Computer Communications 195 (2022) 346–361

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Review

Federated Learning for intrusion detection system: Concepts, challenges and future directions

Shaashwat Agrawal [a], Sagnik Sarkar [a], Ons Aouedi [b], Gokul Yenduri [c], Kandaraj Piamrat [b], Mamoun Alazab [d], Sweta Bhattacharya [c], Praveen Kumar Reddy Maddikunta [c], Thippa Reddy Gadekallu [c,*]

[a] *School of Computer Science and Engineering, VIT, Vellore, India*
[b] *University of Nantes, France*
[c] *School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, India*
[d] *Charles Darwin University, Australia*

ARTICLE INFO

Keywords:
Federated Learning
Intrusion detection system
Deep Learning
Machine Learning
Anomaly detection

ABSTRACT

The rapid development of the Internet and smart devices trigger surge in network traffic making its infrastructure more complex and heterogeneous. The predominated usage of mobile phones, wearable devices and autonomous vehicles are examples of distributed networks which generate huge amount of data each and every day. The computational power of these devices have also seen steady progression which has created the need to transmit information, store data locally and drive network computations towards edge devices. Intrusion detection systems (IDS) play a significant role in ensuring security and privacy of such devices. Machine Learning (ML) and Deep Learning (DL) with Intrusion Detection Systems have gained great momentum due to their achievement of high classification accuracy. However the privacy and security aspects potentially gets jeopardized due to the need of storing and communicating data to centralized server. On the contrary, Federated Learning (FL) fits in appropriately as a privacy-preserving decentralized learning technique that does not transfer data but trains models locally and transfers the parameters to the centralized server. The present paper aims to present an extensive and exhaustive review on the use of FL in intrusion detection system. In order to establish the need for FL, various types of IDS, relevant ML approaches and its associated issues are discussed. The paper presents detailed overview of the implementation of FL in various aspects of anomaly detection. The allied challenges of FL implementations are also identified which provides idea on the scope of future direction of research. The paper finally presents the plausible solutions associated with the identified challenges in FL based intrusion detection system implementation acting as a baseline for prospective research.

Contents

* Corresponding author.
E-mail addresses: shaashwat.agrawal2018@vitstudent.ac.in (S. Agrawal), sagnik.sarkar2018@vitstudent.ac.in (S. Sarkar), ons.aouedi@ls2n.fr (O. Aouedi), gokul.yenduri@vit.ac.in (G. Yenduri), kandaraj.piamrat@ls2n.fr (K. Piamrat), alazab.m@ieee.org (M. Alazab), sweta.b@vit.ac.in (S. Bhattacharya), praveenkumarreddy@vit.ac.in (P.K.R. Maddikunta), thippareddy.g@vit.ac.in (T.R. Gadekallu).

https://doi.org/10.1016/j.comcom.2022.09.012
Received 12 July 2021; Received in revised form 29 October 2021; Accepted 8 September 2022
Available online 16 September 2022

1. Introduction

The Internet has greatly expanded in size as well as in complexity over the last decade. At the same time, new types of connected devices have emerged. As a consequence, the network systems are subjected to several security vulnerabilities including intrusions due to the increase in network complexity. According to [1], intrusions are defined as attempts to compromise the confidentiality, integrity, or availability of a computer or network, or to bypass the security mechanisms of a computer or network. Therefore, in order to efficiently mitigate different types of attacks, developing an accurate Intrusion Detection System (IDS) is the need of the hour. IDS is an important component for network security [2], which is the first line of defense in the network. To do so, IDS uses different techniques for detecting anomalies. However, traditional methods such as signature-based detection is becoming less efficient since this technique uses a signature database to recognize known attacks, and hence new attacks cannot be identified [3].

In recent years, Machine Learning (ML) including shallow models, especially Deep Learning (DL) have advanced considerably and being widely adopted in several domains such as healthcare systems [4,5], computer vision [6], and wireless communication [7]. It can provide methods to detect different types of attacks without extensive human-based intervention. For example, IDS using ML techniques learns from traffic (normal/abnormal), implementing the training process of the model. It is implemented on the dataset or the environment to identify pattern relevant to malicious traffic [8]. Although these models have been successfully used for IDS, they typically require a central entity to process the data collected from all users in the network. [9] found that the prediction accuracy decreases when network scale increases (the more the switches, the less the accuracy), due to high packet loss rate.

Motivated by the above issues, Google proposed the concept of Federated Learning (FL) for on-device learning and data privacy preservation [10]. FL enables the devices to learn in a collaborative way without the need of data sharing with a centralized server. In other words, thanks to its features, ML/DL can be trained across multiple devices and servers with decentralized data over multiple iterations [11,12]. In general, it consists of two major steps: local learning and model transmission which enables attainment of privacy preservation and cost reduction which are expected in traditional centralized machine learning methods [13]. FL is an iterative process where in each round, the whole ML/DL model can be improved. At the beginning of each round, the FL server selects a subset of clients to participate in the learning process and disseminates to them its current global model [14]. Upon receiving the global model, each client uses its own data for the local training. Then these clients send back their new model (i.e., the learned parameters) to the FL server for global aggregation (Fig. 1). This process is repeated various rounds until the desired performance is achieved. In summary, a federated learning scenario consists of two main phases: local update and global aggregation. This demonstrates the fact that with the help of FL, the clients can benefit from the other clients' data without sending their privacy-sensitive personal data to a central server [15].

In general, according to how the clients' data are distributed, FL can be divided into Horizontal Federated Learning (HFL), Vertical Federated Learning (VFL), and Federated Transfer Learning (TFL). In more details, HFL is an FL approach where datasets on the clients (i.e., devices) have the same features space with different observation. VFL (known as features-based FL) is an FL approach where the data between different domains is used to train the global model [16–18]. Here, dataset on the clients can have the same observations with different features. In addition to HFL and VFL, there is TFL architecture proposed in [19], which can be used when the dataset on the devices differ not only in instances but also in features. For further details, we refer the reader to [20].

The research community has advocated the application of FL for IDS in order to achieve anomaly detection in local devices. Consequently, the introduction of FL for IDS enables DL/ML to provide customized defense mechanisms in several devices. Moreover, it helps to alleviate the computational burden of the central processing server, preserves data privacy, improves the bandwidth utilization and further deals better with a surge of diverse data communications. Multiple number of surveys and tutorials related to FL for computer networks have been published in the recent times [21,22]. Although there has been a tremendous growth in the development of FL for IDS, there still exists lack of comprehensive survey that focusing exclusively on the applications of FL for IDS.

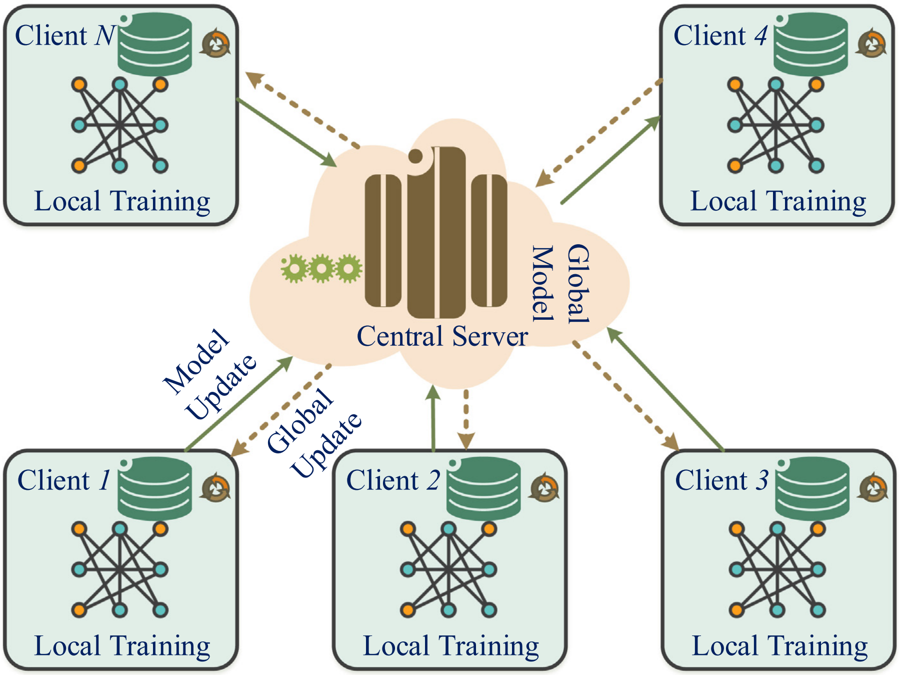


Fig. 1. General federated learning architecture.

1.1. Related work

There exists minuscule number of research publications focused exclusively on providing an overview of the use of FL in IDS. But there does exists quite a number of surveys focused on FL and IDS separately though. Hence in order to fill this gap, the present paper intends to provide a comprehensive survey of the FL technique applied to IDS. It is highly expected that the discussion and exploration could give scientific and technical readers an overall understanding of this domain and foster more subsequent studies on this issue. The earlier efforts are summarized in Table 1. In this section, the main contributions of the various works in this field are presented which helps to compare the same with our contributions.

