fed+ fusion technique, and then broadcasts them to every node, Kundu et al. (2022). In terms of DNN architecture, the models consist of an input layer, two fully connected hidden layers, and an output layer. Simulation results of the proposed system attained an accuracy of 99.27%, a precision of 97.03%, a recall of 98.06%, and an F1-score of 97.50%, proving the superiority of FDL to local DNN models, while employing the NF-TON-IoT-v2, Sarhan, Layeghy, Moustafa and Portmann (2021c), NF-UNSW-NB15-v2, Sarhan et al. (2021c), and NF-CSE-CIC-IDS2018-v2, Sarhan et al. (2021c), datasets. Also, in order to choose the best fusion technique, they ran the same experiments using FedAvg, Fed+, and Coordinate Median (CM). The results indicated that Fed+ outperformed the other two SOTA fusion techniques, making the DNN-Fed+ the preferable way to detect intrusions (backdoor, DoS, DDoS, SQL injection, MITM, password, ransomware, scanning and XSS) in heterogeneous wireless networks for a variety of protocols such as IP, ARP, TCP, HTTPS, HTTPS, UDP, mDNS, DHCP, among others.

Modern NNs that are used for the detection of network intrusions and attacks depend on the quality and the quantity of the data, while their interpretation is not clear. T. Dong et al. in Dong, Li, Qiu and Lu (2022) proposed FedForest, a novel learning-based IDS which employs Gradient Boosting Decision Trees (GBDTs) and FL mechanisms, Friedman (2001). More precisely, each client trains a local encoder (GBDT classifier) using their private data and sends the resulting parameters to the server. Then, the server chooses the most appropriate encoders and sends them to all clients. Subsequently, the clients encode their data using the encoders sent by the server, and the final step is the training and deployment of the models. In addition, a random data masking algorithm is implemented in order to preserve data privacy. The attacks that this implementation endeavored to identify were DoS and DDoS, while the communication protocols that were exploited were IP, TCP, HTTPS and DNS. Regarding the evaluation of FedForest, it was compared with a Multi-Layer Perceptron (MLP) with 3, 5, and 7 layers, and it was found that it achieved higher accuracy in all four experiments conducted, achieving an accuracy of 67.03% on the CIC-DDos2019, Sharafaldin, Lashkari, Hakak and Ghorbani (2019), dataset, 89.63% on the MalDroid2020 Mahdavifar, Kadir, Fatemi, Alhadidi and Ghorbani (2020), 86.76% on the Darknet2020, Hristov, Nenova, Iliev and Avresky (2021), and 79.63% on the DoHBrw2020, MontazeriShatoori, Davidson, Kaur and Lashkari (2020a), dataset.

T. Markovic et al. in Markovic, Leon, Buffoni and Punnekkat (2022) proposed an RF algorithm for identifying and detecting attacks in an FL environment in order to avoid data leakage and keep the data of each network private, Resende and Drummond (2018); Farnaaz and Jabbar (2016). Regarding the FL aspect, each client trains an RF, Zhang, Zulkernine and Haque (2008); Zhang and Zulkernine (2006), locally and transmits the resulting parameters for aggregation to the server. In terms of performance, the proposed method was evaluated on four intrusion detection datasets (KDD, NSL-KDD, UNSW-NB15, Moustafa and Slay (2015, 2016); Moustafa, Slay and Creech (2017b); Moustafa, Creech and Slay (2017a); Sarhan, Layeghy, Moustafa and Portmann (2021b), and CIC-IDS-2017, Sharafaldin, Lashkari and Ghorbani (2018c)), regarding several kinds of attacks and utilizing various communication protocols. The results indicated that the global RF, which was collaboratively trained, on the server produced higher accuracy than the max accuracy that the individual RFs on clients managed to achieve in most of the datasets, achieving 71.82% in the IDS2017 dataset.

Nowadays, the increased popularity of IoT devices in people's everyday lives revealed their vulnerability to intrusion attacks the main reasons being their design, implementation, and configuration. As a result, there is a high chance a network has vulnerable IoT devices which can compromise sensitive data. Based on T. D. Nguyen et al. in Nguyen, Marchal, Miettinen, Fereidooni, Asokan and Sadeghi (2019), existing IDSs cannot detect IoT devices that compromise the whole network due to the different fundamentals on which IoT devices have been built. In their paper, they propose DÏoT, an autonomous self-learning distributed system for the detection of compromised IoT devices, Mohammadi and Amiri (2019); Shone, Ngoc, Phai and Shi (2018). More precisely, DÏoT tries to identify communication anomalies based on communication profiles that have been built internally on each device, such as IP, TCP and WiFi protocols. The process of building a communication profile does not require any human involvement or data to be labelled. Furthermore, in order to increase the efficiency of communication profiles and the accuracy in detecting anomalies, T. D. Nguyen et al. (2019) implemented a federated learning approach encompassing the FedAvg fusion technique. In terms of architecture, DÏoT uses GRU models for anomaly detection and Flask for the implementation of federated learning, Grinberg (2018a); Copperwaite and Leifer (2015). Regarding the performance of DÏoT, it was evaluated using 30 IoT devices compromised by the famous Mirai malware, Antonakakis, April, Bailey, Bernhard, Bursztein, Cochran, Durumeric, Halderman, Invernizzi, Kallitsis et al. (2017), including pre-infection, infection, scanning and DoS attacks, and it was found that the proposed system detected on average 95.6% of compromised IoT devices in 257ms, Antonakakis et al. (2017). Finally, it is worth noting that DÏoT did not report any false alarms.

In order to increase the agricultural-IoT infrastructures, O. Friha et al. in Friha, Ferrag, Shu, Maglaras, Choo and Nafaa (2022) proposed FELIDS, targeting not only IoT, but MEC, SDN and Cyber-Physical Production Systems (CPPS) systems. FELIDS is a federated learning-based IDS that preserves data privacy and security by training models locally while increasing the detection rate by aggregating the knowledge which was gained by training the local models of all participating devices, resulting in a global model with improved detection capabilities. In terms of architecture, the proposed system implements a CNN, which consists of pooling, and fully connected layers, for the pre-processing of the data, and a Recurrent Neural Network (RNN), like LSTM for processing input sequences, Hopfield (1982); Yin, Zhu, Fei and He (2017). For this implementation, the TensorFlow and Sherpa.AI frameworks were utilized, while enclosing the FedAvg fusion technique, towards addressing numerous kinds of attacks and exploiting the IP, TCP, HTTP, Secure Shell (SSH) and MQTT protocols. The evaluation of FELIDS took place using CSE-CIC-IDS2018, Sharafaldin et al. (2018b), MQTTset, Vaccari, Chiola, Aiello, Mongelli and Cambiaso (2020), and InSDN datasets, Elsayed, Le-Khac and Jurcut (2020), and the results indicated that the proposed method outperformed the classic centralized methods in detecting attacks, while achieving approximately 94% and 99% accuracy on the IDS2018 and the InSDN datasets, respectively, along with maintaining data privacy.

The increased popularity of IoT networks brought an increase in the number of intrusion attacks aimed at Medical environments, with their main goal being the access to confidential data and the disruption of services. In order to mitigate this danger, I. Siniosoglou et al. in Siniosoglou, Sarigiannidis, Argyriou, Lagkas, Goudos and Poveda (2021) proposed a Federated Layered Architecture for the Medical

Cyber-Physical Systems (MCPS) Networks which was used for increasing the security of the network during the model's training phase by implementing several aggregation layers, while employing the FedAvg fusion technique, Lee, Sokolsky, Chen, Hatcliff, Jee, Kim, King, Mullen-Fortino, Park, Roederer et al. (2011); Lee and Sokolsky (2010); Dey, Ashour, Shi, Fong and Tavares (2018). In terms of architecture, two Deep Generative Adversarial Networks (GANs) are implemented. More precisely, the Generator of each GAN consists of a Dense layer with the hyperbolic tangent (tanh) activation function, another Dense layer with a batch normalization layer and the ReLU activation function, a third Dense Layer, and a final Dense layer with the tanh activation function. The Discriminator of each GAN consisted of a Dense layer with the LeakyReLU activation function, another Dense layer with a batch normalization layer as well as the ReLU activation function, and a final Dense layer. The target systems of the authors' implementation were IoT and MCPS while exploiting IP, TCP, ICMP, UDP among other communication protocols. Regarding the evaluation of the proposed method, it was found that the detection rate increased compared to the commonly trained models, reaching 78.37% accuracy on the CHARIS, Kim, Krasner, Kosinski, Wininger, Qadri, Kappus, Danish and Craelius (2016), and UNSW-NB15 datasets, in an effort to identify fuzzers, analysis, backdoor, DoS, exploit, generic, reconnaissance, shellcode and worms attacks.

Due to the fact that it is challenging for Smart Grid (SG) architectures to provide secure and resilient systems, developers are implementing ML algorithms for the detection of intrusions by monitoring the traffic flow of the network, Fang, Misra, Xue and Yang (2011); Ma, Chen, Huang and Meng (2013); Tuballa and Abundo (2016). While this approach may increase the detection rate of attacks, it threatens consumers' privacy because it needs access to user data to train such models. P. H. Mirzaee et al. in Mirzaee, Shojafar, Pooranian, Asefy, Cruickshank and Tafazolli (2021) proposed an FIDS architecture in a 5G environment with the main goal being the preservation of users' privacy while keeping the detection rate high. Towards achieving this, the protocols that they exploited included IP, TCP, ICMP, UDP, SMTP, SSH, HTTPS, and FTP, among others, and targeted SG, Advanced Metering Infrastructure (AMI), Demand Response (DR), Real-Time Pricing (RTP) and Smart Manufacturing (SM) systems. More specifically, they designed a Federated Deep Neural Network (FDNN) model that keeps users' information private and a server that aggregates the updated local models and broadcasts the produced model back to the network devices, while utilizing the FedAvg fusion method. In terms of evaluation, the proposed method achieved 99.5% accuracy, 99.5% precision/recall, and 99.5% F1-score when it was evaluated on the NSL-KDD dataset, against attacks such as DoS, probing, R2L and U2R.

Even though ML methods produce a high detection rate in identifying attacks in an IoT environment, their need for labelled data to train a model is challenging due to privacy reasons. In order to address this challenge, K. Yadav et al. in Yadav, Gupta, Hsu and Chui (2021) designed and proposed an unsupervised deep learning system that implements AEs to learn from unlabelled data, Choi, Kim, Lee and Kim (2019); Song, Hyun and Cheong (2021b). In addition, they implemented an FL approach in order to let different IoT devices train their models locally without the need to share their private data with a server. More precisely, the proposed method consists of a global server model with random initial weights which is distributed to all edge devices. Then, each device trains a model copy locally and sends its updates to the server for averaging (the FedAvg fusion technique is employed). The averaging result is sent back to edge devices again for the next round of training. In terms of architecture, edge devices use an AE to label the unlabelled data and an NN for detecting intrusions, Kingma and Welling (2013b). For the evaluation, the CIC-IDS2017 dataset was used, including various attack types and while exploiting distinct communication protocols such as IP, TCP, ICMP, UDP, SMTP, SSH, HTTP, FTP, among others. The proposed method achieved an accuracy of 97.75% in detecting intrusions.

With regard to IoV networks, an increased number of communication interfaces exist, making the whole network vulnerable to intrusion attacks. These attacks can take control of a vehicle remotely, and, also, invade any neighboring vehicle that is part of the same IoV network. In order to increase the protection of vehicles in IoV networks, T. Yu et al. in Yu, Hua, Wang, Yang and Hu (2022) proposed a federated LSTM NN-based IDS, Lin, Clark, Birke, Schönborn, Trigoni and Roberts (2020). More precisely, based on the message sequence of the In-Vehicle Network (IVN), an LSTM NN will be used to identify any intrusions in the incoming messages. However, other target systems are employed as well, including Intelligent Connected Vehicle (ICV), Electronic Control Unit (ECU) and On-Board Unit (OBU). Furthermore, a federated learning approach is implemented to increase the security and the efficiency of the LSTM NN training, with interconnected vehicles working as clients that train a model locally, and base stations that work as servers for the aggregation of clients' resulting models, encompassing the FedAvg fusion method. Regarding the architecture of the LSTM NN, it consists of 6 layers using tanh as the activation function in all layers except for the final layer, where the activation function employed was softmax. In terms of performance, it was found that the proposed method attained an accuracy of over 90% on the OTIDS, Lee, Jeong and Kim (2017), dataset, towards detecting DoS, spoofing, replay and drop attacks.

AMI systems play an important role in SG architecture, but they are exposed to cyberattacks, Mohassel, Fung, Mohammadi and Raahemifar (2014a,b). The current methods of intrusion detection in AMI systems require gathering all data in a single node or a data center, making this approach almost infeasible due to privacy and security reasons. To address this problem, H. Liang et al. in Liang, Liu, Zeng and Ye (2022) proposed a federated learning-based intrusion detection framework for AMI systems. In this method, the training does not take place in a computing node, or a data center, but it is performed in the data concentrators, and only

the resulting weights of each concentrator's local model have to be sent to the data center. Furthermore, the data center aggregates the resulting weights of the concentrators' trained models, in order to increase the detection accuracy in a collaborative learning manner, while employing the FedAvg fusion technique. In terms of architecture, a DNN was implemented, consisting of an input linear layer with the ReLU activation function, 3 linear layers with the ReLU activation function, a dropout layer in the hidden layer and an output linear layer with softmax as the activation function. Regarding the results, the proposed method produced 99.32% accuracy, higher than the 98.94% accuracy of the centralized approach on the NSL-KDD dataset, including the identification of DoS, probe, R2L and U2R attacks, while exploiting a variety of communication protocols. Additionally, the proposed method, which was implemented using the PyTorch library, reduced computation and communication costs, while preserving data privacy.

R. Zhao et al. in Zhao, Wang, Xue, Ohtsuki, Adebisi and Gui (2022) tried to address three important challenges that influence the performance of IDSs in an FL framework. More precisely, as it is noted, the first challenge involves the security of data due to the fact that private data can be extracted from the transmitted parameters, while the second challenge focuses on non-IID data and the level of effect this has on federated training. The third challenge is relevant to the communication overhead caused by the large size of DNN models which makes the actual deployment difficult. In order to address these limitations, they designed an IDS in a semi-supervised FL framework using knowledge distillation, during which the PyTorch library was utilized, Zhu and Goldberg (2009); Chen, Gong and Tian (2008). Initially, the proposed method utilized unlabelled data by implementing distillation methods in order to increase the classifier's accuracy. Then, a CNN, which consists of an input layer, 4 convolutional layers, a fully connected linear layer, and an output layer, is implemented with its main goal being the extraction of features from the network traffic. Finally, they designed a discriminator to improve each client's predictions on intrusions, and, simultaneously, to avoid any failures caused by non-IID data. To reduce the communication overhead even further, they implemented a hard-label strategy and voting mechanisms. The systems that were targeted during this implementation included IoT devices, while utilizing the IP, TCP, ICMP and UDP protocols, among others. Regarding the evaluation of the proposed method, it outperformed SOTA methods on real-world traffic dataset, namely the N-BaIoT, Meidan, Bohadana, Mathov, Mirsky, Shabtai, Breitenbacher and Elovici (2018), dataset, while achieving distinct accuracy levels with three non-IID scenarios.

Attacks and intrusions on MCPSs can lead to data leakage of very sensitive and private information on patients and hospitals, as mentioned before. Also, the level of heterogeneity of devices participating in an MCPS network makes the network vulnerable to a variety of attacks. In order to address these security issues, W. Schneble and G. Thamilarasu in Schneble and Thamilarasu (2019) designed a massively distributed ML IDS in a federated learning scheme for MCPS, while encompassing the FedAvg fusion technique. As they state, they used an FL approach to decrease communication and computation costs and increase the security of the network. The proposed method was implemented using the scikit-learn library, and was evaluated using real-world data and attacks such as DoS, Data Modification, and Data Injection. Simultaneously, the targeting system was MCPS and the communication protocols that were utilized were Bluetooth, Zighee and 802.11. The results indicated that the proposed model achieved an accuracy of 99% and an FP Rate of 1% on the MIMIC dataset, while reducing communication costs.

Even though DL methods can accurately predict different types of cyberattacks, their need to gather all the data in a centralized entity is raising security and privacy issues while increasing communication costs and latency. Moreover, it is inefficient and requires many resources to label all the data generated in IoT devices due to their volume. In order to address these issues, O. Aouedi et al. in Aouedi, Piamrat, Muller and Singh (2022a) proposed a semi-supervised learning approach, for IoT and IIoT systems, in a federated learning framework, which benefits from both labelled and unlabelled data the fusion method of which was chosen to be FedAvg. More precisely, an AE is implemented on every device in the network to discover and extract low-dimensional features using only local data. Then, a server receives the parameters of all AE from the devices and aggregates them into a global AE. Finally, the server builds a supervised NN, by adding a dense layer to the global AE and trains it using labelled data which are publicly available, namely the Gas pipeline SCADA, Morris and Gao (2014a), dataset. For this implementation, the widely utilized library PyTorch was utilized. In terms of performance, the results indicated that the proposed method preserved data privacy and identified 95.84% of the attacks, while reducing communication overhead by 50% for the Modbus, Simple Network Management Protocol (SNMP) and C37.118 communication protocols.

New requirements regarding the reliability and security of the network domain are the result of the current digital transformation of the world. Even though ML algorithms can successfully detect intrusions in a network, there are concerns about the generalization ability of these approaches to detect attacks between different contexts. In order to address this issue, G. d. C. Bertoli et al. in Bertoli, Junior, Santos and Saotome (2022) proposed a stacked-unsupervised FL framework with the main goal being the generalization in detection intrusions in a cross-silo configuration. In terms of the architecture, it uses a Deep AutoEncoder combined with an energy flow classifier, while for the federated learning configuration, they used a server for the aggregation of the resulting parameters of each silo using FedAvg, FedOpt, Asad et al. (2020a), and FedAvgM as fusion techniques. Regarding the evaluation of the proposed method, the authors used a variety of datasets, namely UNSW-NB15, CSE-CIC-IDS-2018, Sharafaldin et al. (2018c), Bot-IoT, Koroniotis, Moustafa, Sitnikova and Turnbull (2019); Koroniotis,

Moustafa, Sitnikova and Slay (2018); Koroniotis, Moustafa and Sitnikova (2020b); Koroniotis and Moustafa (2020); Koroniotis, Moustafa, Schiliro, Gauravaram and Janicke (2020a); Koroniotis (2020), and TON_IoT dataset. The results indicated that it outperformed the traditional local learning and cross-evaluation methods, achieving an accuracy of 97%, 98%, 93%, and 74%, respectively.