One of the recent survey paper [3] presented and compared several deep learning-based IDS works. The authors proposed a fine-grained taxonomy, which classifies the state-of-the-art of deep learning-based IDS with respect to four different aspects: input strategy, detection strategy, deployment strategy, and performance evaluation strategy.

Kwon et al. [23] reviewed only seven deep learning-based solutions for IDS. Resende and Drummond [24] presented a comprehensive discussion of using Random Forest models for developing an efficient IDS. In addition, Sultana et al. [25] proposed a survey on SDN based network intrusion detection system using ML/DL approaches. Aburomman and Reaz [26] presented an overview of intrusion classification algorithms where various ensemble and hybrid techniques were examined, considering both homogeneous and heterogeneous types of ensemble methods. Hodo et al. [8] proposed a survey of shallow and deep networks intrusion detection systems and explained the role of feature selection in the classification and training phase of Machine Learning for IDS.

1.2. Contributions

This work aims to provide a survey on FL for IDS in terms of achievements and challenges, which makes it stand out in comparison to the previous surveys. In brief, the contributions of this work can be summarized as follows:

- It discusses the role of FL in intrusion detection.
- It describes how FL can help centralized ML/DL approaches to provide an efficient intrusion detection solution.
- It reviews the approaches and technologies that applied FL for intrusion detection.
- It presents the contribution and the shortcoming of the FL literature review for IDS.
- It highlights open research challenges and outlines possible future research directions in order to help the research community for designing efficient future solutions.

1.3. Paper organization

The rest of this paper is organized as follows. In Section 2 we give a summary of IDS, as well as the role of machine learning and artificial intelligence (ML/AI) in anomaly intrusion detection. The main purpose of Section 3 is to provide a comprehensive survey of FL applications with anomaly-based Intrusion Detection Systems. This survey also highlights open challenges and issues of using FL for IDSs as presented in Section 4. Next, Section 5 discusses future research directions for FL-based intrusion detection. Finally, Section 6 outlines the conclusions of this paper. For the purpose of better comprehension, the definitions of the abbreviations used in the paper are summarized in Table 2.

2. Background study

Ensuring reliable and trusted information exchange between different entities requires a high degree of protection. IDS assures resilient protection over distinct anomalies. It investigates numerous types of attacks, identifies malicious patterns, and supports administrators in tuning, organizing, and implementing effective controls [37,38]. This section gives a summary of IDS, as well as the role of machine learning and artificial intelligence (ML/AI) in anomaly detection. It also discusses the motivation for the use of federated learning for anomaly-based IDS.

2.1. Intrusion detection systems

An intrusion detection system (IDS) is a hardware or software system that tracks a network for unauthorized behavior or policy

Table 1
Summary of existing surveys, related to ML/DL learning, FL, and IDSs. The symbol (✓) indicate a publication is in the scope of a domain, x marks papers that do not cover that area.

Ref	Brief summary	Scope		
		DL	FL	IDS
[27]	A comprehensive deep learning survey.	✓	x	x
[28]	A recent survey on deep learning.	✓	x	x
[29]	A tutorial on deep learning	✓	x	x
[3]	A survey on the proposed deep learning-based IDSs.	✓	x	✓
[23]	An overview of anomaly detection methodologies have been introduced with the topics of data reduction, dimensionality reduction, classification, as well as a group of deep learning techniques.	✓	x	✓
[24]	A comprehensive survey on IDSs using Random Forest as a features selection method and as a classifier.	x	x	✓
[25]	A brief survey on SDN based network intrusion detection system using machine and deep learning approaches.	✓	x	✓
[30]	An overview of various deep learning approaches.	✓	x	x
[26]	An overview of intrusion classification algorithms, based on ensemble and hybrid classifiers.	x	x	✓
[20]	An introduction to the basic concept, architecture, and techniques of federated learning, and discussion of its potential in various applications.	x	✓	x
[31]	A survey on deep learning and its applications.	x	x	x
[8]	A survey of shallow and deep networks for intrusion detection systems.	✓	x	✓
[32]	A comprehensive study of Federated Learning (FL) with an emphasis on enabling software and hardware platforms, protocols, real-life applications and use-cases.	x	✓	x
[33]	An overview on integrating FL with Industrial Internet of Things in terms of privacy, resource and data management.	x	✓	x
[34]	An introduction to FL along with general applications and challenges.	x	✓	x
[13]	An introduction to FL and its related technologies	x	✓	x
[35]	A comprehensive study concerning FL's security and privacy aspects.	x	✓	x
[21]	A survey on FL in mobile edge networks.	x	✓	x
[36]	A survey on the FL challenges and the main approaches that employed FL to solve networking domain issues.	x	✓	x
Our work	**A comprehensive survey of federated learning for intrusion detection systems**	✓	✓	✓

Table 2
List of abbreviations.

Acronym	Definition
ADS	Anomaly Detection Systems
AFL	Asynchronous Federated Learning
AI	Artificial Intelligence
BNN	Binarized Neural Networks
CNN	Convolution Neural Networks
DL	Deep Learning
DAD	Deep Anomaly Detection
DDoS	Distributed denial of service
DNN	Deep Neural Network
DRL	Deep Reinforcement Learning
ENN	Encrypted Neural Network
FL	Federated Learning
GAN	Generative Adversarial Network
HIDS	Host Intrusion Detection Systems
IDS	Intrusion Detection Systems
IIoT	Industrial Internet of Things
LSTM	Long Short Term Memory
ML	Machine Learning
MLP	Multi-Layer Perceptron
MITM	Man in the Middle
NIDS	Network Intrusion Detection Systems
QoS	Quality of Service
STIN	Satellite-Terrestrial Integrated Networks
VPN	Virtual Private Network
WSN	Wireless Sensor Network
WCN	Wireless Communication Network

breaches [39,40]. The IDS logs the details related to the intrusion, initiates warnings and takes necessary mitigation or corrective measures in case an intrusion is detected as shown in the Fig. 2.

2.2. Types of IDS

Host Intrusion Detection System (HIDS) and Network Intrusion Detection System (NIDS) are two types of IDS [41]. HIDS is in-charge of monitoring a system internally, including connections to log files, user actions, and so on. NIDS, on the other hand, examines both incoming and outgoing traffic between network nodes.

2.2.1. Host Intrusion Detection System (HIDS)

HIDS maintains a record of system events to indicate occurrence of unusual activities and stays updated with device artifacts, operations, memory regions [42]. HIDS faces issues pertaining to tampering attacks and the use of anti-tampering mechanism makes its adaption reliable. A host-based IDS is not an ideal solution on its own. It suffers from severe disadvantages such as high consumption of resources that harms host's performance. Such attacks that may not be exposed unless they penetrate the host.

2.2.2. Network Intrusion Detection System (NIDS)

NIDS are deployed at a predetermined location throughout the network to scrutinize traffic from all connected networks [38]. It interprets all the traffic that passes through the sub-net and based on the comparison with anomalies library, an intrusion is identified. During the process of identifying anomalies, NIDS remains undetected while monitoring the traffic . There exists issues in scrutinizing every packet during heavy traffic failing to detect an attack at that time. The problem with the identification of virtual private networks (VPN) and understanding encrypted data add to the list of challenges.

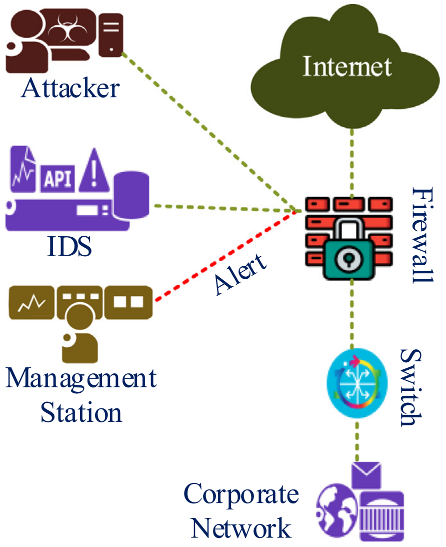


Fig. 2. Intrusion detection system.

2.3. Types of intrusion detection methods

There are primarily two approaches for IDS, signature based detection and anomaly based detection [43].

2.3.1. Signature-based intrusion detection

Signature-based intrusion detection employs pattern recognition methods to detect attacks, and they are often known as Knowledge-based Detection approaches because they keep track of patterns from past attacks. Based on the information pattern or signature, matching strategies are applied to compare known patterns from the new data. A warning signal would be activated if a pattern looks similar to a previously existing pattern. The ultimate aim is to use a library of known signatures from past attacks to identify the intrusion. Signature-based intrusion detection faces a major challenge; if an intruder uses a new attack that is not already existing in the library, the method will fail to identify the attack. This type of attack is also known as a zero-day attack.

2.3.2. Anomaly-based intrusion detection

Anomaly-based intrusion detection overcomes the challenges elevated by signature-based intrusion detection. The detection of intrusion is based on the behavior of the user. A normal model of the system's behavior is generated using statistical-based and other approaches. The difference between actual and predicted behavior is considered an anomaly. However, anomaly-based intrusion detection has various limitations. As an example, it cannot recognize encrypted packets and thus leaves an opportunity for attack. Moreover, the creation of a normal model for enormous dynamic data is extremely challenging which leads to false alarms.

2.4. ML and DL for intrusion detection

As a part of this section, some of the achievements of ML/DL in solving recent real-world intrusion detection in various applications have been discussed.

To safeguard the train ground communication systems, the authors of the study in [44] used ML algorithms, such as random forest and gradient boosted decision tree algorithms, in the primary layer. They used state observer in the second layer to detect abnormalities in the physical state of train operation. A full intrusion detection result is achieved when both layers are combined. This approach uses only two ML techniques, and there is a requirement for more contemporary methods to be applied to enhance the effectiveness of intrusion detection. The authors in [45] proposed a two-stage approach to protect Advanced Metering Infrastructure (AMI) from cyber attacks. Support Vector Machines (SVM) is used to identify suspicious behaviors within the meter . The Temporal Failure Propagation Graph (TFPG) is used to generate attack routes that identify possible cyber-attacks, where the proposed pattern recognition algorithm is used to determine the similarity between a newly identified abnormal event and pre-defined cyber attacks. However, it was designed for a precise type of data, and adaptation of the proposed technique to all types of networks is not feasible.

Gated Recurrent Unit (GRU)-based Recurrent Autoencoder (RAE) is used by the authors [46] to propose a framework named INDRA to identify anomalies in controller area network (CAN) bus-based automotive embedded systems under a variety of different attack scenarios. The adaption of the proposed IS metric for measuring the deviation of the prediction signal from the actual input still remains a concern. Smart Grids provide reliable energy to homes and industries with the help of SCADA (Supervisory Control and Data Acquisition) systems. The SCADA systems are vulnerable to cyber-attacks because they are interconnected and can be accessed remotely. To solve this issue, the authors in [47] used the gradient boosting method for feature selection along with decision-tree-based algorithms to classify the various attacks and normal events. There is a requirement for more contemporary methods to be applied to enhance the effectiveness of the framework.

Several deep learning techniques were adopted by Tian et al. [48] for proposing web attack detection systems. This system is built to safeguard multiple web applications, located across multiple sites in a distributed environment of Edge of Things (EoT). Although such system is capable of working in real-time, it still needs to adapt better deep learning algorithms for enhanced performance. Farivar et al. [49] used intelligent variable structure control to develop a new technique for estimating and compensating attacks initiated in the forward link of nonlinear Cyber-Physical Systems (CPS). A mixture of nonlinear control and artificial neural networks is used in the proposed technique as it is limited to CPS. and Industrial IoT only.

Considering the review of the recent applications, it is understood that the techniques used have significant contributions, potential and merits undoubtedly. But the adaptability of these techniques in case of other similar applications remains a concern.

2.5. Motivation to adapt FL

Even though ML and DL are making significant contributions to real-world problem solving, they have a wide range of constraints which are:

- The users need to upload their own private data to a central entity for training the central model.
- The performance decreases while network scale increases, and introduces a single point of failure that could compromise the integrity and Quality of Services (QoS).
- IDS requires fast analysis while the centralized processing is time-consuming.
- The IoT devices often collect data of end-users and can lead to exposing their sensitive data.
- Collecting the data in the 5G/6G network is a burdening and costly task since the network consists of highly diversified data types (e.g. text, audio, video, AR/VR).

It can thus be stated that FL as a technology has the potential to solve various issues. It provides data privacy and ubiquitous intelligence, seamless, flexible, scalable, high-performance communication, ultra-low latency and high-energy efficient networking to support highly dynamic time-critical applications. It is due to the existence of these features that FL provides new opportunity for ML/DL algorithms to be applied more efficiently in IDS.

3. FL for anomaly based IDS

In this section, the state-of-the-art review on FL for the use of anomaly-based IDS is exhibited. The practicality of FL architecture for the implementation of IDS is explored through key terms such as privacy, communication, data storage, and efficiency. Privacy is reviewed through the flow of data, communication through the distributed structure of FL and various types of networks, data by investigating usefulness of heterogeneous data and finally, checking efficiency of FL for low power devices. A summary of the surveyed papers is provided in Table 3.

3.1. FL for privacy preserving data handling

The core principle of an IDS is to uphold the confidentiality, integrity, and availability of the information against outside intrusions [50,51]. Training an anomaly detection statistical model requires huge amounts of data and collecting that data may cause a direct breach in privacy. In traditional Deep Anomaly Detection (DAD) Systems, two important limitations against preserving privacy are identified: presence of large amounts of unprotected data in a central location (i.e., server) and transmission or collection of said data through vulnerable channels.

3.1.1. Data at rest

Information theft is one of the leading causes of cyber attacks and intrusion attacks in any system or network. Irrespective of the level of security measures, no system is completely safe, and gathering data in a central location for the training of DAD systems can easily attract intrusion attempts. Data stealing attacks become easier when all the data is located in a single address. Common cloud intrusion attacks like wrapping, flooding, browser malware injection, and honey potting have been identified in [52]. The work analyzes such attacks in detail and describes how to deal with them. Vishal et al. [53] simulated such attacks in a cloud-based environment to exploit its security using DDoS attack, jamming attack, etc. To avoid vulnerability exploitation due to these attacks, an FL approach is to be utilized for DAD systems where the data is distributed.

The data in a federated approach is only available in the client edge devices. Unlike global datasets, federated data cannot be traced or pinpointed, its location and availability are altered dynamically due to the continuous generation of new data in an edge device. Training DAD systems in such heterogeneous environments drastically improve the confidentiality of the data as well as the accuracy of intrusion detection systems. [54] proposes a novel Federated Anomaly Detection Systems (ADS) using deep reinforcement learning that utilizes local client data effectively to train IDS. It introduces a privacy leakage degree based on leakage of sensitive and non-sensitive information to identify abnormal users and isolate them. This methodology not only prevents privacy leakage but also attains high anomaly detection accuracy in a hierarchical federated setup. Another major advantage of using federated distribution instead of centralized architecture is that incidental intrusions in one or even several client systems does not affect the system as a whole. The corruption of one or several clients could have an impact on the global model but in realistic scenarios, consisting of a minimum of 10 000 clients, the influence is negligible. Similar to [54,55] introduces a new algorithm FL-based Attention Gated Recurrent Unit (FedAGRU) for federated training of IDS that includes an attention mechanism for efficient selection of important client updates. FedAGRU was able to get relatively better accuracy and robustness to unimportant and abnormal clients.

3.1.2. Data in motion

Intrusion detection information usually includes protocol type, service type, source IP address, destination IP address, etc. Transferring these data through open communication channels or even vulnerable routes can be fatal to the users (i.e, source) as well as the server (i.e., destination). Packet transmissions can easily be monitored and altered using Man in the Middle (MITM) attacks. Unlike stationary data, data in transit is not protected by the device's security measures. Thus transferring IDS training data from client devices to a central location is not feasible. Various encryption standards [69] as well as compressed learning [70] methodologies can be used to avoid outsiders from accessing the information even though the covertness of the documents have been breached. Although these algorithms have been proven useful, they increase the communication and computation overhead. An FL approach, on the other end, trains DL models in local client devices and communicates the same to the central server. A DL model does not represent the data it trains on and hence is hard to attack. As compared to encryption/compression and huge amounts of data communication, FL proves to be very efficient. [56] evaluates such a federated approach for IDS to overcome privacy concerns for sharing sensitive data. It uses wireless communication in a simulated environment to detect intrusions without breaching privacy. Experiments on the Aegean Wi-Fi Intrusion Dataset (AWID) are conducted in this work with a good output accuracy of about 98% is attained for most classes in an FL setup.

Although difficult, a model may be able to leak some information about the source data it was trained on using model reverse engineering processes [71] or back-door attacks on DNNs [72]. To prevent these, further research on preventing such reverse engineering attacks [73] and back-door attacks [74] have been done on DL architectures. Apart from the above developments, work on FL architectures including encryption algorithms, secure communication protocols, and other federated learning-based strategies have also been done to improve the security of model transmissions. Li et al. [57] proposed a novel algorithm DeepFed that uses a federated learning approach using Convolutional Neural Networks (CNNs) and Gated Recurrent Unit (GRU) deep learning architectures for IDS in industrial cyber–physical systems. To prevent aforementioned attacks on deep learning models, DeepFed also uses a Paillier cryptosystem-based secure communication protocol that can protect the privacy of model parameters. Using the cryptosystem, a pair of public and private keys are generated. The public key encrypts the local model weights and parameters using the public key after training. The encrypted model parameters are aggregated and sent back to the clients who then use the private key to decrypt the model. Although this methodology ensures the safe passage of model parameters through encryption, introduction of advanced deciphering techniques have made decryption tasks simpler. Instead of using encryption, [75] proposes the use of mimic learning in combination with FL for better security and protection against reverse engineering attacks. In Mimic Learning, a teacher ML model is trained on a labeled private dataset. Then a student ML model is initialized and trained on an unlabeled public dataset, which is labeled by using the teacher model. Using this architecture, the probability of exposure of student model parameters reduces drastically. The work was able to achieve 98.11% accuracy using the NSL-KDD dataset with minimized security risks.