Due to the rapid growth in the amount of network data, IDSs have lost part of their efficiency and accuracy in predicting attacks. Moreover, most of the network data is currently generated in mobile phones, smart devices, and wearables, while their privacy is important as they store sensitive information concerning individuals. In order to increase the detection accuracy, and, simultaneously, preserve the privacy of data, J. Shi et al. in Shi, Ge, Liu, Yan and Li (2021) proposed a Federated Learning-based IDS. In terms of architecture, they used a CNN which consists of an Input Layer, 2 Convolutional Layers, a MaxPooling Layer, a Flatten Layer, a Dense Layer, a Dropout Layer and, finally, an Output Layer. Regarding the evaluation of the proposed method, they conducted experiments using the UNSW-NB15 dataset, which contains 9 different attack types and 49-dimensional feature data, and the CSE-CIC-IDS2018 dataset which contains 80 features per attack scenario. The results indicated that the proposed method achieved a lower accuracy of 81.19% compared to the 83.46% of a centralized CNN for the first dataset, and 78.46% compared to 98.77% for the second dataset while preserving the privacy of the data.

O. Aouedi et al. in Aouedi, Piamrat, Muller and Singh (2022b) proposed an FL with a semi-supervised approach for IDSs, named FLUIDS. More precisely, their work consists of a number of devices that train locally an AE using unlabelled data in order to find representations of low-dimensional features (unsupervised learning) and the resulting parameters are sent to an FL server for aggregation, the fusion method of which was FedAvg, using the PyTorch library. In addition, the server does not only build a global AE but also uses an amount of labelled data to perform supervised learning using a Fully Connected Neural Network (FCNN) for the classification of the attacks (supervised learning). Finally, the server sends back the updated global AE, and the supervised model to perform the IDS task to the devices. Their implementation was tested on IoT devices while exploiting IP, TCP, ICMP, and UDP protocols, among others. For the evaluation of FLUIDS, experiments were conducted using the UNSW-NB15 dataset, which consists of 175,341 training and 82,332 testing samples. The results showed that when FLUIDS was combined with MLP, RF, SVM, and DT classifiers, the F1-score increased by 3.68%, 5.46%, 6.21%, and 7.55% respectively, compared to when these classifiers were employed on their own, reaching an F1 score ranging among 80% and 90%.

Although DNNs achieve a high efficiency in cybersecurity monitoring, in a resource-constrained environment, like the IoT, DNNs are almost impossible to be trained due to the high computational resources they need. More precisely, in an FL environment, the devices are forced to train a computationally heavy model in order to keep their data private, making it eminently hard for the device both in terms of time and accuracy. I. Zakariyya et al. in Zakariyya, Kalutarage and Al-Kadri (2021) proposed a memory-efficient method of training an FCNN for IoT to detect intrusions and attacks such as BASHLITE, Mirai and DDoS incidents, in FL settings, while employing the FedAvg fusion technique. Additionally, the communication protocols onto which they tested their implementation included Bluetooth, Zighee, XBee and 6LoWPAN. In terms of architecture, the proposed FCNN consists of an input layer, three hidden layers, and an output layer, implemented using PySyft and PyTorch libraries. The number of neurons varied in every experiment that was conducted. Specifically, eleven experiments were performed using eleven distinct datasets including the N-BaIoT, the Kitsume, IoT-DDoS and WUSTL, Teixeira, Salman, Zolanvari, Jain, Meskin and Samaka (2018) datasets. The results indicated that the proposed method may decrease memory requirements by up to 99.46% while maintaining the same accuracy - reaching a maximum level of 97% - as well as the F1-score.

SGs are becoming a necessity due to the increasing power demands. Advanced networking capabilities, which are introduced with the 5G networks, can enable smart meters in the AMI of the SG core. However, these networks are vulnerable to a wide range of cyberattacks. Towards their protection, a transformer-based hierarchical Federated Learning-based Intrusion Detection System (FL-IDS) is proposed in Sun, Tang, Du, Deng, Lin, Chen, Qi and Zheng (2022) by X. Sun et al. which is able to preserve client data protection and privacy while simultaneously reducing communication costs in the IP, TCP, ICMP and UDP protocols, among others, Han, Xiao, Wu, Guo, Xu and Wang (2021). The proposed Transformer Intrusion Detection Model consists of a feature extraction layer, a column embedding layer, a stack of N Transformer layers, and an MLP, using a custom fusion technique, implemented with TensorFlow. The model's performance was evaluated on the NSL-KDD dataset, using two feature extraction layers and two transformer layers while reaching an accuracy of 99% in detecting DoS, Probe, R2L and U2R attacks. Through the evaluation, it was shown that the proposed Transformer-IDM model can act as an IDS in FL while maintaining only a small number of NN parameters reducing the communication cost in FL.

In an IoT environment, FL for IDS can reduce the required network resources and bandwidth needs, while simultaneously maintaining a higher battery charge for all IoT devices. The authors in Saadat, Aboumadi, Mohamed, Erbad and Guizani (2021) explore and demonstrate the superiority of Hierarchical FL over classic FL, alongside its advantages in the case of an IoT environment with an edge infrastructure and for the ZigBee, Bluetooth and RFID communication protocols. They constructed an NN with 122 neurons for the input, 80 neurons for the first hidden layer, 40 neurons for the second hidden layer, and 5 neurons for the output layer, which they evaluated on the NSL-KDD dataset, reaching approximately 78% accuracy in detecting DoS, Probe, R2L and U2R attacks. More specifically, they

tested a scenario with 8 edge clients which in the HFL scenario were organized in two edge clusters with 4 clients per cluster, where each cluster uses FedAvg, and the data between different clients was non-IID. The HFL model proved to have faster convergence and overall better training statistics, with the edge layer possibly absorbing some of the effects of the non-IID of data before the aggregated models are forwarded to the centralized cloud.

In a similar manner to classic ML, the features chosen for training the model are of utmost importance for the whole ML pipeline. The authors in Qin and Kondo (2021) have demonstrated that the feature selection can significantly increase the detection accuracy of an IDS, by differentiating the feature selection between different attack types, Li, Cheng, Wang, Morstatter, Trevino, Tang and Liu (2017); Kira and Rendell (1992); Di Mauro, Galatro, Fortino and Liotta (2021); Alazab, Hobbs, Abawajy and Alazab (2012). They propose a greedy feature selection algorithm, which they evaluate on the NSL-KDD dataset using the ONLAD NN, Tsukada, Kondo and Matsutani (2020), having a significantly different output when the input is an anomaly. The proposed algorithm referred to IoT environments, implementing the FedAvg fusion method. To deal with decreasing accuracy in detecting attacks such as DoS, Probe, R2L and U2R attacks, multiple global detection models are trained using FL. The evaluation showcased an accuracy increase from 2.2% up to 29%, reaching approximately 70%.

The IoT-based Transactive Energy System (IoTES) which enables innovative services with independent distributed systems in SGs is often susceptible to False Data Injection Attacks (FDIA), which not only are hard to detect in IoTES but can also lead to privacy violations during detection. To cope with these problems, a decentralized-based FDIA detection model has been proposed in Tahir, Jolfaei and Tariq (2021), which is based on deep federated learning using attentive aggregation (DeepFed-AA) with a GRU module, while using Differential Privacy through randomization before sending the client models to the centralized server. The attentive aggregation process considers the significance of each client while optimizing the central model during aggregation. The model made use of the PyTorch and MATPOWER libraries and was evaluated on the National Renewable Energy Laboratory dataset while outperforming all other SOTA models, achieving 96% detection accuracy on a large-scale model, requiring less training and detection time.

All NN-based NIDS suffer from the need for variety and diversity in data as well as the lack of interpretability, while some of them are also limited regarding multi-class attack classification. To overcome these limitations, a GBDT has been proposed in Dong, Qiu, Lu, Qiu and Fan (2021) as a NIDS, which is based on DTs but generates a distinct prediction score for each class and classifies a sample to the class with the highest prediction score. Towards training this classifier, partial data masking is used for client privacy, then the server decides upon small bins of predefined width based on the unique values of client data, and finally, clients transform their data into bin numbers which are sent to the server for model training. This approach was evaluated on the CIC-DDoS2019 DDoS attack dataset and was compared to a 5-layer MLP using FedAvg and to a centralized GBDT model trained on the full data, taking advantage of the IP, TCP, HTTPS and DNS communication protocols. The Federated GBDT performed much better than the MLP model but worse than the local GBDT model due to the masking procedure, achieving approximately 65% accuracy.

Federated Learning-based Network Intrusion and Detection Systems (FL-NIDSs) have been shown to be vulnerable to poisoning attacks launched by malicious clients. In these attacks, poisoned traffic is injected into the local training dataset, impairing the NIDS protection capabilities. Current SOTA FL-NIDS first uses model-level defense with an offline intrusion detection model on the server side to detect and reject the models that have been poisoned in the global model aggregation, then the data-level defense is applied, which cleans the data from poisonous traffic. However, the huge number of NN model parameters, combined with the heterogeneous and time-varying nature of IoT traffic data, makes these solutions far from usable. To overcome these restrictions, SecFedNIDS is proposed in Zhang, Zhang, Guo, Yao and Li (2022), replacing the offline detection model with a gradient-based important model parameter selection method for the poisoned model detection, alongside an online unsupervised poisoned model detection method based on SOS. The proposed model, implemented with PyTorch and exploiting the IP, TCP, ICMP, UDP and other communication protocols, was evaluated on the NSW-NB15 and CSE-CIC-IDS2018 datasets and was compared to other SOTA defense mechanisms, like Krum, Blanchard, El Mhamdi, Guerraoui and Stainer (2017), Geomed, Chen, Su and Xu (2017), and a baseline FedAvg FL-NIDS, managing to achieve an overall significant improvement and better detection and defense against most attacks. More specifically, in the task of detecting the label flipping and clean label attacks SecFedNIDS achieved approximately 99% and 95% accuracy levels on the NSW-NB15 and CSE-CIC-IDS2018 dataset, respectively.

Controller area networks (CANs) are usually used for managing in-vehicle communication systems and broadcast packets to their buses, so all nodes and Electronic Control Units (ECU) attached to the bus can receive transmitted packets, Farsi, Ratcliff and Barbosa (1999); HPL (2002); Foster and Koscher (2015). However, packet authentication is impossible, making CANs vulnerable to attacks, like steering and braking, or speedometer display information manipulation. Furthermore, FL-IDS requires a central aggregation server which adds latency and is also vulnerable to a single point of failure. To overcome these limitations, the use of blockchain with Federated Learning (BC-FL) has been proposed in Aliyu, Feliciano, Van Engelenburg, Kim and Lim (2021), in which local models are stored in the blockchain for local and independent aggregation, Zheng, Xie, Dai, Chen and Wang (2018). To address the problem of sharing sensitive CAN data, a Blockchain-based Federated Forest SDN-enabled Intrusion Detection System (BFF-IDS) for an IVN has been proposed, in which every vehicle is treated as a client with

generated data that can be used to train IDS models, Sultana, Chilamkurti, Peng and Alhadad (2019). These models are exchanged using a blockchain approach, implemented with scikit-learn, Ethereum and Mininet frameworks, managed by the SDN, which dynamically routes packets and model exchanges from InterPlanetary File Systems (IPFS) through the blockchain. IPFS is used to upload the model while its location's (unique client ID) hash is exchanged over the blockchain, reducing the network requirements and cost, for which the On Board Diagnostics II (OBD-II) and Bluetooth communication protocols are utilized. This solution was evaluated on the OTIDS dataset, Lee et al. (2017), against other detectors and related IDS, using the scheme of 10-fold cross-validation, displaying a higher classification accuracy than every other competing model reaching 95% in detecting fuzzy, SoS, impersonation and attack-free state attacks.

The increase in network complexity makes networks vulnerable to a variety of attacks, which IDS tries to mitigate. Developing reliable IDS is crucial for defending networks, so adding IDS to FL enables ML to deliver fine-tuned protection mechanisms for various networks and Internet of Medical Things (IoMT) devices, Vishnu, Ramson and Jegan (2020); Thamilarasu, Odesile and Hoang (2020); Zachos, Essop, Mantas, Porfyrakis, Ribeiro and Rodriguez (2021). However, the FL improvements of IDS resulted in more sophisticated attacks, which signifies that the performance of IDSs needs continuous improvements. Furthermore, the question "which IDS is the best one?" is still difficult to answer, considering that there is no agreement on which criteria are the most suitable in evaluating IDS classifiers, as most of the current ones depend on a single incomplete aspect. Authors in Alamleh, Albahri, Zaidan, Albahri, Alamoodi, Zaidan, Qahtan, Alsatar, Al-Samarraay and Jasim (2022) try to standardize and propose a benchmarking framework for IDS towards detecting DDoS attacks using the integration of direct rating, entropy weighing and Vlsekriterijumska Optimizcija I Kaompromisno Resenje (VIKOR) methods, while for the evaluation purposes they employ the NSL-KDD dataset.

It is a common procedure for IoT devices to produce distinct data types or describe the same information with different sets of data attributes. An example of such infrastructure is an industrial facility that consists of multiple collaborating sensors, each one of which is in charge of different steps of a process, where data from all sensors are required simultaneously for the monitoring and analysis of the state of such processes. In literature, this is called vertically partitioned data. While several FL-IDS exist when dealing with horizontally partitioned data, very few exist for dealing with vertically partitioned data. Authors in Novikova, Doynikova and Golubev (2022) propose an FL-IDS for dealing with vertically partitioned data, using the model of the SWaT water treatment facility, which consists of the number of basic processes corresponding to the physical and control components of a water treatment plant, with the SecureBoost model algorithm which implements GBDT and is also preserving privacy through appropriate mechanisms to protect inputs while not limiting the number of clients, Goh, Adepu, Junejo and Mathur (2017). Both technological and network data are collected by the hub of each process, which differs between distinct processes. Due to the differences in the sets of data attributes, each client has the part of the model that corresponds to its data. Hence, to make an assumption on a new input sample, all clients need to be available and work together during the inference process, which requires additional privacy-preserving mechanisms. While the identification accuracy was very high, yielding approximately 97%, the inference time for new input samples was unacceptably high, possibly due to the use of homomorphic encryption and the way it is implemented in the selected framework, namely the PyTorch and TensorFlow.

FL models can be vulnerable to Backdoor attacks (BDA), in which an attacker performs an attack against the model in order to make it produce inaccurate predictions. BDAs exist in image classification and word prediction. The authors in Nguyen, Rieger, Miettinen and Sadeghi (2020) present BDAs on Federated Learning-based Internet of Things Network Intrusion Detection System (FL-IoT-NIDS). The introduced attack makes IoT devices gradually inject malicious traffic, without requiring the attacker to compromise clients. This attack was evaluated on DIoT-Benign, Nguyen et al. (2019), DIoT-Attack, Nguyen et al. (2019), and UNSW-Benign, Sivanathan, Gharakheili, Loi, Radford, Wijenayake, Vishwanath and Sivaraman (2018), three real-world datasets generated by 46 commodity IoT devices, and IoT malware Mirai, which showed that the attacker can successfully launch poison attacks undetected with a poisoned data rate lower than 20%. Moreover, with their approach, an accuracy of 100% was attained in detecting infection, scanning, SYN flood and HTTPS flood, among various others. The authors also propose defense mechanisms, including server-side FL defenses on the aggregator, which is implemented using FedAvg, client-side filtering or poisoned data tolerating, and malicious traffic injection identification and discard.

IoT devices are constantly increasing in number, use cases, and popularity, resulting in a fast production and distribution phase by several different companies. As a result, they differ in the communication protocols and standards they use, which makes them vulnerable to attacks. ML, RL, FL, functional virtualization and blockchain have been proposed as IoT and IIoT security solutions. However, a bridge between IoT healthcare devices and healthcare informatics must be developed, in order for better intrusion detection to be achieved. Towards this goal, the authors in Otoum, Guizani and Mouftah (2021) propose a Federated Reinforcement Learning, Sutton and Barto (2018); Thrun and Littman (2000); Li (2017), structure utilizing the Q-learning, Servin and Kudenko (2005); Lopez-Martin, Carro and Sanchez-Esguevillas (2020b); Watkins (1989), technique to preserve the security and the optimization of the data that is required in all IoT devices in a healthcare based IoT network topology. Their model was evaluated on the CIC-IDS2017 dataset, against an SVM-IDS model, Mukkamala, Janoski and Sung (2002); Tao, Sun and Sun (2018), by exploiting Simulink.

The proposed model accomplished an accuracy of about 98% and a detection rate of about 97%, surpassing the SVM-IDS model in both metrics in the task of detecting DoS, DDoS, PortScan and brute force attacks, among others, when the communication protocols which were utilized were WiFi, Bluetooth and LAN, among others.

Traditional NIDS are not effective enough as cyberattacks become more sophisticated. For example, dictionary and signature matching strategies are not able to effectively detect modern DDoS attacks. Although FL-NIDS have been proven to be effective defense mechanisms, training data are often shared between clients with imbalanced non-IID features, making it hard to identify large-scale joint or distributed intrusion attacks, Zhang, Wang, Sun, Green II and Alam (2011). Traditional FL methods, like FedAvg, lack sensitivity in differentiating among the distributions between sub-datasets. The authors in Li, Zhang, Li, Guo and Li (2021b) propose an FL methodology for defending against DDoS, among other attacks, by identifying data characteristics that can establish an efficient IDS, considering the prototypical features of each worker, to represent how local data spaces are correlated to global data space. With this approach, feature spaces are expanded while data privacy is preserved. The FIDS model that was developed consists of two identical GRU layers followed by two fully connected layers, and was evaluated on the CIC-DDoS2019 against a baseline FedAvg-based model without prototypical features and an LSTM-based model with 2 LSTM layers followed by 2 fully connected layers with prototypical features. Both the FIDS and the LSTM-based models outperformed the baseline model, while the proposed FIDS model achieved an accuracy of 97% in a PyTorch implementation.

In an IoT infrastructure, traffic from multiple devices may not contain the same features. This is where model personalisation comes into play. On the contrary, the dataset labeling process costs a lot both in money and time. Active Learning (AL) solutions have emerged so that the learner can choose the samples to learn from, making it a perfect fit for model personalisation, Settles (2012, 2009); Almgren and Jonsson (2004); Görnitz, Kloft, Rieck and Brefeld (2009). AL is a semi-supervised ML approach that tries to solve the issues of manually adding labels to unlabelled samples, by dynamically selecting samples and making a query at an oracle database to provide the labels. With regard to a combination of FL with AL, the authors in Kelli, Argyriou, Lagkas, Fragulis, Grigoriou and Sarigiannidis (2021) propose an IDS model which initially performs FL global model training and then performs model personalisation using AL, while for the fusion technique the FedAvg method is employed. The developed model is a 6-layer FCNN, which was evaluated on an undisclosed dataset and achieved up to about 85% accuracy on the DNP3 communication protocol.

IoT devices are limited in storage capacity and computing power and cannot improve their local training towards a better FL-IDS, Wang, Zhu, Hei, Kong, Ji and Zhu (2019). Moreover, network limitations forbid uploading a large number of NN parameters to the server. To overcome these limitations, the authors in Hei, Yin, Wang, Ren and Zhu (2020) propose a Blockchained FL cloud IDS (BFL-CIDS), which consists of 4 distinct layers, alongside a Regional Service Party Alert Filter Identification (RSP-AFI). This architecture sets up an RSP to collect the detection and alert sets of IoT devices in its region and filter the false alert information since it accounts for about 90% of the unfiltered alerts which means it can reduce the accuracy of FL models, and then initiate local model training. Its training results are stored on the blockchain, ensuring that they are unaltered and permanently saved. The proposed blockchain model adopts Hyperledger Fabric, which is a permission blockchain scheme in which the Fabric's blockchain network identification and approval are required, while also using erasure code-based low storage, which makes every node require less storage effort, allowing low storage devices to participate in the blockchain so that they contribute to keeping the decentralized characteristic of the blockchain and reduce network load on each node by calculating the product of the number of nodes, Antwi, Adnane, Ahmad, Hussain, ur Rehman and Kerrache (2021). The proposed blockchain solution was evaluated on the DARPA1999, Keogh, Lin and Fu (2005a), dataset against the Ethereum, Zhu, Wang, Hei, Ji and Zhang (2018), blockchain and the proposed RSP-AFI alert solution was evaluated on the KDDCup99 dataset using MLP and DTs against RF and SVM. The RSP-AFI (MLP) is a semi-supervised learning algorithm, which is more suitable for real-world scenarios since it achieves a performance score similar to the supervised learning algorithm although being trained with 20% of the total labelled data. The accuracy levels that were reached for the distinct attacks' identification, namely the DoS, probe, R2L and U2R attacks, reached from about 80% to 97% based on the scenario tested. On the other hand, the Fabric blockchain achieved a much smaller sample uploading time and a much higher number of transactions per second.

Modern industrial control systems (ICS) are equipped with IIoT devices to improve the facilities' functionalities. However, this growth in the number of IIoT devices exposes ICSs to several cybersecurity threats, since all ICSs are connected to the Internet. Thus, protecting IIoT-based ICSs against threats that are becoming increasingly more complex requires an IDS which can be both effective enough and light on resources to run on IIoT devices that have limited computing power and resources. Towards this goal, the authors in Huong, Bac, Long, Luong, Dan, Thang, Tran et al. (2021) propose an SM, Davis, Edgar, Graybill, Korambath, Schott, Swink, Wang and Wetzel (2015); Ren, Wu, Zhang, Terpenny and Liu (2017), architecture which performs the anomaly detection task at the edge using a hybrid model of Variational AutoEncoder (VAE), Li, Cheng, Wang, Liu and Chen (2020a), and LSTM NN using time-series data, and then employs FL to only transmit the trained model of every edge client to the centralized cloud for the server model aggregation. VAE helps with capturing the structural characteristics of the time series over time windows, whereas LSTM estimates how the long-term in the time series is correlated to the features inferred by VAE. This makes

this model able to detect new anomalies over multiple time scales even if they have never occurred before. The hybrid model is built with an ideal threshold using Kernel Quantile Estimator (KQE), Sheather and Marron (1990), to achieve a high detection accuracy and was evaluated against a competing VAE-LSTM solution with heuristics proposed by Lin et al., Lin et al. (2020), using the SCADA systems, Turnipseed (2015), and other time-series data sets collected in several different fields such as ECGs, Keogh et al. (2005a), respiration data, power demand, gesture and space shuttle, and NYC taxi, Tlc (2017). For the designated architecture, Tensorflow and FedML were utilized, while FedAvg was employed as the fusion technique when exploiting the MQTT protocol. Additionally, the proposed solution achieved better Precision and F1-score than the competing model even when the competing model was trained in a completely centralized manner.

The increasing number of connected IoT smart devices offers innovative smart system infrastructures like smart homes, cities, etc. However, to keep such infrastructures safe and functional, adaptive threat and malicious activity detectors must be implemented. FL is suitable for developing such systems since it combines the advantages of modern ML and DL technologies with privacy protection techniques. The authors in Tian, Chen, Yu and Liao (2021) propose the Delay Compensated Adam (DC-Adam) for distributed anomaly detection with DL for IoT systems as well as for CPS, in which the cloud server will aggregate partial weights updates as it receives them from each local client independently, global parameters are initialized to guarantee convergence and the post-training process is appended to each client independently. Each client implements a five-layer Deep Autoencoder (DAE) DNN model. The proposed architecture was evaluated on the MNIST dataset, Deng (2012), the CIC-IDS2017, and the IoT-23 dataset, Garcia, Parmisano and Erquiaga (2020), against three baseline models - variations of DC-Adam - Asynchronous Adam (Asyn-Adam), Asynchronous SGD (Asyn-SGD) and Synchronous Adam (Syn-Adam). DC-Adam achieved convergence and overall higher Accuracy, Precision, Recall, and F1-Score compared to all other baseline models. More specifically, it attained 91% accuracy and 92% F1 score on the MNIST dataset, 88% accuracy and 93% F1 score on the IDS2017 dataset and 90% accuracy and 90% F1 score on the IoT-23 dataset. The attacks that it attempted to identify were brute force, DDoS and Web-based, while the communication protocols that it exploited were WiFi, Bluetooth and LAN, among others.

While data collection and analysis in IoT networks is essential in building innovative operations and services, FL models in some occurrences exhibit lower performance compared to centralized ML models due to the limited available data on every participating worker device and device heterogeneity. For example, Anomaly Detection performance degrades compared to centralized ML using the TON_IoT dataset. To solve this issue, the authors in Weinger, Kim, Sim, Nakashima, Moustafa and Wu (2022) propose the use of data augmentation for dataset rebalancing, experimenting with various techniques like random sampling, Synthetic Minority Over-sampling Technique (SMOTE), Chawla, Bowyer, Hall and Kegelmeyer (2002), and Adaptive Synthetic Sampling (ADASYN), He, Bai, Garcia and Li (2008). The proposed methods were evaluated on the Modbus, Frazão, Abreu, Cruz, Araújo and Simões (2019), dataset against a centralized ML baseline model, towards identifying PortScan, DoS and backdoor attacks. Although the FL model showcases lower performance (78% - 95% accuracy with the implementation of a custom fusion technique) and higher training time compared to the centralized model, dataset rebalancing through random oversampling significantly improves detection performance with manageable complexity when training over a large number of workers, while it also converges within significantly fewer rounds compared to the centralized model.

IoT applications also include the Maritime Transportation System (MTS), through which information exchange offers improvements in maritime transportation like intelligent navigation, avoiding obstacles, traffic monitoring, and vessel collision. However, such critical operations networks must be protected against cybersecurity threats using IDSs. FL has been proposed for developing adaptive IDS while maintaining data privacy. However, due to the lack of stability in the communication environment in the ocean and hardware heterogeneity, FL workers occasionally may not be able to upload their locally trained model parameters on time for aggregation. This is called "the straggler problem" and results in higher model variance. To deal with this problem, as well as with similar instances in Automatic Identification Systems (AIS) and IoT in general, the authors in Liu, Xu, Wu, Qi, Jolfaei, Ding and Khosravi (2022) propose a CNN-MLP-based IDS, in which the CNN performs feature extraction from the data and the MLP implements the actual classification. The proposed architecture is implemented using TensorFlow Federated to detecting DoS, backdoor and various other attacks, while utilizing the WiFi communication protocol. The model is trained through an adaptive batch federated aggregation, named FedBatch, which adjusts the reservation of the global model dynamically. The model was evaluated on the NSL-KDD dataset against an MLP, a CNN and a Bidirectional Gated Recurrent Unit (BGRU) model and against CNN-MLP with FedAvg. CNN-MLP had comparable and sometimes better performance than the other NN models, while FedBatch showcased a higher and more stable accuracy than FedAvg with non-IID client data distribution as well as faster convergence. Eventually testing the derived model yielded an accuracy among 83% and 94%.

In most FL-NIDS scenarios, each client is treated as an independent worker with its own data and hardware capabilities, only responsible for training its local model and uploading its parameters to some centralized node for aggregation. However, the authors in Sarhan, Layeghy, Moustafa and Portmann (2021a) propose a different FL approach, where each local client is observed as a single entity with a unique network of highly heterogeneous endpoints, thus allowing multiple organizations to share Cyber-Threat Intelligence (CTI) to work together in order to design and

build an effective ML-based NIDS, suitable for real-world deployment. In many cases, such a smart model does not produce a high rate of false alarms in cases of variations in the benign traffic distribution caused by a modification of the organization environment. In this configuration, the authors used the FedBatch fusion technique in a PyTorch implementation to detect DoS, probe, R2L and U2R attacks. The proposed architecture, using an MLP and an LSTM separately as a NIDS, was evaluated on the NF-UNSW-NB15-v2 and NF-BoT-IoT-v2 datasets, Sarhan et al. (2021c), against a centralized and localized approach. The FL architecture always achieved better performance than the localized one (approximately 80% accuracy), while having somewhat worse performance than the centralized one. Moreover, MLP and LSTM were trading blows since their performance was very close.

For NIDS, multiple classifiers of different types can be combined to classify network traffic as benign or malignant. The authors in Chatterjee and Hanawal (2021) proposed using a Noise-Tolerant Probabilistic Hybrid Ensemble Classification (PHEC) model, which is suitable for detecting threats in IoT environments (in centralized settings), and adapting it to an FL setting. This model considers the confidence values in the predicted labels rather than the true labels of each individual classifier. While simultaneously achieving a high FP Rate (FPR) and a high True Positive Rate (TPR) may not be feasible, the model allows tuning a single hyperparameter γ to achieve the desired trade-off between TPR and FPR. In this architecture, each node is responsible for the detection of only a particular type of intrusion, so instead of averaging out models, they are stacked, preventing the high influence of majority samples on the global model, whilst utilizing the FedAvg fusion technique. The use of weighted convex surrogate loss functions, like Biased SVM and Weighted Logistic Regression make the model Noise Robust, Maalouf and Siddiqi (2014). The proposed architecture, implemented using TensorFlow Federated, was evaluated on the NSL-KDD, DS2OS, Aubet and Pahl (2018), and SCADA datasets, Morris and Gao (2014a,b), for the IP, TCP, ICMP, UDP and other communication protocols. For the model architecture, an MLP NN was considered, while for the baseline centralized models, KNN and RF algorithms were utilized. Even though performance in FL PHEC was lower than in centralized PHEC, it yet achieved very high TPR along with a decent accuracy level, reaching 92% when detecting a variety of attacks.

Traditional NIDS methods, like Deep packet inspection and stateful protocol analysis, tend to be insufficient due to the huge amount of high-dimensional modern network traffic data, Bremler-Barr, Harchol, Hay and Koral (2014). The authors in Toldinas, Venčkauskas, Liutkevičius and Morkevičius (2022) propose transforming network traffic features (NTF) data into images, by collecting NTF data in a frame and then transforming each frame into an image, to get a limited but sufficient image dataset for model training and evaluation. The model's architecture is a 12-layer FL DNN using Stochastic GD with momentum (SGDM) and a custom training loop along with a custom fusion technique. The model's implementation was achieved with Simulink and its performance was evaluated on the BOUN-DDoS Dataset, Erhan and Anarım (2020), against a centralized Transfer Learning ResNet50 model and a 13-layer FTL DNN model. The performance of the FL and FTL models was close to the centralized model in some cases, reaching about 93% accuracy in detecting DDoS attacks, which is good enough, but also showcases that further research is required.

While FL-IDSs have displayed promising improvements, handling heterogeneous data distribution across multiple organizations is still a major challenge. The authors in Vucovich, Tarcar, Rebelo, Gade, Porwal, Rahman, Redino, Choi, Nandakumar, Schiller et al. (2022) propose the use of an under-complete AE with a Root Mean Square Propagation (RMSProp) optimizer and MSE loss function, paired with a sequential binary classifier. The AE was trained using FL on each client's private data separately, subsequently, the aggregated (global) AE was used by each client independently to create new local training data for the binary classifier and finally produce an aggregated (global) classifier using FL. To handle clients with different data distributions, the authors introduce the FedSam min-max scaler algorithm as well as a new sampling technique which is a combination of the Mini-Batch and Multi-Epoch FedAvg strategy and is suitable for equally weighting updates from all client nodes. The proposed solution was evaluated on the CIC-IDS2017, CSE-CIC-IDS2018, and NCC-DC, University of Southern California-Information Sciences Institute, datasets, against centrally trained AEs, classifiers and Federated Multi-Mini-Batch (Fed-MMB), Nasirigerdeh, Bakhtiari, Torkzadehmahani, Bayat, List, Blumenthal and Baumbach (2020). The proposed model achieved an F1-score of 91% and even outperformed the rival FedMMB model for the task of detecting DDoS attacks.

The authors in Verma, Breslin and O'Shea (2022) proposed the FLDID - FL-enabled hybrid model composed of a CNN, an LSTM and an MLP - as an IDS in SM environments, which utilizes a Paillier-based encryption during model parameters exchange. More specifically, the CNN extracts high-level feature representations, while the LSTM identifies the time-series patterns. The reason behind the use of encryption is the model's capability of allowing collaborative learning between different SM industries, using FL through secure communication, while exploiting the MQTT, CoAP and WebSocker protocols. The model was implemented using TensorFlow and was designed so that it can detect membership inference attacks, unwanted data leakage and reconstruction through inference, as well as GANs-based inference attacks. The proposed solution was evaluated on the X-IIoTID, Al-Hawawreh, Sitnikova and Aboutorab (2021), dataset, which takes into account the heterogeneity of IIoT network traffic as well as system procedures produced by a variety of IIoT devices, against a centralized IDS model, an isolated IDS model, and 3 SOTA models. Compared to the SOTA models, FLDID achieved higher accuracy (approximately 99%) and F1-score. It also

outperformed the isolated model while having a negligibly worse performance than the centralized model.

GANs solve the problem of limited, missing and imbalanced malicious IoT network traffic data by generating synthetic data, while FL allows different IoT devices to contribute to implementing a reliable IDS. The authors in Tabassum, Erbad, Lebda, Mohamed and Guizani (2022) proposed FEDGAN-IDS as a Privacy-preserving IDS using GANs and FL, where each IoT device has two NN models, a Generator and a Discriminator which are CNNs. Both the original local traffic data and the synthetic local data created by the local Generator are used in training the local Discriminator. A global Generator and a global Discriminator are produced through FL. The proposed solution was evaluated on the KDDCup99, the NSL-KDD, and the NSW-NB15 datasets, against several SOTA models, outperforming all of them with all three datasets, encompassing a variety of communication protocols, achieving an accuracy score of more than 99%.

Having a single central server makes FL susceptible to several risks, including hardware failures and security issues. Furthermore, it is difficult for some devices participating in the FL procedure to create direct links to the central server. To overcome these problems, the authors in Lian and Su (2022) propose Peer-to-Peer Orthogonal-Search Training for Edge-based FL (POSTER) as a decentralized FL architecture for IoT anomaly detection, in which the server performs worker management and model initialization for all workers before the training starts. Thereafter, all workers perform the training in a completely decentralized Peer-to-Peer (P2P) mode. For each worker C, if another randomly selected peer C' has a more recently trained model, then C receives the weights of C' and keeps them in a list. Finally, it performs weight averaging over its local model and all received peer models. The proposed architecture was evaluated on the IoT23, Sebastian Garcia (2020), dataset against a non-federated model version and a centralized model version, outperforming both of them while achieving an accuracy score of approximately 84%, in a TensorFlow implementation for detecting PortScan, botnet and DDoS attacks and exploiting various communication protocols.

Cyberattacks in IoMT could endanger patients' lives and expose healthcare organizations to legal actions against them. Towards IoMT defense, the authors in Singh, Gaba, Kaur, Hedabou and Gurtov (2022) proposed a Dew-Cloud, Ray (2017), IoMT framework with edge cloud computing to securely monitor patients' health. The result which occurs when decentralized systems are combined with centralized systems is a hierarchical architecture that can be trained using hierarchical FL. The Dew Intelligent Service (DIS) is embedded into the Dew-Cloud architecture to identify anomalies in the network traffic, through an HFL-HLSTM model. The global model is distributed to Dew servers deployed in various health institutions. The proposed architecture was evaluated on the NSL-KDD and TON_IoT datasets, against three SOTA models, outperforming all three of them while attaining an accuracy score of more than 99%. The designed implementation was performed with scikit-learn and TensorFlow libraries, in an effort to detect PortScan, XSS, ransomware, DDoS, password, injection and backdoor attacks, while exploiting the IP, TCP, MQTT, HTTP and DNS protocols.

6. Discussion, Lessons Learnt and Future Directions

The survey in the previous section on FL-based IDSs has shed light on the potential benefits of FL-IDS compared to traditional centralized IDS. One of the main advantages of FL-IDS is its decentralized and distributed approach, which eliminates the risk of a single point of failure. Another great advantage of FL-IDS is that it maintains data isolation and perpetuates privacy and confidentiality while enforcing limited and secure information exchange between participating parties. This decentralization allows multiple nodes in the network to participate and contribute to the learning process and model improvement, resulting in a more robust and accurate final AI model which benefits every node and organization. Finally, FL-IDS constitutes a scalable ML solution, since it offloads the training and evaluation procedure across multiple independent nodes.

However, it is crucial to keep in mind that the survey has some limitations that should not be overlooked. To begin with, it is based on a wide yet limited number of recently published studies and may not accurately reflect the current SOTA in this field, as it is constantly and actively evolving. Additionally, it may not have considered all the potential challenges and limitations of FL-IDS, such as the difficulties in implementing such systems in real-world scenarios. For instance, towards implementing Horizontal FL there is a necessity for the existence of multiple subnets providing the same features for FL to be applied, creating a prerequisite for such applications. Moreover, it may also have not fully evaluated the trade-offs between privacy, security, and model performance in FL-IDS. Therefore, while it provides valuable insights into the potential of FL-IDS in intrusion detection, it is important to acknowledge its limitations and take them into account when interpreting the results. Moving forward, continuous research and development in this field can help address these limitations and improve the accuracy and efficiency of FL-IDS. Furthermore, considering the trade-offs between privacy, security, and model performance can ensure that FL-IDS is implemented in the most effective and secure manner possible.