3.2. FL architecture for deployment of IDS

The efficiency of any IDS is derived from the precise selection of the deployment architecture and the type of IDS used. There are three major types of IDS deployment architectures that are implemented based on the requirements of the system: Centralized architecture, Distributed architecture, and Decentralized architecture. Centralized architecture generally proceeds in two phases, (1) collecting the data of the users on the cloud/central entity, (2) use this data to train a chosen ML/DL algorithm. Centralized deployment of IDS is usually done for small networks with low scalability. This is because this architecture demands the transmission of data to a central server in order to train

Table 3
Summary of FL literature review for IDS.

Ref.	Topic	Dataset	Contribution	Shortcomings
[53]	Fl for privacy preserving data handling	N/A	Deep Reinforcement learning for ruling out privacy vulnerable clients	Model reverse Engineering
[54]		N/A	Attention GRUs for FL	Back Door Attacks
[56]		AWID	Encryption using cryptosystems	Data loss due to lossy compression
[57]		CPS Data	IDS for Cyber Physical Systems	Vulnerable cryptosystems
[58]	FL architecture for deployment of IDS	NSL-KDD	FL framework for both MLP and autoencoder	Computation power of the server
[59]		N/A	Performance for various architectures of FL (distributed, centralized, decentralized)	Division of work at the server
[60]		N/A	Distributed GANs	Latency due to overhead operations
[61]	Fl based IDS for various Computer Networks	TER20, SAT20	Network Topology Optimization for Satellite Networks using FL	High Rate of False alarms
[62]		LAN Data	Segmented Federated Learning	High Energy Consumption
[63]		Distributed Network Data	Adaptive IDS using Segmented FL	High Energy Consumption
[64]	FL for Heterogeneous Anomaly Detection	DARPA	Ensemble type FL	Traffic masking
[65]		MNIST, FEMNIST	Semi-supervised training	Network Steganography
[66]		N/A	Multi task deep neural networks for FL	High False Positive rates
[67]	FL for DDoS attack Detection	N/A	Combines FL and fog computing to reduce mitigation time and improve detection accuracy	Only the DDoS is considered
[68]	FL for IDS in low power IoT devices	IoT Activity, Deployment, Attack Data	Self-learning Anomaly Detection System for IoT devices.	Support of low Security Measures. Low computation power High Communication Latency

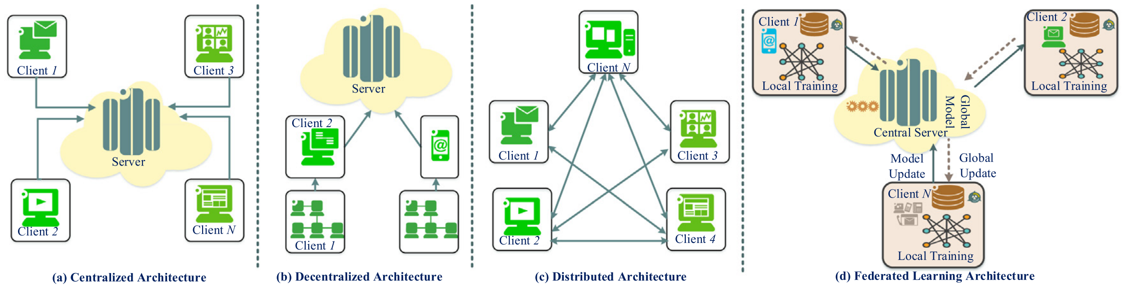
Table 4
Comparison of distributed, centralized, and FL architecture.

Architecture	Strengths	Difference with FL
Centralized	• Easy to implement. • Can be used by non expert. • Ability to find a globally optimal ML model.	• Lack of scalability and flexibility. • Important overhead and overload for data collection. • Private data must be collected in one place. • Requires large storage and computation resources.
Distributed	• Multiple computing entities. • Low latency. • Scalable.	• Lack of privacy-preserving. • Datasets exchanged between the slaves and master.

the ML model for IDS. Irrespective of the bandwidth of the network, upon increasing the number of devices, the server tends to become a bottleneck for the whole network. Unlike the centralized architecture, decentralized and distributed architectures deploy multiple IDS for active detection of attacks. They generally use the edge computing in order to reduce the network congestion and latency by training the model closer to the end users. In other words, they encompass inter-agent communication and hierarchical decision-making for better intrusion detection but are limited due to the localization of data. The differences between centralized and distributed architectures are summarized in Table 4. Work done in [76] overcomes data localization by implementing distributed Generative Adversarial Networks (GANs) that can monitor intrusions in their systems as well as neighboring systems. This literature is able to improve the distributed working of an IDS but the process may introduce latency in the network. The introduction of federated architecture can overcome the majority of the shortcomings of current IDS architectures. Different IDS deployment types are shown in Fig. 3.

FL provides a server–client architecture that involves computation at the server end (i.e., model aggregation) as well as the client end (i.e., model training). This style of work division prevents the server from becoming the bottleneck while taking advantage of edge computation. Rey et al. [58] proposed FL frameworks for both Multi-Layer Perceptron (MLP) and autoencoder models in IDS. They conducted experiments using centralized, distributed and FL architectures followed by statistical comparison between them. Their output shows the superiority of FL, which secures higher data utilization. Work done by Rahman et al. [59] evaluates the efficiency of federated architecture in IDS through NSL-KDD dataset [77]. The evaluation is done considering various real-life scenarios and intrusion attacks. A final comparison is made between FL, centralized and self-learning approaches using the obtained experimental results. FL outperformed the other approaches in almost all training rounds. Apart from efficient work division and data utilization, federated architectures are also highly scalable. The larger the federated network, the greater will be its intrusion detection capability.

Depending on the application and user demands, IDS can be deployed in large or small networks (Network Intrusion Detection Systems), on individual host systems (Host Intrusion Detection Systems), or even based on specific protocols (Protocol-based Intrusion Detection Systems). The latency caused by the implementation of a single IDS on huge networks can affect packet transmission rates drastically. A single Network Intrusion Detection System (NIDS) may not be able to handle intrusion attacks through edge devices whereas a Host Intrusion Detection System (HIDS) may not be able to protect the host from indirect intrusions through the network. [60] reviews the various types of intrusion detection systems (i.e., NIDS, HIDS, etc.), their data sets, and various limitations. To overcome these limitations and avoid the



(a) Centralized Architecture (b) Decentralized Architecture (c) Distributed Architecture (d) Federated Learning Architecture

Fig. 3. Various architectures for deployment of IDS.

increase of network latency, IDSs may employ a federated form of deployment. Federated intrusion detection systems are assisted by the size of the network and tend to maximize work division and throughput.

3.3. FL based IDS for various computer networks

Based on the scale of a network of connected devices, networks have been classified into PAN, LAN, MAN, WAN, and satellite networks [78]. These types of networks depend on the number of users, density, distance, and traffic. Therefore an FL framework for IDS needs to be tailored to the scenario, the type of data expected, the amount of generated data, and client device interaction. These factors have a high correlation with the type of attack expected and its scale.

In the recent past, satellite communication has opened up the doors to extraterrestrial and worldwide network. This has evolved as a medium for information and also for attackers to access and attack any connected device regardless of the distance between them. This makes IDS an essential service for STINs (Satellite-Terrestrial Integrated Networks). As discussed in [61], there is a huge gap in the computation resources, bandwidth, and dedicated energy between the satellite networks and the terrestrial networks. It suggests the fact that efficiency in resource usage and useful work done needs to be maximized. [61] introduces a STIN security dataset consisting of attacks on client devices, especially in DoS attacks. These attacks are a mixture of those performed in the satellite network and those performed in the terrestrial network. [79] also proposes an FL adapted STIN algorithm and a satellite network topology optimization algorithm to adapt to the IDS. This prototype has been deployed and it outperforms traditional deep learning and intrusion detection system. A low packet loss rate and lower CPU utilization rate reduce the overall energy usage making the system efficient and practical.

In a local area network setting, the number of users can range from 3 devices to a network of about 60 devices. This type of network is typically found in organizations like offices, universities, etc. [62] and [63] propose a segmented FL architecture, where instead of implementing a single server model for the federated architecture, multiple global models are assigned. This tackles the problem of non-IID data by grouping the evaluations of client models. Grouping of the models reduces the variance of the weight updates within a particular group thus promoting better model aggregation results and prediction accuracy. A dynamic metric to determine the similarity between the client models has been defined based on which the models are grouped. Careful optimization of the hyper-parameters of the models and the metrics have yielded better performances as compared to generic Federated Learning architectures.