Based on the reviewed literature, it was more than obvious that although FL is a great conceptualization for applications like IDS, a significant number of limitations and unresolved issues nonetheless persist, making it laborious to apply in practice. As was encountered in the reviewed articles, different model aggregation strategies can result in completely different produced models, which may or may not have converged to a global minimum loss. No globally accepted aggregation strategy exists and is unlikely to exist due to different network infrastructures and participating

nodes' hardware. Also, the size of data that needs to be shared in each training round is enormous and needs to be significantly reduced, either through data compression or other innovative techniques. Additionally, with regard to the data there exists an overall lack of balanced datasets including the newly evolved attacks in the cybersecurity domain, posing a barrier to utilizing the data-centric approach, even if it is thought of as a finer approach. Another aspect is the encryption algorithm utilized for securing the data being transferred between participating devices, which needs to be fast both in encryption and decryption, to reduce the FL time and required hardware resources.

Another major factor in the slow adoption of FL-IDS is the lack of fully-featured and suitable FL frameworks which are suitable towards this goal. Most FL frameworks are very new and recently initiated projects and thus are being actively maintained with daily development activity. However, although they are advancing fast, a lot of useful features are still missing. Furthermore, given the immense number of new studies and approaches proposed in FL, extracting the best ideas and properly implementing them within existing frameworks requires a lot of development time and resources. On the other hand, in expectation of standardized and globally accepted FL features as well as a number of feature-complete FL frameworks which implement said features, FL-IDS will not be ready for mass deployment.

In conclusion, FL-IDS has the potential to offer a more accurate and efficient way to detect intrusions compared to traditional centralized systems. However, it's important to keep in mind the limitations of the current SOTA and therefore of the current survey and continue wandering in the field further. With additional research and development, and by taking into account the trade-offs between privacy, security, and model performance, FL-IDS can be implemented in the best and most secure way possible.

7. Conclusions

To conclude, after conducting a thorough analysis of several research papers on Federated IDPS, this survey has provided valuable insights into the techniques, challenges, and solutions related to this topic. The research indicates that these systems have several advantages, such as privacy and resilience against potential cyberattacks. However, the analysis also revealed several challenges that need to be addressed, including scalability, the difficulty of ensuring data privacy without degrading accuracy, and the importance of having proper communication and collaboration between the participating clients. Overall, this survey intends to serve as a helpful guide for researchers in the field of IDPS. The classification of the reviewed papers provides a comprehensive overview of the current SOTA and can guide future research. The challenges identified in this document offer valuable insights into the key issues that need to be addressed to develop effective federated IDPS.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. URL: https://www.tensorflow.org/. software available from tensorflow.org.

Abdel-Basset, M., Moustafa, N., Hawash, H., Razzak, I., Sallam, K.M., Elkomy, O.M., 2021. Federated intrusion detection in blockchain-based smart transportation systems. IEEE Transactions on Intelligent Transportation Systems 23, 2523–2537.

Al-Hawawreh, M., Sitnikova, E., Aboutorab, N., 2021. X-iiotid: A connectivity- and device-agnostic intrusion dataset for industrial internet of things. URL: https://dx.doi.org/10.21227/mpb6-py55, doi:10.21227/mpb6-py55.

Alaeddine, B., 2020. Lstm deep learning method for network intrusion detection system. International Journal of Electrical and Computer Engineering 10. doi:10.11591/ijece.v10i3.pp3315-3322.

Alamleh, A., Albahri, O., Zaidan, A., Albahri, A., Alamoodi, A., Zaidan, B., Qahtan, S., Alsatar, H., Al-Samarraay, M.S., Jasim, A.N., 2022. Federated learning for iomt applications: A standardisation and benchmarking framework of intrusion detection systems. IEEE Journal of Biomedical and Health Informatics .

Alazab, A., Hobbs, M., Abawajy, J., Alazab, M., 2012. Using feature selection for intrusion detection system, in: 2012 international symposium on communications and information technologies (ISCIT), IEEE. pp. 296–301.

Albawi, S., Mohammed, T.A., Al-Zawi, S., 2017. Understanding of a convolutional neural network, in: 2017 International Conference on Engineering and Technology (ICET), pp. 1–6. doi:10.1109/ICEngTechnol.2017.8308186.

Alexopoulos, N., Vasilomanolakis, E., Ivánkó, N., Mühlhäuser, M., 2018. Towards Blockchain-Based Collaborative Intrusion Detection Systems: 12th International Conference, CRITIS 2017, Lucca, Italy, October 8-13, 2017, Revised Selected Papers. Springer. chapter 10. pp. 107–118. doi:10.1007/978-3-319-99843-5_10.

Alghushairy, O., Alsini, R., Soule, T., Ma, X., 2020. A review of local outlier factor algorithms for outlier detection in big data streams. Big Data and Cognitive Computing 5, 1.

Ali, S., Li, Q., Yousafzai, A., 2024. Blockchain and federated learning-based intrusion detection approaches for edge-enabled industrial iot networks: a survey. Ad Hoc Networks 152. doi:https://doi.org/10.1016/j.adhoc.2023.103320.

Ali, S., Qaisar, S.B., Saeed, H., Khan, M.F., Naeem, M., Anpalagan, A., 2015. Network challenges for cyber physical systems with tiny wireless devices: A case study on reliable pipeline condition monitoring. Sensors 15, 7172–7205. URL: https://www.mdpi.com/1424-8220/15/4/7172, doi:10.3390/s150407172.

Aliyu, I., Feliciano, M.C., Van Engelenburg, S., Kim, D.O., Lim, C.G., 2021. A blockchain-based federated forest for sdn-enabled in-vehicle network intrusion detection system. IEEE Access 9, 102593–102608.

Alkadi, O., Moustafa, N., Turnbull, B., Choo, K.K.R., 2021. A deep blockchain framework-enabled collaborative intrusion detection for protecting iot and cloud networks. IEEE Internet of Things Journal 8, 9463–9472. doi:10.1109/JIOT.2020.2996590.

Almgren, M., Jonsson, E., 2004. Using active learning in intrusion detection, in: Proceedings. 17th IEEE Computer Security Foundations Workshop, 2004., IEEE. pp. 88–98.

Almomani, I., Al-Kasasbeh, B., Al-Akhras, M., 2016. Wsn-ds: A dataset for intrusion detection systems in wireless sensor networks. Journal of Sensors 2016.

Alrajeh, N.A., Khan, S., Shams, B., 2013. Intrusion detection systems in wireless sensor networks: a review. International Journal of Distributed Sensor Networks 9, 167575.

Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., Anwar, A., 2020. Ton_iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems. IEEE Access 8, 165130–165150.

Alsamiri, J., Khalid, A., 2023. Federated learning for intrusion detection systems in internet of vehicles: A general taxonomy, applications, and future directions. Future Internet doi:https://doi.org/10.3390/fi15120403.

Althubiti, S.A., Jones, E.M., Roy, K., 2018. Lstm for anomaly-based network intrusion detection, in: 2018 28th International Telecommunication Networks and Applications Conference (ITNAC), pp. 1–3. doi:10.1109/ATNAC.2018.8615300.

Anderson, J.P., 1980. Computer security threat monitoring and surveillance.

Ansari, M.S., Bartoš, V., Lee, B., 2022. Gru-based deep learning approach for network intrusion alert prediction. Future Generation Computer Systems 128, 235–247. doi:https://doi.org/10.1016/j.future.2021.09.040.

Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J.A., Invernizzi, L., Kallitsis, M., et al., 2017. Understanding the mirai botnet, in: 26th USENIX security symposium (USENIX Security 17), pp. 1093–1110.

Antwi, M., Adnane, A., Ahmad, F., Hussain, R., ur Rehman, M.H., Kerrache, C.A., 2021. The case of hyperledger fabric as a blockchain solution for healthcare applications. Blockchain: Research and Applications 2, 100012.

Aouedi, O., Piamrat, K., Muller, G., Singh, K., 2022a. Federated semi-supervised learning for attack detection in industrial internet of things. IEEE Transactions on Industrial Informatics .

Aouedi, O., Piamrat, K., Muller, G., Singh, K., 2022b. Fluids: Federated learning with semi-supervised approach for intrusion detection system, in: 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), IEEE. pp. 523–524.

Arisdakessian, S., Wahab, O.A., Mourad, A., Otrok, H., Guizani, M., 2022. A survey on iot intrusion detection: Federated learning, game theory, social psychology and explainable ai as future directions. IEEE Internet of Things Journal .

Arulkumaran, K., Deisenroth, M.P., Brundage, M., Bharath, A.A., 2017. Deep reinforcement learning: A brief survey. IEEE Signal Processing Magazine 34, 26–38.

Asad, M., Moustafa, A., Ito, T., 2020a. Fedopt: Towards communication efficiency and privacy preservation in federated learning. Applied Sciences 10, 2864.

Asad, M., Moustafa, A., Yu, C., 2020b. A critical evaluation of privacy and security threats in federated learning. Sensors 20. URL: https://www.mdpi.com/1424-8220/20/24/7182, doi:10.3390/s20247182.

Ashraf, J., Keshk, M., Moustafa, N., Abdel-Basset, M., Khurshid, H., Bakhshi, A.D., Mostafa, R.R., 2021. Iotbot-ids: A novel statistical learning-enabled botnet detection framework for protecting networks of smart cities. Sustainable Cities and Society 72, 103041.

Attota, D.C., Mothukuri, V., Parizi, R.M., Pouriyeh, S., 2021. An ensemble multi-view federated learning intrusion detection for iot. IEEE Access 9, 117734–117745.

Aubet, F., Pahl, M., 2018. Ds2os traffic traces.

Barlow, H.B., 1989. Unsupervised learning. Neural computation 1, 295–311.

Bertoli, G.d.C., Junior, L.A.P., Santos, A.L.d., Saotome, O., 2022. Generalizing intrusion detection for heterogeneous networks: A stacked-unsupervised federated learning approach. arXiv preprint arXiv:2209.00721 .

Blanchard, P., El Mhamdi, E.M., Guerraoui, R., Stainer, J., 2017. Machine learning with adversaries: Byzantine tolerant gradient descent. Advances in Neural Information Processing Systems 30.

Booij, T.M., Chiscop, I., Meeuwissen, E., Moustafa, N., den Hartog, F.T., 2021. Ton_iot: The role of heterogeneity and the need for standardization of features and attack types in iot network intrusion data sets. IEEE Internet of Things Journal 9, 485–496.

Bremler-Barr, A., Harchol, Y., Hay, D., Koral, Y., 2014. Deep packet inspection as a service, in: Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies, pp. 271–282.

Caldas, S., Duddu, S.M.K., Wu, P., Li, T., Konečnỳ, J., McMahan, H.B., Smith, V., Talwalkar, A., 2018. Leaf: A benchmark for federated settings.

arXiv preprint arXiv:1812.01097 .

Cetin, B., Lazar, A., Kim, J., Sim, A., Wu, K., 2019. Federated wireless network intrusion detection, in: 2019 IEEE International Conference on Big Data (Big Data), IEEE. pp. 6004–6006.

Chakrabarti, S., Chakraborty, M., Mukhopadhyay, I., 2010. Study of snort-based ids, in: Proceedings of the International Conference and Workshop on Emerging Trends in Technology, pp. 43–47.

Chatterjee, S., Hanawal, M.K., 2021. Federated learning for intrusion detection in iot security: A hybrid ensemble approach. URL: https://arxiv.org/abs/2106.15349, doi:10.48550/ARXIV.2106.15349.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research 16, 321–357.

Chen, C., Gong, Y., Tian, Y., 2008. Semi-supervised learning methods for network intrusion detection, in: 2008 IEEE international conference on systems, man and cybernetics, IEEE. pp. 2603–2608.

Chen, Y., Su, L., Xu, J., 2017. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. Proceedings of the ACM on Measurement and Analysis of Computing Systems 1, 1–25.

Chen, Z., Lv, N., Liu, P., Fang, Y., Chen, K., Pan, W., 2020. Intrusion detection for wireless edge networks based on federated learning. IEEE Access 8, 217463–217472.

Cheng, Y., Lu, J., Niyato, D., Lyu, B., Kang, J., Zhu, S., 2022. Federated transfer learning with client selection for intrusion detection in mobile edge computing. IEEE Communications Letters 26, 552–556.

Choi, H., Kim, M., Lee, G., Kim, W., 2019. Unsupervised learning approach for network intrusion detection system using autoencoders. The Journal of Supercomputing 75, 5597–5621.

Chollet, F., et al., 2015. Keras. https://keras.io.

Contreras-Castillo, J., Zeadally, S., Guerrero-Ibañez, J.A., 2018. Internet of vehicles: Architecture, protocols, and security. IEEE Internet of Things Journal 5, 3701–3709. doi:10.1109/JIOT.2017.2690902.

Copperwaite, M., Leifer, C., 2015. Learning flask framework. Packt Publishing Ltd.

Cunningham, P., Cord, M., Delany, S.J., 2008. Supervised learning. Machine learning techniques for multimedia: case studies on organization and retrieval , 21–49.

Cunningham, P., Delany, S.J., 2021. k-nearest neighbour classifiers-a tutorial. ACM computing surveys (CSUR) 54, 1–25.

Davis, J., Edgar, T., Graybill, R., Korambath, P., Schott, B., Swink, D., Wang, J., Wetzel, J., 2015. Smart manufacturing. Annual review of chemical and biomolecular engineering 6, 141–160.

Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Ranzato, M., Senior, A., Tucker, P., Yang, K., et al., 2012. Large scale distributed deep networks. Advances in neural information processing systems 25.

DeMaris, A., 1995. A tutorial in logistic regression. Journal of Marriage and the Family , 956–968.

Deng, K., Chen, Z., Zhang, S., Gong, C., Zhu, J., 2019. Content compression coding for federated learning, in: 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP), IEEE. pp. 1–6.

Deng, L., 2012. The mnist database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine 29, 141–142.

Denning, D.E., 1987. An intrusion-detection model. IEEE Transactions on software engineering , 222–232.

Dey, N., Ashour, A.S., Shi, F., Fong, S.J., Tavares, J.M.R., 2018. Medical cyber-physical systems: A survey. Journal of medical systems 42, 1–13.

Di Mauro, M., Galatro, G., Fortino, G., Liotta, A., 2021. Supervised feature selection techniques in network intrusion detection: A critical review. Engineering Applications of Artificial Intelligence 101, 104216.

Dobilas, S., 2022. Lstm recurrent neural networks — how to teach a network to remember the past. URL: https://shorturl.at/nEL36.

Documentation, S., 2020. Simulation and model-based design. URL: https://www.mathworks.com/products/simulink.html.

Dong, J., Roth, A., Su, W.J., 2019. Gaussian differential privacy. arXiv preprint arXiv:1905.02383 .

Dong, T., Li, S., Qiu, H., Lu, J., 2022. An interpretable federated learning-based network intrusion detection framework. URL: https://arxiv.org/abs/2201.03134, doi:10.48550/ARXIV.2201.03134.

Dong, T., Qiu, H., Lu, J., Qiu, M., Fan, C., 2021. Towards fast network intrusion detection based on efficiency-preserving federated learning, in: 2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), IEEE. pp. 468–475.

Draper-Gil, G., Lashkari, A.H., Mamun, M.S.I., Ghorbani, A.A., 2016. Characterization of encrypted and vpn traffic using time-related, in: Proceedings of the 2nd international conference on information systems security and privacy (ICISSP), pp. 407–414.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M., 2006a. Our data, ourselves: Privacy via distributed noise generation, in: Annual international conference on the theory and applications of cryptographic techniques, Springer. pp. 486–503.

Dwork, C., McSherry, F., Nissim, K., Smith, A., 2006b. Calibrating noise to sensitivity in private data analysis, in: Theory of cryptography conference, Springer. pp. 265–284.

Dwork, C., McSherry, F., Nissim, K., Smith, A., 2006c. Calibrating noise to sensitivity in private data analysis, in: Theory of cryptography conference, Springer. pp. 265–284.

Elsayed, M.S., Le-Khac, N.A., Jurcut, A.D., 2020. Insdn: A novel sdn intrusion dataset. IEEE Access 8, 165263–165284.

Erhan, D., Anarım, E., 2020. Boğaziçi university distributed denial of service dataset. Data in brief 32.

Fan, Y., Li, Y., Zhan, M., Cui, H., Zhang, Y., 2020. Iotdefender: A federated transfer learning intrusion detection framework for 5g iot, in: 2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE), pp. 88–95. doi:10.1109/BigDataSE50710.2020.00020.

Fang, X., Misra, S., Xue, G., Yang, D., 2011. Smart grid—the new and improved power grid: A survey. IEEE communications surveys & tutorials 14, 944–980.

Fangchun, Y., Wang, S., Li, J., Liu, Z., Sun, Q., 2014. An overview of internet of vehicles. Communications, China 11, 1–15. doi:10.1109/CC.2014.6969789.

Farahnakian, F., Heikkonen, J., 2018. A deep auto-encoder based approach for intrusion detection system, in: 2018 20th International Conference on Advanced Communication Technology (ICACT), IEEE. pp. 178–183.

Farnaaz, N., Jabbar, M., 2016. Random forest modeling for network intrusion detection system. Procedia Computer Science 89, 213–217. URL: https://www.sciencedirect.com/science/article/pii/S1877050916311127, doi:https://doi.org/10.1016/j.procs.2016.06.047. twelfth International Conference on Communication Networks, ICCN 2016, August 19– 21, 2016, Bangalore, India Twelfth International Conference on Data Mining and Warehousing, ICDMW 2016, August 19-21, 2016, Bangalore, India Twelfth International Conference on Image and Signal Processing, ICISP 2016, August 19-21, 2016, Bangalore, India.

Farsi, M., Ratcliff, K., Barbosa, M., 1999. An overview of controller area network. Computing & Control Engineering Journal 10, 113–120.

Fontugne, R., Borgnat, P., Abry, P., Fukuda, K., 2010. Mawilab: combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking, in: Proceedings of the 6th International COnference, pp. 1–12.

Foster, I., Koscher, K., 2015. Exploring controller area networks. login. USENIX Association 40.

Frazão, I., Abreu, P., Cruz, T., Araújo, H., Simões, P., 2019. Cyber-security modbus ics dataset. URL: https://dx.doi.org/10.21227/pjff-1a03, doi:10.21227/pjff-1a03.

Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. Annals of statistics , 1189–1232.

Friha, O., Ferrag, M.A., Shu, L., Maglaras, L., Choo, K.K.R., Nafaa, M., 2022. Felids: Federated learning-based intrusion detection system for agricultural internet of things. Journal of Parallel and Distributed Computing 165, 17–31.

Gao, G., Wang, M., Huang, H., Tang, W., 2021. Agricultural irrigation area prediction based on improved random forest model. doi:10.21203/rs.3.rs-156767/v1.

García, S., Luengo, J., Herrera, F., 2015. Data preprocessing in data mining. Springer.

Garcia, S., Parmisano, A., Erquiaga, M.J., 2020. IoT-23: A labeled dataset with malicious and benign IoT network traffic. URL: https://doi.org/10.5281/zenodo.4743746, doi:10.5281/zenodo.4743746. More details here https://www.stratosphereips.org/datasets-iot23.

Gharib, A., Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A., 2016. An evaluation framework for intrusion detection dataset, in: 2016 International Conference on Information Science and Security (ICISS), pp. 1–6. doi:10.1109/ICISSEC.2016.7885840.

Girdhar, K., Singh, C., Kumar, Y., 2023. Ai and blockchain for cybersecurity in cyber-physical systems: Challenges and future research agenda, in: Blockchain for Cybersecurity in Cyber-Physical Systems, Springer. pp. 185–213.

Goh, J., Adepu, S., Junejo, K.N., Mathur, A., 2017. A dataset to support research in the design of secure water treatment systems, in: International conference on critical information infrastructures security, Springer. pp. 88–99.

Görnitz, N., Kloft, M., Rieck, K., Brefeld, U., 2009. Active learning for network intrusion detection, in: Proceedings of the 2nd ACM workshop on Security and artificial intelligence, pp. 47–54.

Grinberg, M., 2018a. Flask web development: developing web applications with python. " O'Reilly Media, Inc.".

Grinberg, M., 2018b. Flask web development: developing web applications with python. " O'Reilly Media, Inc.".

Gümüşbaş, D., Yıldırım, T., Genovese, A., Scotti, F., 2020. A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems. IEEE Systems Journal 15, 1717–1731.

Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., Wang, Y., 2021. Transformer in transformer. CoRR abs/2103.00112. URL: https://arxiv.org/abs/2103.00112, arXiv:2103.00112.

Hard, A., Kiddon, C.M., Ramage, D., Beaufays, F., Eichner, H., Rao, K., Mathews, R., Augenstein, S., 2018. Federated learning for mobile keyboard prediction. URL: https://arxiv.org/abs/1811.03604.

Hariri, S., Kind, M.C., Brunner, R.J., 2019. Extended isolation forest. IEEE Transactions on Knowledge and Data Engineering 33, 1479–1489.

He, C., Li, S., So, J., Zeng, X., Zhang, M., Wang, H., Wang, X., Vepakomma, P., Singh, A., Qiu, H., et al., 2020. Fedml: A research library and benchmark for federated machine learning. arXiv preprint arXiv:2007.13518 .

He, H., Bai, Y., Garcia, E.A., Li, S., 2008. Adasyn: Adaptive synthetic sampling approach for imbalanced learning, in: 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence), IEEE. pp. 1322–1328.

Hei, X., Yin, X., Wang, Y., Ren, J., Zhu, L., 2020. A trusted feature aggregator federated learning for distributed malicious attack detection. Computers & Security 99, 102033.

Heidari, A., Jabraeil Jamali, M.A., 2022. Internet of things intrusion detection systems: A comprehensive review and future directions. Cluster Computing , 1–28.

Hindy, H., Bayne, E., Bures, M., Atkinson, R., Tachtatzis, C., Bellekens, X., 2021. Machine learning based iot intrusion detection system: An mqtt case study (mqtt-iot-ids2020 dataset), in: Selected Papers from the 12th International Networking Conference: INC 2020, Springer. pp. 73–84.

Hopfield, J.J., 1982. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the national academy of sciences 79, 2554–2558.

HPL, S.C., 2002. Introduction to the controller area network (can). Application Report SLOA101 , 1–17.

Hristov, M., Nenova, M., Iliev, G., Avresky, D., 2021. Didarknet: A contemporary approach to detect and characterize the darknet traffic using deep image learning, in: Proceedings of the 2020 10th International Conference on Communication and Network Security, Association for Computing Machinery. pp. 1–13.

Huang, C., Huang, J., Liu, X., 2022. Cross-silo federated learning: Challenges and opportunities. arXiv preprint arXiv:2206.12949 .

Hubert, M., Debruyne, M., 2010. Minimum covariance determinant. Wiley interdisciplinary reviews: Computational statistics 2, 36–43.

Huong, T.T., Bac, T.P., Long, D.M., Luong, T.D., Dan, N.M., Thang, B.D., Tran, K.P., et al., 2021. Detecting cyberattacks using anomaly detection in industrial control systems: A federated learning approach. Computers in Industry 132, 103509.

Imrana, Y., Xiang, Y., Ali, L., Abdul-Rauf, Z., 2021. A bidirectional lstm deep learning approach for intrusion detection. Expert Systems with Applications 185, 115524. doi:https://doi.org/10.1016/j.eswa.2021.115524.

Janssens, J., Huszár, F., Postma, E., van den Herik, J., 2012. Stochastic outlier selection. Technical Report. Technical Report. Tilburg University, Tilburg Center for Cognition and Communication, Tilburg, The Netherlands.

Jiang, L., Zhang, H., Cai, Z., 2008. A novel bayes model: Hidden naive bayes. IEEE Transactions on knowledge and data engineering 21, 1361–1371.

Johnson, A.E., Pollard, T.J., Shen, L., Lehman, L.w.H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Anthony Celi, L., Mark, R.G., 2016. Mimic-iii, a freely accessible critical care database. Scientific data 3, 1–9.

Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al., 2021. Advances and open problems in federated learning. Foundations and Trends® in Machine Learning 14, 1–210.

Kang, H., Ahn, D.H., Lee, G.M., Yoo, J.D., Park, K.H., Kim, H.K., 2019. Iot network intrusion dataset. URL: https://dx.doi.org/10.21227/q70p-q449, doi:10.21227/q70p-q449.

Kavitha, T., Sridharan, D., 2010. Security vulnerabilities in wireless sensor networks: A survey. Journal of information Assurance and Security 5, 31–44.

Kelli, V., Argyriou, V., Lagkas, T., Fragulis, G., Grigoriou, E., Sarigiannidis, P., 2021. Ids for industrial applications: a federated learning approach with active personalization. Sensors 21, 6743.

Keogh, E., Lin, J., Fu, A., 2005a. Hot sax: Efficiently finding the most unusual time series subsequence, in: Fifth IEEE International Conference on Data Mining (ICDM'05), Ieee. pp. 8–pp.

Keogh, E., Lin, J., Fu, A., 2005b. Hot sax: Efficiently finding the most unusual time series subsequence, in: Fifth IEEE International Conference on Data Mining (ICDM'05), Ieee. pp. 8–pp.

Kholod, I., Yanaki, E., Fomichev, D., Shalugin, E., Novikova, E., Filippov, E., Nordlund, M., 2020a. Open-source federated learning frameworks for iot: A comparative review and analysis. Sensors 21, 167.

Kholod, I., Yanaki, E., Fomichev, D., Shalugin, E., Novikova, E., Filippov, E., Nordlund, M., 2020b. Open-source federated learning frameworks for iot: A comparative review and analysis. Sensors 21, 167. doi:10.3390/s21010167.

Kim, N., Krasner, A., Kosinski, C., Wininger, M., Qadri, M., Kappus, Z., Danish, S., Craelius, W., 2016. Trending autoregulatory indices during treatment for traumatic brain injury. Journal of clinical monitoring and computing 30, 821–831.

Kingma, D.P., Welling, M., 2013a. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 .

Kingma, D.P., Welling, M., 2013b. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 .

Kira, K., Rendell, L.A., 1992. A practical approach to feature selection, in: Machine learning proceedings 1992. Elsevier, pp. 249–256.

Kolias, C., Kambourakis, G., Stavrou, A., Gritzalis, S., 2015. Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset. IEEE Communications Surveys & Tutorials 18, 184–208.

Koroniotis, N., 2020. Designing an effective network forensic framework for the investigation of botnets in the Internet of Things. Ph.D. thesis. UNSW Sydney.

Koroniotis, N., Moustafa, N., 2020. Enhancing network forensics with particle swarm and deep learning: The particle deep framework. arXiv preprint arXiv:2005.00722 .

Koroniotis, N., Moustafa, N., Schiliro, F., Gauravaram, P., Janicke, H., 2020a. A holistic review of cybersecurity and reliability perspectives in smart airports. IEEE Access 8, 209802–209834.

Koroniotis, N., Moustafa, N., Sitnikova, E., 2020b. A new network forensic framework based on deep learning for internet of things networks: A particle deep framework. Future Generation Computer Systems 110, 91–106.

Koroniotis, N., Moustafa, N., Sitnikova, E., Slay, J., 2018. Towards developing network forensic mechanism for botnet activities in the iot based on machine learning techniques, in: Mobile Networks and Management: 9th International Conference, MONAMI 2017, Melbourne, Australia, December 13-15, 2017, Proceedings 9, Springer. pp. 30–44.

Koroniotis, N., Moustafa, N., Sitnikova, E., Turnbull, B., 2019. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. Future Generation Computer Systems 100, 779–796.

Kriegel, H.P., Schubert, M., Zimek, A., 2008. Angle-based outlier detection in high-dimensional data, in: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 444–452.

Krizhevsky, A., Hinton, G., et al., 2009. Learning multiple layers of features from tiny images.

Kumar, K.S., Nair, S.A.H., Roy, D.G., Rajalingam, B., Kumar, R.S., 2021. Security and privacy-aware artificial intrusion detection system using federated machine learning. Computers & Electrical Engineering 96, 107440.

Kundu, A., Yu, P., Wynter, L., Lim, S.H., 2022. Robustness and personalization in federated learning: A unified approach via regularization, in: 2022 IEEE International Conference on Edge Computing and Communications (EDGE), IEEE. pp. 1–11.

Lashkari, A.H., Draper-Gil, G., Mamun, M.S.I., Ghorbani, A.A., et al., 2017. Characterization of tor traffic using time based features., in: ICISSp, pp. 253–262.

Lavaur, L., Pahl, M.O., Busnel, Y., Autrel, F., 2022. The evolution of federated learning-based intrusion detection and mitigation: A survey. IEEE Transactions on Network and Service Management doi:10.1109/TNSM.2022.3177512.

Lee, H., Jeong, S.H., Kim, H.K., 2017. Otids: A novel intrusion detection system for in-vehicle network by using remote frame, in: 2017 15th Annual Conference on Privacy, Security and Trust (PST), IEEE. pp. 57–5709.

Lee, I., Sokolsky, O., 2010. Medical cyber physical systems, in: Design automation conference, IEEE. pp. 743–748.

Lee, I., Sokolsky, O., Chen, S., Hatcliff, J., Jee, E., Kim, B., King, A., Mullen-Fortino, M., Park, S., Roederer, A., et al., 2011. Challenges and research directions in medical cyber–physical systems. Proceedings of the IEEE 100, 75–90.

Li, B., Wu, Y., Song, J., Lu, R., Li, T., Zhao, L., 2021a. Deepfed: Federated deep learning for intrusion detection in industrial cyber–physical systems. IEEE Transactions on Industrial Informatics 17, 5615–5624. doi:10.1109/TII.2020.3023430.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R.P., Tang, J., Liu, H., 2017. Feature selection: A data perspective. ACM computing surveys (CSUR) 50, 1–45.

Li, J., Zhang, Z., Li, Y., Guo, X., Li, H., 2021b. Fids: Detecting ddos through federated learning based method, in: 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE. pp. 856–862.

Li, K.L., Huang, H.K., Tian, S.F., Xu, W., 2003. Improving one-class svm for anomaly detection, in: Proceedings of the 2003 international conference on machine learning and cybernetics (IEEE Cat. No. 03EX693), IEEE. pp. 3077–3081.

Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., Li, Y., Liu, X., He, B., 2021c. A survey on federated learning systems: vision, hype and reality for data privacy and protection. IEEE Transactions on Knowledge and Data Engineering .

Li, S., Cheng, Y., Wang, W., Liu, Y., Chen, T., 2020a. Learning to detect malicious clients for robust federated learning. arXiv preprint arXiv:2002.00211 .

Li, S., Xu, L.D., Zhao, S., 2018. 5g internet of things: A survey. Journal of Industrial Information Integration 10, 1–9. URL: https://www.sciencedirect.com/science/article/pii/S2452414X18300037, doi:https://doi.org/10.1016/j.jii.2018.01.005.

Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V., 2020b. Federated optimization in heterogeneous networks. Proceedings of Machine Learning and Systems 2, 429–450.

Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smithy, V., 2019. Feddane: A federated newton-type method, in: 2019 53rd Asilomar Conference on Signals, Systems, and Computers, IEEE. pp. 1227–1231.

Li, Y., 2017. Deep reinforcement learning: An overview. arXiv preprint arXiv:1701.07274 .

Li, Z., Liu, F., Yang, W., Peng, S., Zhou, J., 2022. A survey of convolutional neural networks: Analysis, applications, and prospects. IEEE Transactions on Neural Networks and Learning Systems 33, 6999–7019. doi:10.1109/TNNLS.2021.3084827.

Lian, Z., Su, C., 2022. Decentralized federated learning for internet of things anomaly detection, in: Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security, pp. 1249–1251.

Liang, C., Shanmugam, B., Azam, S., Karim, A., Islam, A., Zamani, M., Kavianpour, S., Idris, N.B., 2020. Intrusion detection system for the internet of things based on blockchain and multi-agent systems. Electronics 9. URL: https://www.mdpi.com/2079-9292/9/7/1120.

Liang, H., Liu, D., Zeng, X., Ye, C., 2022. An intrusion detection method for advanced metering infrastructure based on federated learning. Journal of Modern Power Systems and Clean Energy .

Lin, S., Clark, R., Birke, R., Schönborn, S., Trigoni, N., Roberts, S., 2020. Anomaly detection for time series using vae-lstm hybrid model, in: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Ieee. pp. 4322–4326.

Liu, G., Bao, H., Han, B., et al., 2018. A stacked autoencoder-based deep neural network for achieving gearbox fault diagnosis. Mathematical Problems in Engineering 2018.

Liu, H., Zhang, S., Zhang, P., Zhou, X., Shao, X., Pu, G., Zhang, Y., 2021. Blockchain and federated learning for collaborative intrusion detection in vehicular edge computing. IEEE Transactions on Vehicular Technology 70, 6073–6084.

Liu, W., Xu, X., Wu, L., Qi, L., Jolfaei, A., Ding, W., Khosravi, M.R., 2022. Intrusion detection for maritime transportation systems with batch federated aggregation. IEEE Transactions on Intelligent Transportation Systems , 1–12doi:10.1109/TITS.2022.3181436.

Lomax, S., Vadera, S., 2013. A survey of cost-sensitive decision tree induction algorithms. ACM Computing Surveys (CSUR) 45, 1–35.

Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., 2020a. Application of deep reinforcement learning to intrusion detection for supervised problems. Expert Systems with Applications .

Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., 2020b. Application of deep reinforcement learning to intrusion detection for supervised problems. Expert Systems with Applications 141, 112963.

Ma, R., Chen, H.H., Huang, Y.R., Meng, W., 2013. Smart grid communication: Its challenges and opportunities. IEEE transactions on Smart Grid 4, 36–46.

Maalouf, M., Siddiqi, M., 2014. Weighted logistic regression for large-scale imbalanced and rare events data. Knowledge-Based Systems 59, 142–148.

Mahdavifar, S., Alhadidi, D., Ghorbani, A.A., 2022. Effective and efficient hybrid android malware classification using pseudo-label stacked auto-encoder. Journal of network and systems management 30, 1–34.

Mahdavifar, S., Kadir, A.F.A., Fatemi, R., Alhadidi, D., Ghorbani, A.A., 2020. Dynamic android malware category classification using semi-supervised deep learning, in: 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), IEEE. pp. 515–522.

Man, D., Zeng, F., Yang, W., Yu, M., Lv, J., Wang, Y., 2021. Intelligent intrusion detection based on federated learning for edge-assisted internet of things. Security and Communication Networks 2021.

Markovic, T., Leon, M., Buffoni, D., Punnekkat, S., 2022. Random forest based on federated learning for intrusion detection, in: IFIP International Conference on Artificial Intelligence Applications and Innovations, Springer. pp. 132–144.

Mathew, A., Mathew, J., Govind, M., Mooppan, A., 2017. An improved transfer learning approach for intrusion detection. Procedia Computer Science 115, 251–257. doi:https://doi.org/10.1016/j.procs.2017.09.132. 7th

International Conference on Advances in Computing & Communications, ICACC-2017, 22-24 August 2017, Cochin, India.

McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A., 2017. Communication-efficient learning of deep networks from decentralized data, in: Artificial intelligence and statistics, PMLR. pp. 1273–1282.

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., Elovici, Y., 2018. N-baiot—network-based detection of iot botnet attacks using deep autoencoders. IEEE Pervasive Computing 17, 12–22.

Meng, Q., Chen, W., Wang, Y., Ma, Z.M., Liu, T.Y., 2017. Convergence analysis of distributed stochastic gradient descent with shuffling. arXiv preprint arXiv:1709.10432 .

Meng, W., Tischhauser, E.W., Wang, Q., Wang, Y., Han, J., 2018. When intrusion detection meets blockchain technology: A review. IEEE Access 6, 10179–10188. doi:10.1109/ACCESS.2018.2799854.

Mirsky, Y., Doitshman, T., Elovici, Y., Shabtai, A., 2018a. Kitsune: an ensemble of autoencoders for online network intrusion detection. arXiv preprint arXiv:1802.09089 .

Mirsky, Y., Doitshman, T., Elovici, Y., Shabtai, A., 2018b. Kitsune: an ensemble of autoencoders for online network intrusion detection. arXiv preprint arXiv:1802.09089 .

Mirzaee, P.H., Shojafar, M., Pooranian, Z., Asefy, P., Cruickshank, H., Tafazolli, R., 2021. Fids: A federated intrusion detection system for 5g smart metering network, in: 2021 17th International Conference on Mobility, Sensing and Networking (MSN), IEEE. pp. 215–222.