3.4. FL for Heterogeneous Anomaly Detection

In realistic scenarios, anomaly-based IDS tend to produce high amount of false alarms caused by insufficient and unlabeled data, inefficient algorithms, etc. Intrusion attempts by attackers can be modified to avoid IDS by traffic masking, network steganography [80], and other IDS bypassing methodologies. It is pertinent to have sufficient resources for the practical deployment of IDS with low false alarms and high precision. A global dataset with constrained heterogeneity and limited size cannot be used as intrusion attempts are getting trickier and harder to detect. Although distributed IDS train on richer data, they are limited by data size. Hence, FL can be used to solve the problem of insufficient data. It allows access to data available at all client devices which at any instance may suffice the expectations of real-time protection for that network (i.e., larger network implies better protection through the use of bulkier data). Li et al. [64] propose a disagreement based semi-supervised learning for collaborative IDS. The semi-supervised learning algorithm uses three trained classifiers to label the unlabeled data based on majority voting. Each time data is labeled, the models are updated as well. This algorithm utilizes FL for tackling data size limitations and disagreement-based semi-supervised learning for handling unlabeled data. It can provide better results than supervised ML algorithms in terms of prediction accuracy as well as a lesser number of false alarms.

In case an IDS is to be deployed in a multinational company to protect its assets against intrusions, it would be difficult or practically impossible to manage the protection of each node of its network. In such huge networks, each team and the department have different data, communication, and bandwidth requirements resulting in vast dissimilarity in traffic flow properties. Even inside a department, each employee consume different amounts of data. To be more specific, on certain days it would consume more and lesser on the others. Fig. 4 depicts the heterogeneity of traffic in IDS deployments as well various applications of FL for IDS. This divergence in data is tedious to be accumulated in a single centralized model. The heterogeneity of data could cause the anomaly detection model to even fail. To handle this vast heterogeneous data, the complexity of federated architecture could be utilized. In [65], a detailed experimental study is conducted regarding the usefulness and implications of using this data in FL. It conducts experiments on state-of-the-art FL algorithms over heterogeneous and non-IID (Independent and Identically Distributed) data. Although FL shows superiority over centralized architectures, it is still challenged by the non-IID nature of data.

Another optimization approach in contrast to traditional DL training architectures is provided by [66]. It takes advantage of the heterogeneity of real-time intrusion data to implement the privacy-preserving federated architecture. More importantly, using the same data, a multi-task deep neural network in federated learning (MT-DNN-FL) can perform multiple anomaly detection tasks (i.e., VPN traffic recognition, traffic classification) using a single model. This not only conserves communication energy and training cost but also outperforms several single-task learning methods.

3.5. FL for DDoS attack detection

Distributed Denial of Service (DDoS) attacks acts as one of the major threats to the Internet. The purpose of DDoS is to overload and exhaust the target (i.e., the victims) system by using different computer systems to launch a coordinated DoS attack [81]. Consequently, DDoS attack detection exists as an important task for guaranteeing the QoS of the end-users. In this context, FL has started to attract the attention of

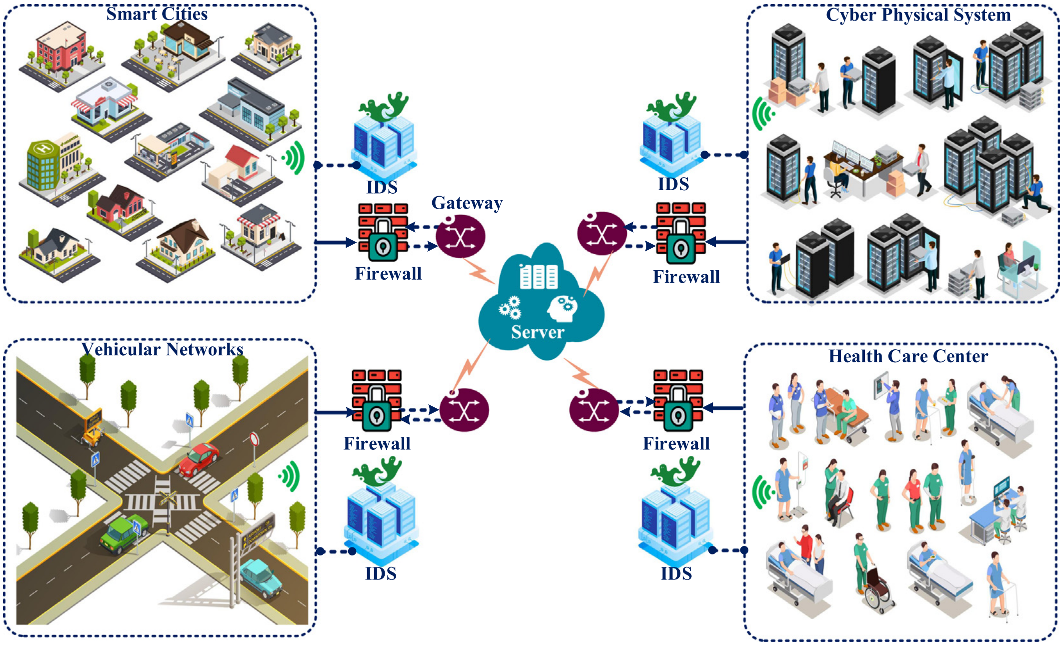


Fig. 4. FL for Heterogeneous Anomaly detection systems based on different environments and applications.

researchers for DDoS attack. However, the use of FL for IDS is still at a nascent stage.

For example, [67] proposed FL empowered mitigation architecture (FLEAM) in the Industrial Internet of Things (IIoT). FLEAM uses FL, fog, and cloud in handling large volume of DDoS attacks. Specifically, each fog node downloads a training model locally, retrains it and uploads its model parameters to the cloud. The results demonstrates that the mitigation delay is about 72% lower, and the accuracy is 47% greater on average than classic solutions.

3.6. FL for IDS in low power IoT devices

Low power IoT devices in client networks, especially edge devices are major targets of data-hungry attackers and snoopers. Be it high-end devices that are used in smart IoT like laptops and cameras, or low consumption gadgets like wearable watches and glasses, the data collected from them is confidential and sensitive making it a huge target. Due to the low power allocation by design, the processing capability of the device is limited. Limited power availability automatically results in applicability of low security measures. FL provides a helping hand in this aspect. It utilizes the distributed architecture of the network to exploit the overall computation capability and provide a stronger and robust IDS against intruders.

Interception and meticulous processing of the packets can potentially reveal anomalous behavior of devices in the network. An automated detection mechanism in this regard can help drop the suspicious packets and safeguard the device from suspicious network packets. [68] introduces a novel system, DIoT, which is a federated self-learning anomaly detection system for IoT devices. This system uses device-type-specific communication profiles without relying on human interaction and a pre-labeled dataset to train the model. In this approach, the data packets are represented as symbols in a predefined language. This representation enables the use of language analysis techniques to analyze the data packets and effectively differentiate anomalous packets from the stream of normal network packets. In this federated architecture, the anomaly detection system resides at each communication node in the network and the model aggregation takes place in a predefined server. In each anomaly detection system, the network packets entering the device are first captured and the characteristic features of the packets are extracted. These characteristic features are then mapped to the symbols of the language. These symbols are then passed through a probability computation model incorporating a recurring unit (here, GRU). The output parameters are evaluated and the packets are finally classified as Anomalous or Normal. This system has been tested against the infamous Mirai attack. Due to the self-learning nature of the system, the effectiveness of the anomaly detection system increases with time. The system adapts to specific device behavior in its due learning process; the rate of learning is high till about 2.5 to 3 h and gradually plateaus to provide a false positive rate of 0 at the 25th hour. This system demonstrates the capability of the Federated Architecture to learn from crowd-sourced data in the network and provide privacy to individual clients. DIOT provides 95.6% accuracy in the detection of anomalies in 257 ms on average; thus making it suitable for low-power IoT devices and sensor networks relying on the speed of the IDS. In addition to that, it provides exceptionally low false alarm rates in real-time scenarios.

4. Challenges

In this section, the limitations observed for FL-based IDS are explored. Although FL has shown superiority over conventional DL models for the deployment of IDS, it has its associated drawbacks and vulnerabilities [82]. Starting from the broadcast of global models to the transmission of trained models, FL for IDS is susceptible to high latency, false alarms, poisoning attacks, etc. This section thus discusses the key challenges and issues of using FL for IDS.