Mitchell, R., Chen, R., 2014. A survey of intrusion detection in wireless network applications. Computer Communications 42, 1–23.

Mohammadi, S., Amiri, F., 2019. An efficient hybrid self-learning intrusion detection system based on neural networks. International Journal of Computational Intelligence and Applications 18, 1950001. doi:10.1142/S1469026819500019.

Mohammadpour, L., Ling, T.C., Liew, C.S., Chong, C.Y., 2018. A convolutional neural network for network intrusion detection system.

Mohassel, R.R., Fung, A., Mohammadi, F., Raahemifar, K., 2014a. A survey on advanced metering infrastructure. International Journal of Electrical Power & Energy Systems 63, 473–484.

Mohassel, R.R., Fung, A.S., Mohammadi, F., Raahemifar, K., 2014b. A survey on advanced metering infrastructure and its application in smart grids, in: 2014 IEEE 27th Canadian conference on electrical and computer engineering (CCECE), IEEE. pp. 1–8.

Monrat, A.A., Schelén, O., Andersson, K., 2019. A survey of blockchain from the perspectives of applications, challenges, and opportunities. IEEE Access 7, 117134–117151. doi:10.1109/ACCESS.2019.2936094.

MontazeriShatoori, M., Davidson, L., Kaur, G., Lashkari, A.H., 2020a. Detection of doh tunnels using time-series classification of encrypted traffic. 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech) , 63–70URL: https://api.semanticscholar.org/CorpusID:226852987.

MontazeriShatoori, M., Davidson, L., Kaur, G., Lashkari, A.H., 2020b. Detection of doh tunnels using time-series classification of encrypted traffic, in: 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), IEEE. pp. 63–70.

Morris, T., Gao, W., 2014a. Industrial control system traffic data sets for intrusion detection research, in: Critical Infrastructure Protection VIII: 8th IFIP WG 11.10 International Conference, ICCIP 2014, Arlington, VA, USA, March 17-19, 2014, Revised Selected Papers 8, Springer. pp. 65–78.

Morris, T.H., Gao, W., 2014b. Industrial control system traffic data sets for intrusion detection research, in: Critical Infrastructure Protection, pp. 65–78. URL: https://api.semanticscholar.org/CorpusID:45945840.

Mothukuri, V., Khare, P., Parizi, R.M., Pouriyeh, S., Dehghantanha, A., Srivastava, G., 2021. Federated-learning-based anomaly detection for iot security attacks. IEEE Internet of Things Journal 9, 2545–2554.

Mourad, A., Otrok, H., Guizani, M., 2023. A survey on iot intrusion detection: Federated learning, game theory, social psychology, and explainable ai as future directions. IEEE Internet of Things Journal doi:10.1109/JIOT.2022.3203249.

Moustafa, N., 2019. New generations of internet of things datasets for cybersecurity applications based machine learning: Ton_iot datasets, in: Proceedings of the eResearch Australasia Conference, Brisbane, Australia, pp. 21–25.

Moustafa, N., 2021a. A new distributed architecture for evaluating ai-based security systems at the edge: Network ton_iot datasets. Sustainable Cities and Society 72, 102994.

Moustafa, N., 2021b. A systemic iot–fog–cloud architecture for big-data analytics and cyber security systems: A review of fog computing. Secure Edge Computing , 41–50.

Moustafa, N., Ahmed, M., Ahmed, S., 2020a. Data analytics-enabled intrusion detection: Evaluations of ton_iot linux datasets, in: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE. pp. 727–735.

Moustafa, N., Creech, G., Slay, J., 2017a. Big data analytics for intrusion detection system: Statistical decision-making using finite dirichlet mixture models, in: Data analytics and decision support for cybersecurity. Springer, pp. 127–156.

Moustafa, N., Keshky, M., Debiez, E., Janicke, H., 2020b. 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), IEEE. pp. 848–855.

Moustafa, N., Slay, J., 2015. 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), IEEE. pp. 1–6.

Moustafa, N., Slay, J., 2016. The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set. Information Security Journal: A Global Perspective 25, 18–31.

Moustafa, N., Slay, J., Creech, G., 2017b. Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks. IEEE Transactions on Big Data 5, 481–494.

Mukkamala, S., Janoski, G., Sung, A., 2002. Intrusion detection using neural networks and support vector machines, in: Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290), IEEE. pp. 1702–1707.

Nakamoto, S., 2009. Bitcoin: A peer-to-peer electronic cash system. Cryptography Mailing list at https://metzdowd.com .

Nasirigerdeh, R., Bakhtiari, M., Torkzadehmahani, R., Bayat, A., List, M., Blumenthal, D.B., Baumbach, J., 2020. Federated multi-mini-batch: an efficient training approach to federated learning in non-iid environments. arXiv preprint arXiv:2011.07006 .

Nguyen, T.D., Marchal, S., Miettinen, M., Fereidooni, H., Asokan, N., Sadeghi, A.R., 2019. Dïot: A federated self-learning anomaly detection system for iot, in: 2019 IEEE 39th International conference on distributed computing systems (ICDCS), IEEE. pp. 756–767.

Nguyen, T.D., Rieger, P., Miettinen, M., Sadeghi, A.R., 2020. Poisoning attacks on federated learning-based iot intrusion detection system, in: Proc. Workshop Decentralized IoT Syst. Secur.(DISS), pp. 1–7.

Nicopolitidis, P., Obaidat, M.S., Papadimitriou, G.I., Pomportsis, A.S., 2003. Wireless networks. John Wiley & Sons.

Niu, Z., Zhong, G., Yu, H., 2021. A review on the attention mechanism of deep learning. Neurocomputing 452, 48–62. doi:https://doi.org/10.1016/j.neucom.2021.03.091.

Novikova, E., Doynikova, E., Golubev, S., 2022. Federated learning for intrusion detection in the critical infrastructures: Vertically partitioned data use case. Algorithms 15, 104.

O'Shea, K., Nash, R., 2015. An introduction to convolutional neural networks. ArXiv e-prints .

Otoum, S., Guizani, N., Mouftah, H., 2021. Federated reinforcement learning-supported ids for iot-steered healthcare systems, in: ICC 2021-IEEE International Conference on Communications, IEEE. pp. 1–6.

Paillier, P., 1999. Public-key cryptosystems based on composite degree residuosity classes, in: International conference on the theory and applications of cryptographic techniques, Springer. pp. 223–238.

Pan, S.J., Yang, Q., 2010. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering 22, 1345–1359. doi:10.1109/TKDE.2009.191.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A., 2017. Automatic differentiation in pytorch.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830.

Phong, L.T., Aono, Y., Hayashi, T., Wang, L., Moriai, S., 2017. Privacy-preserving deep learning: Revisited and enhanced, in: International Conference on Applications and Techniques in Information Security, Springer. pp. 100–110.

Popoola, S.I., Gui, G., Adebisi, B., Hammoudeh, M., Gacanin, H., 2021. Federated deep learning for collaborative intrusion detection in heterogeneous networks, in: 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall), IEEE. pp. 1–6.

Prendki, J., 2022. What is data-centric ai? URL: https://alectio.com/2022/01/30/what-is-data-centric-ai/.

Qin, Y., Kondo, M., 2021. Federated learning-based network intrusion detection with a feature selection approach, in: 2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), IEEE. pp. 1–6.

Radoglou-Grammatikis, P., Sarigiannidis, P., Efstathopoulos, G., Lagkas, T., Fragulis, G., Sarigiannidis, A., 2021. A self-learning approach for detecting intrusions in healthcare systems, in: ICC 2021-IEEE International Conference on Communications, IEEE. pp. 1–6.

Radoglou-Grammatikis, P.I., Sarigiannidis, P.G., 2019. Securing the smart grid: A comprehensive compilation of intrusion detection and prevention systems. IEEE Access 7, 46595–46620.

Ray, P.P., 2017. An introduction to dew computing: definition, concept and implications. IEEE Access 6, 723–737.

Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., Kumar, S., McMahan, H.B., 2020. Adaptive federated optimization. arXiv preprint arXiv:2003.00295 .

Reddi, S.J., Konečnỳ, J., Richtárik, P., Póczós, B., Smola, A., 2016. Aide: Fast and communication efficient distributed optimization. arXiv preprint arXiv:1608.06879 .

Ren, A., Wu, D., Zhang, W., Terpenny, J., Liu, P., 2017. Cyber security in smart manufacturing: Survey and challenges, in: IIE Annual Conference. Proceedings, Institute of Industrial and Systems Engineers (IISE). pp. 716–721.

Resende, P.A.A., Drummond, A.C., 2018. A survey of random forest based methods for intrusion detection systems. ACM Comput. Surv. 51. URL: https://doi.org/10.1145/3178582, doi:10.1145/3178582.

Ring, M., Wunderlich, S., Grüdl, D., Landes, D., Hotho, A., 2017a. Creation of flow-based data sets for intrusion detection. Journal of Information Warfare 16, 41–54.

Ring, M., Wunderlich, S., Grüdl, D., Landes, D., Hotho, A., 2017b. Flow-based benchmark data sets for intrusion detection, in: Proceedings of the 16th European Conference on Cyber Warfare and Security. ACPI, pp. 361–369.

Ringnér, M., 2008. What is principal component analysis? Nature biotechnology 26, 303–304.

Robbins, H., Monro, S., 1951. A stochastic approximation method. The annals of mathematical statistics , 400–407.

Rodríguez-Barroso, N., Stipcich, G., Jiménez-López, D., Ruiz-Millán, J.A., Martínez-Cámara, E., González-Seco, G., Luzón, M.V., Veganzones, M.A., Herrera, F., 2020. Federated learning and differential privacy: Software tools analysis, the sherpa. ai fl framework and methodological guidelines for preserving data privacy. Information Fusion 64, 270–292.

Roth, H., Zephyr, M., Harouni, A., 2021. Federated learning with homomorphic encryption. URL: https://developer.nvidia.com/blog/federated-learning-with-homomorphic-encryption/.

Roy, S., Panaousis, E., Noakes, C., Laszka, A., Panda, S., Loukas, G., 2023. Sok: The mitre att&ck framework in research and practice. arXiv preprint arXiv:2304.07411 doi:https://doi.org/10.48550/arXiv.2304.07411.

Saadat, H., Aboumadi, A., Mohamed, A., Erbad, A., Guizani, M., 2021. Hierarchical federated learning for collaborative ids in iot applications, in: 2021 10th Mediterranean Conference on Embedded Computing (MECO), IEEE. pp. 1–6.

Sagi, O., Rokach, L., 2018. Ensemble learning: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8, e1249.

Sarhan, M., Layeghy, S., Moustafa, N., Portmann, M., 2021a. A cyber threat intelligence sharing scheme based on federated learning for network intrusion detection. URL: https://arxiv.org/abs/2111.02791, doi:10.48550/ARXIV.2111.02791.

Sarhan, M., Layeghy, S., Moustafa, N., Portmann, M., 2021b. Netflow datasets for machine learning-based network intrusion detection systems, in: International Conference on Big Data Technologies and Applications, International Wireless Internet Conference, Springer. pp. 117–135.

Sarhan, M., Layeghy, S., Moustafa, N., Portmann, M., 2021c. Towards a standard feature set of nids datasets. corr, abs/2101.11315. arXiv preprint arXiv:2101.11315 .

Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O.P., Tiwari, A., Er, M.J., Ding, W., Lin, C.T., 2017. A review of clustering techniques and developments. Neurocomputing 267, 664–681.

Schneble, W., Thamilarasu, G., 2019. Attack detection using federated learning in medical cyber-physical systems, in: 28th International conference on computer communications and networks (icccn), pp. 1–8.

Sebastian Garcia, Agustin Parmisano, .M.J.E., 2020. Iot-23: A labeled dataset with malicious and benign iot network traffic (version 1.0.0). URL: https://www.stratosphereips.org/datasets-iot23, doi:10.5281/zenodo.4743746.

Seo, E., Song, H.M., Kim, H.K., 2018. Gids: Gan based intrusion detection system for in-vehicle network, in: 2018 16th Annual Conference on Privacy, Security and Trust (PST), IEEE. pp. 1–6.

Servin, A., Kudenko, D., 2005. Multi-agent reinforcement learning for intrusion detection, in: Adaptive Agents and Multi-Agent Systems III. Adaptation and Multi-Agent Learning. Springer, pp. 211–223.

Settles, B., 2009. Active learning literature survey.

Settles, B., 2012. Active learning. Synthesis lectures on artificial intelligence and machine learning 6, 1–114.

Shaheen, M., Farooq, M.S., Umer, T., Kim, B.S., 2022. Applications of federated learning; taxonomy, challenges, and research trends. Electronics 11. URL: https://www.mdpi.com/2079-9292/11/4/670, doi:10.3390/electronics11040670.

Shamir, O., Srebro, N., Zhang, T., 2014. Communication-efficient distributed optimization using an approximate newton-type method, in: International conference on machine learning, PMLR. pp. 1000–1008.

Sharafaldin, I., Gharib, A., Lashkari, A.H., Ghorbani, A.A., 2018a. Towards a reliable intrusion detection benchmark dataset. Software Networking 2018, 177–200.

Sharafaldin, I., Gharib, A., Lashkari, A.H., Ghorbani, A.A., 2018b. Towards a reliable intrusion detection benchmark dataset. Software Networking 2018, 177–200.

Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A., 2018c. Toward generating a new intrusion detection dataset and intrusion traffic characterization, in: International Conference on Information Systems Security and Privacy, pp. 108–116. URL: https://api.semanticscholar.org/CorpusID:4707749.

Sharafaldin, I., Lashkari, A.H., Hakak, S., Ghorbani, A.A., 2019. Developing realistic distributed denial of service (ddos) attack dataset and taxonomy, in: 2019 International Carnahan Conference on Security Technology (ICCST), IEEE. pp. 1–8.

Sheather, S.J., Marron, J.S., 1990. Kernel quantile estimators. Journal of the American Statistical Association 85, 410–416.

Shi, J., Ge, B., Liu, Y., Yan, Y., Li, S., 2021. Data privacy security guaranteed network intrusion detection system based on federated learning, in: IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE. pp. 1–6.

Shingi, G., Saglani, H., Jain, P., 2021. Segmented federated learning for adaptive intrusion detection system. arXiv preprint arXiv:2107.00881 .

Shirey, R., 2000. Internet security glossary.

Shone, N., Ngoc, T.N., Phai, V.D., Shi, Q., 2018. A deep learning approach to network intrusion detection. IEEE Transactions on Emerging Topics in Computational Intelligence 2, 41–50. doi:10.1109/TETCI.2017.2772792.

Siddharth, M., 2020. Iot-ddos dataset. URL: https://www.kaggle.com/siddharthm1698/ddos-botnet-attack-on-iot-devices.

Singh, P., Gaba, G.S., Kaur, A., Hedabou, M., Gurtov, A., 2022. Dew-cloud-based hierarchical federated learning for intrusion detection in iomt. IEEE Journal of Biomedical and Health Informatics .

Siniosoglou, I., Sarigiannidis, P., Argyriou, V., Lagkas, T., Goudos, S.K., Poveda, M., 2021. Federated intrusion detection in ng-iot healthcare systems: An adversarial approach, in: ICC 2021-IEEE International Conference on Communications, IEEE. pp. 1–6.

Sivanathan, A., Gharakheili, H.H., Loi, F., Radford, A., Wijenayake, C., Vishwanath, A., Sivaraman, V., 2018. Classifying iot devices in smart environments using network traffic characteristics. IEEE Transactions on Mobile Computing 18, 1745–1759.

Song, H.M., Woo, J., Kim, H.K., 2020. In-vehicle network intrusion detection using deep convolutional neural network. Vehicular Communications 21, 100198.

Song, Y., Hyun, S., Cheong, Y.G., 2021a. Analysis of autoencoders for network intrusion detection. Sensors 21, 4294. doi:10.3390/s21134294.

Song, Y., Hyun, S., Cheong, Y.G., 2021b. Analysis of autoencoders for network intrusion detection. Sensors 21, 4294.

Stallings, W., Brown, L., Bauer, M.D., Howard, M., 2012. Computer security: principles and practice. volume 3. Pearson Upper Saddle River.

Stich, S.U., 2018. Local sgd converges fast and communicates little. arXiv preprint arXiv:1805.09767 .

Sultana, N., Chilamkurti, N., Peng, W., Alhadad, R., 2019. Survey on sdn based network intrusion detection system using machine learning approaches. Peer-to-Peer Networking and Applications 12, 493–501.

Sun, S., 2013. A survey of multi-view machine learning. Neural Computing and Applications 23. doi:10.1007/s00521-013-1362-6.

Sun, X., Tang, Z., Du, M., Deng, C., Lin, W., Chen, J., Qi, Q., Zheng, H., 2022. A hierarchical federated learning-based intrusion detection system for 5g smart grids. Electronics 11, 2627.

Sun, Y., Esaki, H., Ochiai, H., 2021. Adaptive intrusion detection in the networking of large-scale lans with segmented federated learning. IEEE Open Journal of the Communications Society 2, 102–112. doi:10.1109/OJCOMS.2020.3044323.

Sun, Y., Ochiai, H., Esaki, H., 2020. Intrusion detection with segmented federated learning for large-scale multiple lans, in: 2020 International Joint Conference on Neural Networks (IJCNN), IEEE. pp. 1–8.

Sutton, R.S., Barto, A.G., 2018. Reinforcement learning: An introduction. MIT press.

Swenson, B., Murray, R., Kar, S., Poor, H.V., 2020. Distributed stochastic gradient descent: Nonconvexity, nonsmoothness, and convergence to local minima. arXiv preprint arXiv:2003.02818 .

Tabassum, A., Erbad, A., Lebda, W., Mohamed, A., Guizani, M., 2022. Fedgan-ids: Privacy-preserving ids using gan and federated learning. Computer Communications 192, 299–310.

Tahir, B., Jolfaei, A., Tariq, M., 2021. Experience driven attack design and federated learning based intrusion detection in industry 4.0. IEEE Transactions on Industrial Informatics .

Tang, Z., Hu, H., Xu, C., 2022. A federated learning method for network intrusion detection. Concurrency and Computation: Practice and Experience 34, e6812.

Tao, P., Sun, Z., Sun, Z., 2018. An improved intrusion detection algorithm based on ga and svm. Ieee Access 6, 13624–13631.

Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A., 2009. A detailed analysis of the kdd cup 99 data set, in: 2009 IEEE symposium on computational intelligence for security and defense applications, Ieee. pp. 1–6.

Teixeira, D., Assunção, L., Pereira, T., Malta, S., Pinto, P., 2019. Ossec ids extension to improve log analysis and override false positive or negative detections. Journal of Sensor and Actuator Networks 8, 46.

Teixeira, M.A., Salman, T., Zolanvari, M., Jain, R., Meskin, N., Samaka, M., 2018. Scada system testbed for cybersecurity research using machine learning approach. Future Internet 10, 76.

Thakkar, A., Lohiya, R., 2023. Fusion of statistical importance for feature selection in deep neural network-based intrusion detection system. Information Fusion 90, 353–363.

Thamilarasu, G., Odesile, A., Hoang, A., 2020. An intrusion detection system for internet of medical things. IEEE Access 8, 181560–181576.

Tharwat, A., 2016. Linear vs. quadratic discriminant analysis classifier: a tutorial. International Journal of Applied Pattern Recognition 3, 145–180.

Tharwat, A., Gaber, T., Ibrahim, A., Hassanien, A.E., 2017. Linear discriminant analysis: A detailed tutorial. AI communications 30, 169–190.

Thrun, S., Littman, M.L., 2000. Reinforcement learning: an introduction. AI Magazine 21, 103–103.

Tian, P., Chen, Z., Yu, W., Liao, W., 2021. Towards asynchronous federated learning based threat detection: A dc-adam approach. Computers & Security 108, 102344.

Tlc, N., 2017. Nyc taxi and limousine commission (tlc) trip record data. URL http://www. nyc. gov/html/tlc/html/about/trip record data. shtml .

Toldinas, J., Venčkauskas, A., Liutkevičius, A., Morkevičius, N., 2022. Framing network flow for anomaly detection using image recognition and federated learning. Electronics 11, 3138.

Tsukada, M., Kondo, M., Matsutani, H., 2020. A neural network-based on-device learning anomaly detector for edge devices. IEEE Transactions on Computers 69, 1027–1044.

Tuballa, M.L., Abundo, M.L., 2016. A review of the development of smart grid technologies. Renewable and Sustainable Energy Reviews 59, 710–725.

Turnipseed, I.P., 2015. A new scada dataset for intrusion detection research. Mississippi State University.

Udd, R., Asplund, M., Nadjm-Tehrani, S., Kazemtabrizi, M., Ekstedt, M., 2016. Exploiting bro for intrusion detection in a scada system, in: Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security, pp. 44–51.

University of California, 2007. Kdd cup 1999. available on: http://kdd.ics.uci.edu/databases/kddcup99/ kddcup99.html.

University of Southern California-Information Sciences Institute, 2015. Ds2os traffic traces. URL: https://www.impactcybertrust.org/dataset_view?idDataset=519, doi:10.23721/115/1354743.

Vaccari, I., Chiola, G., Aiello, M., Mongelli, M., Cambiaso, E., 2020. Mqttset, a new dataset for machine learning techniques on mqtt. Sensors 20, 6578.

Van Engelen, J.E., Hoos, H.H., 2020. A survey on semi-supervised learning. Machine learning 109, 373–440.

Vasilev, I., Slater, D., Spacagna, G., Roelants, P., Zocca, V., 2019. Python Deep Learning - Second Edition. Packt Publishing.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. CoRR abs/1706.03762. URL: http://arxiv.org/abs/1706.03762, arXiv:1706.03762.

Verma, P., Breslin, J.G., O'Shea, D., 2022. Fldid: Federated learning enabled deep intrusion detection in smart manufacturing industries. Sensors 22, 8974.

Vinayakumar, R., Soman, K.P., Poornachandran, P., 2017. Applying convolutional neural network for network intrusion detection, in: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1222–1228. doi:10.1109/ICACCI.2017.8126009.

Vishnu, S., Ramson, S.J., Jegan, R., 2020. Internet of medical things (iomt)-an overview, in: 2020 5th international conference on devices, circuits and systems (ICDCS), IEEE. pp. 101–104.

Vucovich, M., Tarcar, A., Rebelo, P., Gade, N., Porwal, R., Rahman, A., Redino, C., Choi, K., Nandakumar, D., Schiller, R., et al., 2022. Anomaly detection via federated learning. arXiv preprint arXiv:2210.06614 .

Wang, D., Chen, D., Song, B., Guizani, N., Yu, X., Du, X., 2018. From iot to 5g i-iot: The next generation iot-based intelligent algorithms and 5g technologies. IEEE Communications Magazine 56, 114–120.

doi:10.1109/MCOM.2018.1701310.

Wang, W., Hu, Y., Que, X., Gong, X., 2012. Autonomicity design in openflow based software defined networking, in: 2012 IEEE Globecom Workshops, IEEE. pp. 818–823.

Wang, Y., Zhu, H., Hei, X., Kong, Y., Ji, W., Zhu, L., 2019. An energy saving based on task migration for mobile edge computing. EURASIP Journal on Wireless Communications and Networking 2019, 1–10.

Watkins, C.J.C.H., 1989. Learning from delayed rewards.

Weinger, B., Kim, J., Sim, A., Nakashima, M., Moustafa, N., Wu, K.J., 2022. Enhancing iot anomaly detection performance for federated learning. Digital Communications and Networks .

West, J., Ventura, D., Warnick, S., 2007. A theoretical foundation for inductive transfer. Spring Research Presentation .

Wong, K., Dillabaugh, C., Seddigh, N., Nandy, B., 2017. Enhancing suricata intrusion detection system for cyber security in scada networks, in: 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), IEEE. pp. 1–5.

Wu, L., Kong, C., Hao, X., Chen, W., 2020. A short-term load forecasting method based on gru-cnn hybrid neural network model. Mathematical problems in engineering 2020, 1–10.

Wu, P., Guo, H., Buckland, R., 2019. A transfer learning approach for network intrusion detection, in: 2019 IEEE 4th International Conference on Big Data Analytics (ICBDA), pp. 281–285. doi:10.1109/ICBDA.2019.8713213.

Xie, J., Yu, F.R., Huang, T., Xie, R., Liu, J., Wang, C., Liu, Y., 2018. A survey of machine learning techniques applied to software defined networking (sdn): Research issues and challenges. IEEE Communications Surveys & Tutorials 21, 393–430.

Xu, C., Tao, D., Xu, C., 2013. A survey on multi-view learning. CoRR abs/1304.5634. URL: http://arxiv.org/abs/1304.5634, arXiv:1304.5634.

Yadav, K., Gupta, B.B., Hsu, C.H., Chui, K.T., 2021. Unsupervised federated learning based iot intrusion detection, in: 2021 IEEE 10th Global Conference on Consumer Electronics (GCCE), IEEE. pp. 298–301.

Yang, Q., Liu, Y., Chen, T., Tong, Y., 2019. Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST) 10, 1–19.

Yang, Z., Liu, X., Li, T., Wu, D., Wang, J., Zhao, Y., Han, H., 2022. A systematic literature review of methods and datasets for anomaly-based network intrusion detection. Computers & Security , 102675.

Yi, X., Bo, W., Ji, S., Saltzman, A.B., Jaehnig, E.J., Lei, J.T., Gao, Q., Zhang, B., 2023. Deep learning prediction boosts phosphoproteomics-based discoveries through improved phosphopeptide identification. bioRxiv , 2023–01.

Yin, C., Zhu, Y., Fei, J., He, X., 2017. A deep learning approach for intrusion detection using recurrent neural networks. Ieee Access 5, 21954–21961.

Yu, T., Hua, G., Wang, H., Yang, J., Hu, J., 2022. Federated-lstm based network intrusion detection method for intelligent connected vehicles, in: ICC 2022-IEEE International Conference on Communications, IEEE. pp. 4324–4329.

Zachos, G., Essop, I., Mantas, G., Porfyrakis, K., Ribeiro, J.C., Rodriguez, J., 2021. An anomaly-based intrusion detection system for internet of medical things networks. Electronics 10, 2562.

Zakariyya, I., Kalutarage, H., Al-Kadri, M.O., 2021. Memory efficient federated deep learning for intrusion detection in iot networks.

Zarpelão, B.B., Miani, R.S., Kawakani, C.T., de Alvarenga, S.C., 2017. A survey of intrusion detection in internet of things. Journal of Network and Computer Applications 84, 25–37. URL: https://www.sciencedirect.com/science/article/pii/S1084804517300802, doi:https://doi.org/10.1016/j.jnca.2017.02.009.

Zhang, J., Zulkernine, M., 2006. A hybrid network intrusion detection technique using random forests, in: First International Conference on Availability, Reliability and Security (ARES'06), pp. 8 pp.–269. doi:10.1109/ARES.2006.7.

Zhang, J., Zulkernine, M., Haque, A., 2008. Random-forests-based network intrusion detection systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 38, 649–659. doi:10.1109/TSMCC.2008.923876.

Zhang, Y., Wang, L., Sun, W., Green II, R.C., Alam, M., 2011. Distributed intrusion detection system in a multi-layer network architecture of smart grids. IEEE Transactions on Smart Grid 2, 796–808.

Zhang, Z., Zhang, Y., Guo, D., Yao, L., Li, Z., 2022. Secfednids: Robust defense for poisoning attack against federated learning-based network intrusion detection system. Future Generation Computer Systems 134, 154–169.

Zhao, R., Wang, Y., Xue, Z., Ohtsuki, T., Adebisi, B., Gui, G., 2022. Semi-supervised federated learning based intrusion detection method for internet of things. IEEE Internet of Things Journal .

Zhao, R., Yin, Y., Shi, Y., Xue, Z., 2020. Intelligent intrusion detection based on federated learning aided long short-term memory. Physical Communication 42, 101157.

Zhao, Y., Chen, J., Wu, D., Teng, J., Yu, S., 2019. Multi-task network anomaly detection using federated learning, in: Proceedings of the tenth international symposium on information and communication technology, pp. 273–279.

Zheng, Z., Xie, S., Dai, H.N., Chen, X., Wang, H., 2018. Blockchain challenges and opportunities: A survey. International Journal of Web and Grid Services 14, 352. doi:10.1504/IJWGS.2018.095647.

Zhu, H., Wang, Y., Hei, X., Ji, W., Zhang, L., 2018. A blockchain-based decentralized cloud resource scheduling architecture, in: 2018 International Conference on Networking and Network Applications (NaNA), IEEE. pp. 324–329.

Zhu, X., Goldberg, A.B., 2009. Introduction to semi-supervised learning. Synthesis lectures on artificial intelligence and machine learning 3, 1–130.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., He, Q., 2021. A comprehensive survey on transfer learning. Proceedings of the Institute of Radio Engineers 109, 43–76. doi:10.1109/JPROC.2020.3004555.

Ziller, A., Trask, A., Lopardo, A., Szymkow, B., Wagner, B., Bluemke, E., Nounahon, J.M., Passerat-Palmbach, J., Prakash, K., Rose, N., et al., 2021a. Pysyft: A library for easy federated learning, in: Federated Learning Systems. Springer, pp. 111–139.

Ziller, A., Trask, A., Lopardo, A., Szymkow, B., Wagner, B., Bluemke, E., Nounahon, J.M., Passerat-Palmbach, J., Prakash, K., Rose, N., et al., 2021b. Pysyft: A library for easy federated learning. Federated Learning Systems: Towards Next-Generation AI , 111–139.

Zimmerman, R.D., Murillo-Sánchez, C.E., 2016. Matpower. URL: https://doi.org/10.5281/zenodo.3237810, doi:10.5281/zenodo.3237810.

Table 1: Summary of Federated Intrusion Detection Systems

Literature work	Target System	Fusion Technique	Protocols	Attacks	Performance	Dataset	Software
Tang et al. Tang et al. (2022)	N/A	FedAvg	1. IP 2. TCP	1. Begin 2. DoS Hulk 3. PortScan 4. DDoS 5. DoS GoldenEye 6. FTP-Patator 7. SSH-Patator 8. DoS Slowloris 9. DoS SlowHTTPTest 10. Bot 11. Brute Force 12. XSS 13. Infiltration 14. SQL Injection 15. Heartbleed	Acc: 97.2%	CIC-IDS2017 Sharafaldin et al. (2018b)	1. PySyft Ziller, Trask, Lopardo, Szymkow, Wagner, Bluemke, Nounahon, Passerat-Palmbach, Prakash, Rose et al. (2021b) 2. PyTorch Paszke et al. (2017)
Zhao et al. Zhao et al. (2020)	Unix-like OS	FedAvg	Shell	1. Directory Traversal 2. Large Reads 3. File Deletions 4. Batch Uninstalls	Acc: 99.23%	SEA by AT&T Shannon Lab	1. TensorFlow Abadi et al. (2015) 2. scikit-learn Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, Vanderplas, Passos, Cournapeau, Brucher, Perrot and Duchesnay (2011)
Zhao et al. Zhao et al. (2019)	N/A	FedAvg	1. IP 2. TCP 3. VPN 3. Tor	1. Begin 2. DoS Hulk 3. PortScan 4. DDoS 5. DoS GoldenEye 6. FTP-Patator 7. SSH-Patator 8. DoS Slowloris 9. DoS SlowHTTPTest 10. Bot 11. Brute Force 12. XSS 13. Infiltration 14. SQL Injection 15. Heartbleed	Acc: 98.14%	1. CIC-IDS2017 Sharafaldin et al. (2018b) 2. ISCXVPN2016 Draper-Gil et al. (2016) 3. ISCXtor2016 Lashkari et al. (2017)	PyTorch Paszke et al. (2017)
Mothukuri et al. Mothukuri et al. (2021)	1. IoT 2. IIoT	FedAvg	1. Modbus 2. RTU 3. IP 4. TCP 5. MQTT	1. MITM 2. Ping DDoS Flood 3. Modbus Query Flood 4. SYN DDoS	Acc: 90.25%	N/A	1. PySyft Ziller et al. (2021b) 2. PyTorch Paszke et al. (2017)

B. Li et al. Li et al. (2021a)	1. Industrial CPS 2. SCADA	N/A	Modbus	1. Reconnaissance 2. Response Injection 3. Command Injection 4. DoS 5. Eaveasdropping of data resources / model parameters	Acc: >99%	N/A	1. Keras Chollet et al. (2015) 2. Flask Grinberg (2018b)
Cetin et al. Cetin et al. (2019)	Wi-Fi	FedAvg	802.11	1. Injection 2. Impersonation 3. Flood	Acc: ∼ 83%	AWID Dataset Kolias et al. (2015)	LEAF Caldas et al. (2018)
Abdel-Basset et al. Abdel-Basset et al. (2021)	1. IoT 2. IoV 3. STS STS	N/A	MQTT	1. Scanning 2. DoS 3. DDoS 4. Ransomware 5. Backdoor 6. Injection 7. XSS 8. Password Cracking 9. MITM 10. Fuzzy 11. Spoofing Drive Gear 12. Spoofind RPM gauze	Acc: ∼ 92.5% (mean) @ TON_IoT ∼ 97.2% (mean) @ Car-Hack	1. TON_IoT Moustafa (2021a), Booij et al. (2021), Alsaedi et al. (2020), Moustafa et al. (2020b), Moustafa et al. (2020a), Moustafa (2019), Moustafa (2021b), Ashraf et al. (2021) 2. Car-Hacking Song et al. (2020)	PySyft Ziller et al. (2021b)
Chen et al. Chen et al. (2020)	1. IoT 2. WEN	FedAGRU	LEACH	1. Begin 2. DoS Hulk 3. PortScan 4. DDoS 5. DoS GoldenEye 6. FTP-Patator 7. SSH-Patator 8. DoS Slowloris 9. DoS SlowHTTPTest 10. Bot 11. Brute Force 12. XSS 13. Infiltration 14. SQL Injection 15. Heartbleed	Acc: 99.28% (IID) 98.82% (non-IID)	1. KDDCup99 University of California 2. CIC-IDS2017 Sharafaldin et al. (2018b) 3. WSN-DS Almomani et al. (2016)	PySyft Ziller et al. (2021b)
Attota et al. Attota et al. (2021)	1. IoT 2. IIoT	FedAvg	MQTT	1. Scanning 2. Brute Force	Acc: 98%	MQTT-IoT-IDS2020 Hindy et al. (2021)	1. PySyft Ziller et al. (2021b) 2. PyTorch Paszke et al. (2017)
Kumar et al. Kumar et al. (2021)	MEC	N/A	MQTT	1. DoS 2. U2R 3. R2L r2l 4. Probe	Acc: 92.7%	1. CIFAR-10 Krizhevsky et al. (2009) 2. KDDCup99 University of California	PySyft Ziller et al. (2021b)
Liu et al. Liu et al. (2021)	1. IoV 2. V2X	Averaging	Ethereum	1. DoS 2. U2R 3. R2L 4. Probe	Acc: >90% (depends on epochs & data size)	KDDCup99 University of California	1. PySyft Ziller et al. (2021b) 2. PyTorch Paszke et al. (2017)

Y. Fan et al. Fan et al. (2020)	1. IoT 2. MEC	Averaging	6LowPAN	1. PortScan 2. DDoS 3. FTP-Patator 4. SSH-Patator 5. Bot 6. Heartbleed 7. ARP Spoofing 8. DoS 9. Scanning 10. Mirai 11. ARP MITM 12. DoS 13. Fuzzing 14. OS Scan 15. DoS 16. Probe 17. R2L 18. U2R	Acc: 92.81%	1. CIC-IDS2017 Sharafaldin et al. (2018b) 2. NSL-KDD Tavallaee et al. (2009) 3. IoT Datasets Mirsky et al. (2018a), Kang, Ahn, Lee, Yoo, Park and Kim (2019)	N/A
Sun et al. Sun et al. (2021)	LAN	Averaging	1. IP 2. ARP 3. TCP 4. HTTP 5. HTTPS 7. UDP 8. mDNS 9. DHCP 10. Others	1. Server Message Block 2. TCP SYN Flood 3. UDP Unicast	F1: 89.3% (mean)	Custom dataset	N/A
Man et al. Man et al. (2021)	1. IoT 2. IIoT 3. MEC	FedACNN	N/A	1. DoS 2. U2R 3. R2L 4. Probing	Acc: 99.76%	NSL-KDD Tavallaee et al. (2009)	PyTorch Paszke et al. (2017)
Sun et al. Sun et al. (2020)	LAN	Averaging	1. IP 2. ARP 3. TCP 4. HTTP 5. HTTPS 7. UDP 8. mDNS 9. DHCP 10. Others	N/A	Acc: 87.1%	Custom dataset	N/A