4.1. Communication overhead

The primary limitation of any FL application is the cost of communication per round of training [83]. Federated training demands the communication of models parameters (i.e., w^0 ← weights, η ← learning rate, B ← batch size, etc.) first from the central server to n clients and the trained models from each i_{th} client, back to the server. The server traffic, packet transmission loss, the time taken to communicate the parameters may vary greatly depending on the network bandwidth. Also, the use of different devices in a network entails that the computation capability of each is different. Considering all such aspects, it is coherent that in practical situations the overall throughput of any federated network would be low. In scenarios of using the same in intrusion detection applications, the necessity for efficiency and speed increases even more. As an example, a network with millions of devices that collect, generate, and transmit data continuously could be considered. In such case, even if a client ratio of 1% is taken, the server needs to handle a traffic load of at least 10,000 clients every broadcast. In most cases, not only does it have to wait for all clients to transmit their parameters but also aggregates the incoming clients simultaneously. Either way, the resultant server system eventually becomes a bottleneck for the federated architecture.

4.2. Federated poisoning attacks

The superiority of federated architectures comes from the distributed nature of data present in client edge devices. Although this property protects the privacy of the data in transit and avoids their collection in a central location, the data in question is still at risk. In the device of a client (i.e. the source of FL data), the labels of the data can easily be manipulated. These attacks are called poisoning attacks. If a client pretends to be a benign participant, they are capable of modifying the local and global models to produce poisoned predictions. In the worst case, the global model either collapses during training or shows a false performance against the integrity of the data it was trained on. [84] conduct poisoning attacks on FL architecture through the use of Generative Adversarial Networks (GAN). In this proposition, a GAN model is deployed to be trained upon and eventually the private data is cloned. Once trained, labels are poisoned for the data generated by the model. As more and more poisoned data is generated from the client, the amount of effect on the global modes increases. A similar work proposed by [85] introduces a novel poisoning attack for FL-based IoT intrusion detection systems. It inserts small amounts of malicious traffic in the training data to slowly create a backdoor resulting in poisoning attacks. The work proves the significance of the damage that can be caused by these attacks. The diversity of techniques in which poisoning attacks on an FL system can be performed, contributes to the difficulty in the detection and prevention of these attacks before the point of failure. The scale of the attack and the complexity of the FL system determine the extent to which the poisoned data affects the global model. The paper [86] elucidates upon such parameters and the impact they have on the overall performance.

4.3. High false alarms through the use of non-IID data

The characteristics of flowing traffic cannot simply be determined by studying some samples. Although its properties are discrete, numerous attributes are needed to analyze network traffic, many of which may play a significant role in some cases and maybe negligible in others. The complexity of intrusion detection data itself makes the task of classification tougher. On one hand, due to the abundance of data, utilization of FL is advantageous while on the other hand, the variance in the large data itself may cause improper training of the local and global models, causing a high number of false alarms. This variance is not only due to the heterogeneity of data sources but also the non-IID nature of the data. In each edge device, the size, type, ambiguity, and complexity of data vary. This results in asymmetrically trained local models. On aggregation, these models with high variance would result in the under-fitting of the global model. In addition to this, if a local client holds data that contains redundant, imbalanced, or poisoned features or values, then the respective model may even fail irrespective of the quantity of training [87]. The failure of one local model may not affect the system, but considering the practicality of real-time situations, the probability of multiple client failures is significant.

4.4. Resource management in low power IoT devices

FL has the potential to utilize low-power IoT edge devices (i.e., low-power micro-controllers, Remote Telemetry Units, sensor nodes, etc.) for training [88]. In reality, the majority of these IoT devices do not have sustainable computation power for parallel and continuous training of DL models. These low-power IoT devices hardly support programmed algorithms or even sufficient security measures. Even though this causes them to require IDS, even more, training the DL model locally may not be possible for most devices of this category. If utilized for training, they can drain out very quickly, or train slowly creating a huge latency in the server. In addition, a complete FL edge device procedure would entail data elicitation, data storage, model training, and communication apart from its regular actions. In worst-case scenarios, multiple devices could fail resulting in a deadlock on the server. Robust energy-efficient algorithms must be devised for the implementation of FL in low-power devices.

The additional disadvantage of using low power or passive devices is its ability to only support low-power communication protocols like 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Network), LPWAN (Low-Power Wide Area Network) and may not support high-end communication protocols like Wifi, Bluetooth, 5G, etc. Irrespective of the training time, the communication latency caused would also be high. Training on such devices, especially for Wireless Sensor Networks (WSN), would also encompass several natural and unpredictable situations. The situations arise from being located in harsh industrial and natural conditions [89]. The data collected by them is essential but is still not feasible to completely extract even through FL.

4.5. Vulnerabilities in intrusion detection setups

Intrusion Detection Systems may typically be installed as the first line of defense for a single workstation or a computer network. An FL architecture primarily employs DL algorithms to classify or cluster the abnormal properties of the network packets to raise an alert. This leaves the IDS susceptible to a wide array of vulnerabilities. These vulnerabilities can be deemed fatal and threatening to the Confidentiality and Integrity of the system under protection. As discussed in [90], the models that the DL models have been trained upon might not be familiar with that particular type of anomaly. Thus the detection of such an anomaly is random and can cause damage without any alarm to the respective authorities. This rate of false negatives to previously unseen data poses a threat to a centrally trained model. Moreover, the systems are left vulnerable to a Trojan horse fashion of attack due to the lack of such IDS' ability to handle a new type of anomaly. In case of even slightest deviation from the labeled anomaly, it could be treated as benign [91]. At the same time, labeling all the traffic requires huge effort of human annotators sometimes with a specific domain of expertise.

4.6. Precise deployment of IDS

Deep Learning models employ an enormous amount of parameters and hyper-parameters. Some of these parameters and hyper-parameters are user-defined and others are trainable. These user-defined parameters include learning rate, γ, number of hidden layers, dimensions of hidden layers, number of epochs per client model, initial weights, etc. Some of these parameters are model parameters while the others are training parameters. In a federated learning scenario, each client has its own model and the fine-tuning of each of these parameters is not feasible. Due to the heterogeneity and the unpredictable nature of the client data, common parameters for each model would not yield an optimal result. Furthermore, in the case of Intrusion Detection Systems, the data is highly unpredictable based on the type of attacks, source, purpose, and various other factors [92].

5. Future directions

After a critical review of FL-IDS systems and their challenges in the previous sections, the future scope of development in the field is discussed. Covering those issues, directions and possible solutions are provided for each. Considering the current literature and technological advancement on FL systems, many state of the art solutions seem feasible and applicable to the domain of IDS. Plausible solutions include the incorporation of modern communication protocols, encryption standards, blockchain, lightweight DL models, etc. A summary of the challenges faced, and their solutions proposed are presented in Table 5.

5.1. Communication efficient federated IDS

Important factors contributing to the communication overhead of a federated architecture are the transmission cost of bulky model parameters and the lag in packet transmissions. The more complex an application the bulkier the resultant global model would be. Transferring them to and from all clients for each round definitely consumes a lot of energy. To overcome this limitation various compression and encryption standards can be utilized. More importantly, lightweight DL technology can be used for reducing the size of parameters that need to be transferred as well as improve their prediction efficiency. One such algorithm named Binarized Neural Networks (BNN) is proposed by Qin et al. [93]. In this work, a BNN is trained using federated architecture for the deployment of IDS. In a BNN, the floating-point weight of a neural network is converted to binary formats containing 1's and 0's. It converts all heavy-duty computations into bit-wise operations. This methodology not only enables efficient communication but also ensures line-speed packet processing and traffic classification. Through the use of the P4 programming language, it demonstrates high classification accuracy as well as a low rate of false alarms [93].

The other major cause of communication overhead being the absence of efficient communication protocols. The bandwidth range for each communication protocol varies greatly. For example, Bluetooth transmissions under 802.11 radio protocols can reach about 24 Mbps where Wifi network bandwidths under 802.11ac can reach 1.7 Gbps. Similarly, a device employing low power protocols like LPWAN can only transmit about 400 Kbps. To make FL-based IDS more feasible, the implementation of modern communication technologies is mandatory. In proposition, [94] implemented a IDS using 5G Wireless Communication Network (WCN) for bandwidth spoofing attacks. They analyzed such attacks using game theory while optimizing power and resources. Another example of 5G based IDS is contributed in [95]. This literature introduces a new IDS framework called IoTDefender using 5G IoT and Transfer Learning. It is a one of a kind algorithm that integrates the reliability and connectivity of 5G networks with privacy preserving architecture of federated learning. It also utilizes transfer learning for providing customized learning for better detection models. They perform experiments using public CICIDS dataset as well as private datasets. The obtained accuracy is at 91.93% in addition to better reliability and privacy.