Cheng et al. Cheng et al. (2022)	MEC	N/A	N/A	1. DoS 2. U2R 3. R2L 4. Probing 5. Fuzzers 6. Analysis 7. Backdoor 8. DoS 9. Exploits 10. Generic 11. Reconnaissance 12. shellcode & Worms	Acc: 73%	1. NSL-KDD Tavallaee et al. (2009) 2. UNSW-NB15 Moustafa and Slay (2015), Moustafa and Slay (2016), Moustafa et al. (2017b), Moustafa et al. (2017a), Sarhan et al. (2021b)	1. PySyft Ziller et al. (2021b) 2. PyTorch Paszke et al. (2017)
Shingi et al. Shingi et al. (2021)	N/A	Custom Technique	1. ICMP 2. IP 3. TCP 4. UDP	1. DoS 2. Brute Force 3. PortScan	F1: 92%	1. CIDDS-001 Ring et al. (2017b) 2. CIDDS-002 Ring et al. (2017a)	N/A
Popoola et al. Popoola et al. (2021)	1. IoT 2. IIoT 3. IoV	Fed+	1. IP 2. ARP 3. TCP 4. HTTP 5. HTTPS 6. UDP 7. mDNS 8. DHCP 9. Others	1. Backdoor 2. DoS 3. DDoS 4. SQL Injection 5. MITM 6. Password 7. Ransomware 8. Scanning 9. XSS	Acc: 99.27%	1. NF-TON-IoT-v2 Sarhan et al. (2021c) 2. NF-UNSW-NB15-v2 Sarhan et al. (2021c) 3. NF-BoT-IoT-v2 Sarhan et al. (2021c) 4. NF-CSE-CIC-IDS2018-v2 Sarhan et al. (2021c)	N/A
Dong et al. Dong et al. (2022)	N/A	N/A	1. IP 2. TCP 3. HTTPS 4. DNS	1. DoS 2. DDoS	Acc: 94.60% @ DDoS2019 88.59% @ MalDroid2020 99.54% @ Darknet2020	1. CIC-DDoS2019 Sharafaldin et al. (2019) 2. CICMalDroid2020 Mahdavifar et al. (2020), Mahdavifar, Alhadidi and Ghorbani (2022) 3. CIC-Darknet2020 Al-Hawawreh et al. (2021) 4. CIRA-CIC-DoHBrw-2020 Montazer-iShatoori, Davidson, Kaur and Lashkari (2020b)	N/A

Markovic et al. Markovic et al. (2022)	N/A	N/A	1. IP 2. TCP 3. ICMP 4. HTTP 5. DNS 6. FTP 7. FTP-DATA 8. SMTP 9. Others	1. Begin 2. DoS Hulk 3. PortScan 4. DDoS 5. DoS GoldenEye 6. FTP-Patator 7. SSH-Patator 8. DoS Slowloris 9. DoS SlowHTTPTest 10. Bot 11. Brute Force 12. XSS 13. Infiltration 14. SQL Injection 15. Heartbleed	Acc: 71.82% @ IDS2017	1. KDDCup99 University of California 2. NSL-KDD Tavallaee et al. (2009) 3. UNSW-NB15 Moustafa and Slay (2015) 4. CIC-IDS2017 Gharib et al. (2016)	N/A
Nguyen et al. Nguyen et al. (2019)	IoT	FedAvg	1. IP 2. TCP 3. WiFi	1. Pre-Infection 2. Infection 3. Scanning 4. DoS	Acc: 96.6%	Custom	Flask Grinberg (2018b)
Friha et al. Friha et al. (2022)	1. IoT 2. MEC 3. SDN 4. CPPS	FedAvg	1. IP 2. TCP 3. HTTP 4. SSH 5. MQTT	1. DoS 2. DDoS 3. Brute Force 4. Web-based 5. Infiltration 6. Botnet 7. Flood 8. MQTT Publish Flood 9. SlowITe 10. Malformed data 11. Brute Force 12. Probe 13. U2R	Acc: ∼94% @ IDS2018 ∼99% @ InSDN	1. CSE-CIC-IDS2018 Sharafaldin et al. (2018b) 2. MQTTset Vaccari et al. (2020) 3. InSDN Elsayed et al. (2020)	1. TensorFlow Abadi et al. (2015) 2. Sherpa.AI Rodríguez-Barroso, Stipcich, Jiménez-López, Ruiz-Millán, Martínez-Cámara, González-Seco, Luzón, Veganzones and Herrera (2020)
Siniosoglou et al. Siniosoglou et al. (2021)	1. IoT 2. MCPS	FedAvg	1. IP 2. TCP 3. ICMP 4. UDP 5. Others	1. Fuzzers 2. Analysis 3. Backdoor 4. DoS 5. Exploit 6. Generic 7. Reconnaissance 8. shellcode 9. Worms	Acc: 78.37%	1. CHARIS Kim et al. (2016) 2. UNSW-NB15 Moustafa and Slay (2015), Moustafa and Slay (2016), Moustafa et al. (2017b), Moustafa et al. (2017a), Sarhan et al. (2021b)	N/A
Mirzaee et al. Mirzaee et al. (2021)	1. SG 2. AMI 3. DR 4. RTP 5. SM	FedAvg	1. IP 2. TCP 3. ICMP 4. UDP 5. SMTP 6. SSH 7. HTTP 8. FTP 9. Others	1. DoS 2. Probing 3. R2L 4. U2R	Acc: 99.5%	NSL-KDD Tavallaee et al. (2009)	N/A

Yadav et al. Yadav et al. (2021)	IoT	FedAvg	1. IP 2. TCP 3. ICMP 4. UDP 5. SMTP 6. SSH 7. HTTP 8. FTP 9. Others	1. Begin 2. DoS Hulk 3. PortScan 4. DDoS 5. DoS GoldenEye 6. FTP-Patator 7. SSH-Patator 8. DoS Slowloris 9. DoS SlowHTTPTest 10. Bot 11. Brute Force 12. XSS 13. Infiltration 14. SQL Injection 15. Heartbleed	Acc: 97.75%	CIC-IDS2017 Gharib et al. (2016)	N/A
Yu et al. Yu et al. (2022)	1. IoV 2. ICV 3. ECU 4. OBU 5. IVN	FedAvg	N/A	1. DoS 2. Spoofing 3. Replay 4. Drop	Acc: ∼92% - ∼99%	OTIDS Lee et al. (2017)	N/A
Liang et al. Liang et al. (2022)	1. AMI 2. SG	FedAvg	1. IP 2. TCP 3. ICMP 4. UDP 5. SMTP 6. SSH 7. HTTP 8. FTP 9. Others	1. DoS 2. Probe 3. R2L 4. U2R	Acc: ∼99%	NSL-KDD Tavallaee et al. (2009)	PyTorch Paszke et al. (2017)
Zhao et al. Zhao et al. (2022)	IoT	Custom Technique	1. IP 2. TCP 3. ICMP 4. UDP 5. Others	N/A	Acc: ∼80% - ∼85%	N-BaIoT Meidan et al. (2018)	PyTorch Paszke et al. (2017)
Schneble et al. Schneble and Thamilarasu (2019)	MCPS	FedAvg	1. Bluetooth 2. Zigbee 3. 802.11	1. DoS 2. Data Modification 3. Data Injection 4. Eavesdropping	Acc: ∼99%	MIMIC Johnson, Pollard, Shen, Lehman, Feng, Ghassemi, Moody, Szolovits, Anthony Celi and Mark (2016)	scikit-learn Pedregosa et al. (2011)
Aouedi et al. Aouedi et al. (2022a)	1. IoT 2. IIoT	FedAvg	1. Modbus 2. SNMP 3. C37.118	N/A	Acc: ∼90%	Gas pipeline SCADA Morris and Gao (2014a)	PyTorch Paszke et al. (2017)

Shi et al. Shi et al. (2021)	N/A	N/A	1. IP 2. TCP 3. ICMP 4. UDP 5. ARP 6. HTTP 6. HTTPS 7. UDP 8. DHCP 9. Others	N/A	Acc: ∼81%	1. UNSW-NB15 Moustafa and Slay (2015), Moustafa and Slay (2016), Moustafa et al. (2017b), Moustafa et al. (2017a), Sarhan et al. (2021b) 2. CSE-CIC-IDS2018 Sharafaldin et al. (2018b)	N/A
Aouedi et al. Aouedi et al. (2022b)	IoT	FedAvg	1. IP 2. TCP 3. ICMP 4. UDP 5. Others	N/A	F1: ∼80% - ∼90%	UNSW-NB15 Moustafa and Slay (2015), Moustafa and Slay (2016), Moustafa et al. (2017b), Moustafa et al. (2017a), Sarhan et al. (2021b)	PyTorch Paszke et al. (2017)
Zakariyya et al. Zakariyya et al. (2021)	IoT	FedAvg	1. Bluetooth 2. Zigbee 3. XBee 4. 6LoWPAN	1. BASHLITE 2. Mirai 3. DDoS	Acc: ∼83% - ∼97%	1. N-BaIoT Meidan et al. (2018) 2. Kitsune Mirsky, Doitshman, Elovici and Shabtai (2018b) 3. IoT-DDoS Siddharth (2020) 4. WUSTL Teixeira et al. (2018)	1. PySyft Ziller et al. (2021b) 2. PyTorch Paszke et al. (2017)
Sun et al. Sun et al. (2022)	AMI	Custom Technique	1. IP 2. TCP 3. ICMP 4. UDP 5. Others	1. DoS 2. Probe 3. R2L 4. U2R	Acc: ∼99%	NSL-KDD Tavallaee et al. (2009)	TensorFlow Abadi et al. (2015)
Saadat et al. Saadat et al. (2021)	IoT	FedAvg	1. ZigBee 2. Bluetooth 3. RFID	1. DoS 2. Probe 3. R2L 4. U2R	Acc: ∼78%	NSL-KDD Tavallaee et al. (2009)	N/A
Yang Qin and Masaaki Kondo Qin and Kondo (2021)	IoT	FedAvg	N/A	1. DoS 2. Probe 3. R2L 4. U2R	Acc: ∼70%	NSL-KDD Tavallaee et al. (2009)	N/A
Tahir et al. Tahir et al. (2021)	IoTES	DeepFed-AA	N/A	FDIA	Acc: ∼96%	N/A	1. MATPOWER Zimmerman and Murillo-Sánchez (2016) 2. PyTorch Paszke et al. (2017)
Dong et al. Dong et al. (2021)	IoT	N/A	1. IP 2. TCP 3. HTTPS 4. DNS	DDoS	Acc: ∼65%	CIC-DDoS2019 Sharafaldin et al. (2019)	N/A

Zhang et al. Zhang et al. (2022)	IoT	N/A	1. IP 2. TCP 3. ICMP 4. UDP 5. Others	1. Label flipping 2. Clean label	Acc: ~99% @ UNSW-NB15 ~95% @ IDS2018	1. UNSW-NB15 Moustafa and Slay (2015), Moustafa and Slay (2016), Moustafa et al. (2017b), Moustafa et al. (2017a), Sarhan et al. (2021b) 2. CSE-CIC-IDS2018 Sharafaldin et al. (2018b)	PyTorch Paszke et al. (2017)
Aliyu et al. Aliyu et al. (2021)	1. CAN 2. ECU	Custom Technique	1. OBD-II 2. Bluetooth	1. Fuzzy 2. DoS 3. Impersonation 4. Attack-free state	Acc: ~95%	OTIDS Lee et al. (2017)	1. scikit-learn Pedregosa et al. (2011) 2. Ethereum Zhu et al. (2018) 3. Mininet Wang, Hu, Que and Gong (2012)
Alamleh et al. Alamleh et al. (2022)	IoMT	N/A	N/A	DDoS	N/A	1. NSL-KDD Tavallaee et al. (2009)	N/A
Novikova et al. Novikova et al. (2022)	IoT	N/A	N/A	N/A	Acc: ~97%	SWaT Goh et al. (2017)	1. PyTorch Paszke et al. (2017) 2. TensorFlow Abadi et al. (2015)
Nguyen et al. Nguyen et al. (2020)	IoT	FedAvg	N/A	1. Infection 2. Scanning 3. SYN flood 4. HTTP flood 5. various	Acc: 100%	1. DIoT-Benign Nguyen et al. (2019) 2. DIoT-Attack Nguyen et al. (2019) 3. UNSW-Benign Sivanathan et al. (2018)	PyTorch Paszke et al. (2017)
Otoum et al. Otoum et al. (2021)	IoT	N/A	1. WiFi 2. Bluetooth 3. LAN 4. Others	1. DoS 2. DDoS 3. PortScan 4. Brute Force 5. various.	Acc: ~97%	CIC-IDS2017 Sharafaldin et al. (2018b)	Simulink Documentation (2020)
Jingyi Li et al. Li et al. (2021b)	N/A	N/A	N/A	1. NTP 2. DNS 3. LDAP 4. MSSQL 5. NetBIOS 6. SNMP 7. SSDP 8. UDP 9. UDP-Lag 10. WebDDoS 11. SYN 12. TFTPDDoS 13. PortScan	Acc: ~97%	CIC-DDoS2019 Sharafaldin et al. (2019)	PyTorch Paszke et al. (2017)
Kelli et al. Kelli et al. (2021)	IoT	FedAvg	DNP3	N/A	N/A	N/A	N/A
Hei et al. Hei et al. (2020)	IoT	N/A	Ethereum	1. DoS 2. Probe 3. R2L 4. U2R	Acc: ~80% - ~97%	1. DARPA1999 Keogh et al. (2005a) 2. KDDCup99 University of California	1. Ethereum Zhu et al. (2018) 2. Fabric Antwi et al. (2021)

Huong et al. Huong et al. (2021)	1. SM 2. ICS 3. IIoT	FedAvg	MQTT	N/A	N/A	1. SCADA Turnipseed (2015) 2. Time-series Keogh, Lin and Fu (2005b)	1. TensorFlow Abadi et al. (2015) 2. FedML He, Li, So, Zeng, Zhang, Wang, Wang, Vepakomma, Singh, Qiu et al. (2020)
Tian et al. Tian et al. (2021)	1. IoT 2. CPS	Custom Technique	1. WiFi 2. Bluetooth 3. LAN 4. Others	1. Brute Force 2. DDoS 3. Web-based	Acc: ∼91% @ MNIST ∼88% @ IDS2017 ∼90% @ IoT-23 F1: ∼92% @ MNIST ∼93% @ IDS2017 ∼90% @ IoT-23	1. MNIST Deng (2012) 2. CIC-IDS2017 Sharafaldin et al. (2018b) 3. IoT-23 Garcia et al. (2020)	N/A
Weinger et al. Weinger et al. (2022)	IoT	Custom Technique	Modbus	1. PortScan 2. DoS 3. Backdoor	Acc: ∼78% - ∼95%	TON_IoT Moustafa (2021a), Booij et al. (2021), Alsaedi et al. (2020), Moustafa et al. (2020b), Moustafa et al. (2020a), Moustafa (2019), Moustafa (2021b), Ashraf et al. (2021)	N/A
Xiaolong Xu and Lianyong Qi Liu et al. (2022)	1. IoT 2. MTS 3. AIS	FedAvg	WiFi	1. DoS 2. Backdoor 3. various	Acc: ∼83% - ∼94%	Modbus Frazão et al. (2019)	TensorFlow Federated Abadi et al. (2015)
Sarhan et al. Sarhan et al. (2021a)	N/A	FedBatch	N/A	1. DoS 2. Probe 3. R2L 4. U2R	Acc: ∼80%	NSL-KDD Tavallaee et al. (2009)	PyTorch Paszke et al. (2017)
Sayan Chatterjee and Manjesh Hanawal Chatterjee and Hanawal (2021)	IoT	FedAvg	1. IP 2. TCP 3. ICMP 4. UDP 5. Others	1. DoS 2. Exploits 3. Fuzzers 4. Generic 5. Recon 6. shellcode 7. Worms	Acc: ∼90% - ∼92%	1. NF-UNSW-NB15-v2 Sarhan et al. (2021c) 2. NF-BoT-IoT-v2 Sarhan et al. (2021c)	TensorFlow Federated Abadi et al. (2015)
Toldinas et al. Toldinas et al. (2022)	N/A	Custom Technique	N/A	DDoS	Acc: ∼93%	BOUN-DDoS Erhan and Anarım (2020)	Simulink Documentation (2020)
Vucovich et al. Vucovich et al. (2022)	N/A	FedSam	N/A	DDoS	F1: ∼91%	1. CIC-IDS2017 Gharib et al. (2016) 2. CSE-CIC-IDS2018 Sharafaldin et al. (2018b) 3. MAWI Fontugne, Borgnat, Abry and Fukuda (2010)	N/A

Verma et al. Verma et al. (2022)	1. SM 2. IIoT	N/A	1. MQTT 2. CoAP 3. WebSocket	1. DoS 2. DDoS 3. Reconnaissance 4. Exploitation 5. Weaponization 6. RDoS 7. Ransomware 8. Injection 9. various	Acc: ∼99%	X-IIoTID Al-Hawawreh et al. (2021)	TensorFlow Abadi et al. (2015)
Tabassum et al. Tabassum et al. (2022)	IoMT	N/A	1. IP 2. TCP 3. ICMP 4. HTTP 5. DNS 6. FTP 7. FTP-DATA 8. SMTP 9. Others	1. Analysis 2. shellcode 3. Worms 4. Backdoor 5. Generic 6. Reconnaissance 7. Exploits 8. Fuzzers 9. DoS 10. U2R 11. R2L 12. PROBE	Acc: ∼99%	1. KDDCup99 University of California 2. NSL-KDD Tavallaee et al. (2009) 3. UNSW-NB15 Moustafa and Slay (2015)	N/A
Zhuotao Lian and Chunhua Su Lian and Su (2022)	IoT	N/A	1. HTTP 2. DNS 3. DHCP 4. Telnet 5. SSL 6. IRC	1. PortScan 2. Botnet 3. DDoS	Acc: ∼84%	IoT-23 Garcia et al. (2020)	TensorFlow Abadi et al. (2015)
Singh et al. Singh et al. (2022)	IoMT	Custom Technique	1. IP 2. TCP 3. MQTT 4. HTTP 5. DNS	1. PortScan 2. XSS 3. Ransomware 4. DDoS 5. Password 6. Injection 7. Backdoor	Acc: ∼99%	1. NSL-KDD Tavallaee et al. (2009) 2. TON_IoT Moustafa (2021a), Booij et al. (2021), Alsaedi et al. (2020), Moustafa et al. (2020b), Moustafa et al. (2020a), Moustafa (2019), Moustafa (2021b), Ashraf et al. (2021)	1. scikit-learn Pedregosa et al. (2011) 2. TensorFlow Abadi et al. (2015)