Apart from communication cost, latency in each round is also caused by traffic accumulation on the server-side. On the basis of security as well as sustainability, this traffic can be harmful to the federated system. The server at times may become overloaded with continuous traffic flow leaving it vulnerable to unknown intrusion attacks. To reduce such traffic accumulation and ensure continuous flow, Asynchronous Federated Learning (AFL) has been proposed in recent literature. In [96], ASO-Fed, an asynchronous online federated learning approach is proposed. In this algorithm, the central server gets updated in an asynchronous manner to remove system lag. The output is measured against an image dataset and three real-world datasets containing non-IID data. Through this, faster model convergence and a much higher system throughput is attained. Lu et al. [97] proposed a similar approach that involves AFL and self-adaptive threshold gradient compression for continuous and smooth communication in edge devices. They are able to implement this algorithm without the asynchronicity affecting the output performance of the models. Lastly, communication sparsification [98] has also been proposed to counter accumulating traffic on the server side. In this algorithm, certain clients are selected from the total broadcast. This selection may be done in terms of training efficiency, data quality, and overall contribution. Nevertheless, the final accuracy as well as throughput of the system are increased. Moreover, the selected client devices have optimized training that promote efficient local training iterations. Further exploration of the optimized training is presented in Section 5.5.

5.2. Encryption standards for federated learning

The major concern of FL is privacy. The transfer of the information about the data through weights and their subsequent update in the server model is a feature of FL. This transfer of weights does not contain the user data rather, they incorporate the features of this data, thus protecting the user's private information. These weights however could represent some features of the data that are sensitive and could cause harm if extracted by a third party. Moreover, the weights could contain parameters in the order of 10^5 to 10^9 or even higher than that depending on the type of the model, the shape of the inputs, outputs and many other factors. Encryption of the weights comes in handy to tackle these issues. Encryption-based privacy preservation is one of the widely used mechanisms for secure data transmission. Initially, each client receives two types of keys, which are public and private, via a secure channel [99,100]. Secondly, at each round, the clients/devices send the encrypted local model to the FL server for aggregation. Thirdly, the global model is then sent back to the client for local decryption. As a result, the clients participate to the FL process without leaking their private information. Certain types of lossy and lossless encryption functions could help in the compression of the weights. Although there might be certain information loss, the bandwidth and the speed of data transfer are increased significantly improving the overall throughput and the performance of the deep learning model [79,99].

Li et al. [79] proposed an end-to-end encrypted neural network (ENN) for the transmission of gradient updates in FL. ENN compresses and encrypts the gradient updates in the network simultaneously to provide a higher degree of confidentiality and lower charges for the transmission of the gradient updates for the weights. In this proposed framework, the clients use an encoding sub-network on the gradient updates of the model obtained while training to convert it into a lower-dimensional vector. These low-dimensional vectors are then transmitted to the server. Similar to the clients, a decoding sub-network is deployed on the server to retrieve the gradient updates for the weights to be aggregated to the server model. The encoder and the decoder parameters in the structure of ENN are predetermined and do not cause an overhead load on the communication line. The major distinguishing feature of the ENN architecture is that the encrypted weights need not be explicitly updated for the weights to be aggregated. This makes sure that the gradient updates are not revealed on either side of the communication link. Even if the communication is intercepted the extraction of information from the obtained low dimensional vectors is inherently infeasible. This provides strong confidentiality to the client data and reduces the communication overhead. In the same context, Li et al. [57] proposed an FL-based framework for intrusion detection in Industrial Cyber-Physical Systems, called *DeepFed*. DeepFed uses Homomorphic Encryption (HE) and specifically, Paillier encryption, to enable secure model sharing with the cloud server (i.e., FL server). Specifically, after the local training, the public key encrypts the local model. Then, the ciphertext is aggregated and sent back to the clients who then use the private key to decrypt the model. This makes the FL training more secure since the cloud server cannot access the plaintext of the local model. HE performs operations on a ciphertext, i.e., allows encrypted values to be involved in the computation. Although HE helps to improve the security of the FL process, it requires a large communication and computational overhead [101]. This makes HE hard to be used with lightweight devices and for real-time application. To avoid these issues, Zhang et al. [99] proposed a batch encryption-based FL framework, called *BatchCrypt*. BatchCrypt uses batch encoding and quantization schemes to get lower communication and computational costs. For further applications of HE in FL, the reader can refer to [102–104].

5.3. Edge computing in FL based IDS

Edge devices are not limited to our smartphones or laptops but extend to low-power IoT devices, sensors, etc. The processing power of each is different through which their computation support for federated architectures is reflected [105]. As an example, a high-end smartphone accomplishes training of 5 rounds in 5 units of time. At this same time, a low-end IoT device may only manage 1 or 2 rounds of training. This computational difference causes intra-structural latency. To avoid this, efficient resource management is desired in these low power edge devices. Efficient resource management and optimization algorithms have been proposed for faster and reliable edge computation using DL models.

[106] proposed a Deep Reinforcement Learning (DRL) based computation offloading optimization for IoT edge devices. Multiple DRL agents are trained and deployed using FL to create a dynamic IoT system with economical allocation of communication and computing resources. Similarly, in [107] a resource management scheme is developed that balances training time and test accuracy for stabilized overall performance. It emphasizes on data importance, communication, and computation cost to model an ideal resource management scheme. They are able to produce a 4x–10x reduction in training time while maintaining high accuracy.

A new practice of implementing lightweight DL models has also uplifted the utilization of edge computing. In [108], Doriguzzi et al. proposed a Lightweight, Usable Convolution Neural Networks (CNN) in DDoS Detection (LUCID). LUCID is a lightweight CNN developed in the work to classify benign traffic from DDoS attacks. They exploit the pattern recognition ability of CNN instead of using time-consuming feature engineering on ML algorithms for lightweight implementation. Another approach to attack detection in the IIoT using a lightweight model is provided in [109]. The work proposes a lightweight Random Neural Network (RaNN) based DL model for the classification of cybersecurity attacks. RaNNs are modified versions of ANNs that depict biological neural networks and their transmissions in humans more closely. Their predictive capabilities and distributive nature make them more suitable for resource-constrained deployments as in the case of IIoT devices. Through this approach, they were able to achieve an accuracy of about 99.20% with a quick prediction time of 34.51 ms.

5.4. Implementation of FL through secure channels

Intrusion detection systems are placed to prevent cyber attacks on the targeted systems. Deploying such IDS systems using FL increases the possibility of intrusions despite the better performance of the overall system. This is due to the creation of various vulnerable points on becoming a huge network. This is to say that no defense mechanism is full proof and as FL is integrated, the system is not only at risk from a single network channel or a virtual machine but an entire network consisting of the server and numerous clients. To overcome these, collaborative learning has adopted the use of secure communication and transaction channels like blockchain. Blockchain has emerged as a secure way of storing transactions or recording information that makes it almost impossible to hack or intrude. In [110], Alkadi et al. propose a Deep Blockchain Framework (DBF) using collaborative intrusion detection. They use a Bidirectional Long Short Term Memory (BiLSTM) DL model for training the IDS. To make data sharing simpler in the cloud, blockchain is used to store raw data alerts created by participating IDS nodes in the form of secure transactions. The algorithm is not only able to secure data from poisoning and inference attacks but also achieve good accuracy in UNSW-NB15 [111] and BoT-IoT [112] datasets. To protect federated architectures from poisoning attacks, [113] proposed a blockchain-based network that acts as the central server in this scenario. Each model update is done through blockchain transactions. By utilizing these secure transaction records, selective model aggregation is done based on improvement. The overall system is secured from poisoning attacks not only through blockchain but also through verifiable updates. A similar approach using Reputation-Based Worker Selection Scheme with Consortium Blockchain has been proposed by Kang et al. [86] to provide a reliable federated learning system that is appreciably resistant against poisoning attacks. Controlling the effect on the global model based on the reliability of a particular node in the network filters the untrustworthy data keeping the poisoning of the global model to the minimum.

An emerging technology that can be used for securing FL-based IDS systems is the digital twin technology [114]. A digital twin represents a virtual replica of real-world physical objects and systems. A digital twin simulation takes in real-world data of those objects as inputs and predicts the outputs. They are able to predict any vulnerabilities and exploitation that can be used against the system before real deployment. [115] proposed a new algorithm for the implementation of digital twin technology for anomaly detection in cyber–physical systems. They used a GCN-LSTM based GAN network for mimicking the cyber–physical systems. The generator of the GAN helps to generate real-world unlabeled data while a Timed Automation Machine (TAM) labels these data. The discriminator is the model that classifies the data as benign or anomalous. Anomaly detection with digital twin (ATTAIN) shows an average increase of 8.39% increase in prediction accuracy with respect to ADS. Similarly, other algorithms using digital twin have been developed to support vulnerabilities produced by the massive structure of FL.

5.5. Optimization of FL and IDS parameters

Federated Learning majorly employ deep learning models that utilize a wide array of trainable and user defined parameters. Moreover Intrusion Detection Systems are highly sensitive to these parameters. Optimizing these parameters has a direct relationship with better outputs and optimized training. Efforts to optimize these parameters have been made with the major focus on meta-heuristic optimization algorithms like evolutionary algorithms, nature-inspired algorithms, etc.

A multi-objective evolutionary algorithm is employed by [116] to optimize the connectivity and hyper-parameters of the artificial neural network thus simultaneously minimizing the communication costs and maximize the overall accuracy of the models. [116] used an elitist non-dominated sorting genetic algorithm as the multi-objective evolutionary algorithm. In this algorithm, a parent population is first generated. From this parent population, an offspring population is generated using mutation and crossover functions. A combination of the parent and the offspring population is sorted in a non-dominated fashion into a number of non-dominated fronts based on their dominance relationships. The crowding distance is then calculated for each individual within its front. The parent population for the next generation is then determined by selecting the better solutions front by front. The sorting criteria are directed towards increasing the learning accuracy and decreasing the communication overhead. They also proposed a Modified SET algorithm to construct a sparse neural network. This helps in reducing the total space taken by the parameters thus increasing the bandwidth and decreasing the communication costs. Similarly, [117] proposes a genetic optimization of DL parameters such as learning rate and batch size in an FL structure. Genetic, Clustered Federated Learning (CFL) algorithm clusters the client devices based on those parameters and genetically modifies them for each cluster. This results in efficient local model training due to availability of suitable hyperparameters for each client device. The algorithm was able to achieve 97.99% accuracy on MNIST data with only 10 rounds of training.

Table 5
Summary of Challenges and Future Directions for FL based IDS.

SL No	Challenges	Description	Future directions
1	Communication Overhead	• High Communication Overhead due to transmission of model parameters to and from the clients each round. • Server bottleneck caused over handling of IDS systems based on a huge network of edge devices.	• Binarized Neural Networks (BNN) are implemented in federated IDS systems for line-speed traffic classification and minimal latency [93]. • Modern Communication protocols like 5G and 6G are sought for better reliability [95]. • Asynchronous Federated Learning (AFL) methodology is proposed that ensures continuous communication and prevents central server bottleneck [96].
2	Federated Poisoning Attacks	• Manipulation of local client data and generation of poisoned data [84] • Plausible creation of backdoors in federated architecture for data poisoning attacks [85]	• Use of digital twin technology to ensure that solely intended data updates are used [114] • Using blockchain based network transactions for the federated architecture to maintain secure transaction records. These transaction records assist in selective model aggregation preventing poisoning attacks.
3	High False Alarms through the use of non-IID data	• High complexity, variance and ambiguity of federated data cause low convergence rate. • Data redundancy may result in model collapse. • Inefficient training due to above reasons of local and global model results in a high number of false alarms.	• Hierarchical clustering techniques to combat the heterogeneity by grouping similar client weight update responses which simulates homogeneity [65]. • Reinforcement learning techniques to fabricate an agent that rejects misleading weight updates due to heterogeneous data properties.
4	Resource Management in Low Power IoT Devices	• In addition to have low support for security measures, low power IoT devices also have low computation and communication capabilities. • Minimal support for high end communication protocols may cause a system lag in the federated training procedure.	• Efficient Resource Management, allocation and optimization techniques for better throughput and overall edge device performance [106]. • Implementation of lightweight ML/DL models for faster prediction speeds and lower computation costs [109].
5	Vulnerabilities in Intrusion Detection Setups	• Continuous threat to the integrity of information assets protected by the IDS. • Difficult to classify new anomalous traffic, especially in an heterogeneous environment.	• Utilization and implementation of Blockchain for secure communication between server and clients [113]. • Introduction of digital twin technology in FL based IDS systems for better trustworthiness of FL architectures [115].
6	Precise deployment of IDS	• Huge pool of parameters that need to be fine tuned. • Slight deviations might not produce optimal results. • Manual tuning is time consuming and a tedious process.	• Utilizing meta-heuristic optimization algorithms to tune the parameters mid-training [116]. • Reducing the overall number of parameters to provide fault tolerance.

5.6. *Efficient handling of non-IID data*

Federated Learning provides a personalized prediction for each client based on the experience from the data generated and usage by the clients. The data generated by the users in the real-time scenario is not homogeneous and is subject to high amounts of noise. Specifically, in the case of IDS, the properties and characteristics of the intercepted packets vary greatly based on usage, time, purpose, and many other factors. Thus it cannot be projected that the data is uniform, identical or independent of one another. These properties of the data tend to cause irregularity and issues in the training and aggregation phases of the client models. Thus handling of the non-IID data becomes an essential part of FL frameworks to ensure proper training and fault tolerance.

One of the techniques to handle non-IID data as proposed in [65] is to group the client models into smaller clusters based on the similarity of weight updates. This method of performing hierarchical clustering before training and aggregation attempt to preserve uniformity in the cluster. The results show a high convergence rate as compared to traditional FL architectures. A drop in the communication cost of over 5 times is observed to obtain the target accuracy in the FL system. A dedicated cluster head aggregates the weight updates in its locality and acts as a server for the federated framework within the cluster. The cluster heads then act as the clients for the next level of hierarchy for the federated architecture. This type of clustering can provide divided uniformity in an otherwise non-uniform data scenario. In [65], the weight updates have been used as the parameter for clustering to preserve the confidentiality of the client data. Since the weight updates represent the actual features of the data, it serves as a better parameter over other variables as criteria for clustering. The additional major advantage of hierarchical clustering is that it reduces the workload of the server by distributing this workload with the cluster heads. This increases efficiency in communication and reduces the overall communication cost. [118] proposed an experience-driven control framework, namely FAVOR. This framework focuses on intelligent device selection for aggregation in federated learning architecture. FAVOR relies on reinforcement learning to create a smart agent that selects a subset of the devices from the received weight updates during a round. This behavior of the agent counteracts the irregularity in the training caused by the non-IID data. The agent is designed and trained to maximize the accuracy of the server model at the end of every round. The filtration of the misleading weight updates trains the server model properly and converges quickly with better performance of the global model.

5.7. *FL model heterogeneity and interpretability*

Existing FL solutions mainly use the same ML/DL model (i.e., all the clients have the same model and same model architecture), where edge devices are highly heterogeneous in terms of hardware (e.g., CPU, memory). Developing the same ML/DL may not be possible on some devices. Moreover, using different models/architectures could help the FL system to benefit from this heterogeneity and hence can increase its performance. Thus, more attention should be given to this direction and study the performance of FL system with heterogeneity model or architecture [119,120].

On the other hand, while using FL to make decisions for a specific scenario it is important to gain insights into the decision-making process of the models. In other words, the interpretability of the FL model is of paramount importance. For example, when in case of FL yielding poor performance, it is necessary to understand the source and reason of the underlying problem. Therefore, it is necessary to explain the final results to find the issue or the strength of the FL system.

6. Conclusion

The enormous success of ML techniques and related applications have been guided by three major factors. Firstly, the availability of data collected through years. Secondly, the evolution of technology enabling enhanced computational power which have led to the development of flexible and integrated devices, that trains data model faster in reduced costs. Thirdly, the use of deep learning models which have enabled self-learning ensuring optimized accuracy. But there exists several challenges associated with data privacy and security. The modern world mobile phones, smart domestic and professional devices, wearables, autonomous vehicles generate humongous amount of data every single day. The privacy protection of such data and related devices is a necessity. The machine learning based Intrusion detection systems (IDS) aims to fulfill that necessity. But these frameworks still face issues to meet the privacy and security requirements due to its need to store and communicate data to the centralized server. The use of federated learning based IDS solutions train high-quality centralized model ensuring user privacy. The paper provides a comprehensive survey of FL based solutions in IDS emphasizing on the security, privacy and reliability aspects. The paper also discusses the challenges and vulnerabilities in FL implementations pertinent to high latency, false alarms, poisoning attacks and various others. The negative impact of such issues on various aspects of IDS are also described. The paper finally presents potential scope of future research and also proposes possible solutions to combat the challenges of FL implementations in IDS.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] R. Base, P. Mell, Special publication on intrusion detection systems, NIST Infidel, Inc., National Institute of Standards and Technology, Scotts Valley, CA, 2001.

[2] Y. Li, R. Ma, R. Jiao, A hybrid malicious code detection method based on deep learning, Int. J. Secur. Appl. 9 (5) (2015) 205–216.

[3] A. Aldweesh, A. Derhab, A.Z. Emam, Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues, Knowl.-Based Syst. 189 (2020) 105124.

[4] Y. Xiao, J. Wu, Z. Lin, X. Zhao, A deep learning-based multi-model ensemble method for cancer prediction, Comput. Methods Programs Biomed. 153 (2018) 1–9.

[5] O. Aouedi, M.A.B. Tobji, A. Abraham, Internet of things and ambient intelligence for mobile health monitoring: A review of a decade of research, Int. J. Comput. Inf. Syst. Ind. Manag. Appl. 10 (2018) 261–270.

[6] A. Voulodimos, N. Doulamis, A. Doulamis, E. Protopapadakis, Deep learning for computer vision: A brief review, Comput. Intell. Neurosci. 2018 (2018).

[7] C. Zhang, P. Patras, H. Haddadi, Deep learning in mobile and wireless networking: A survey, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2224–2287.

[8] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, R. Atkinson, Shallow and deep networks intrusion detection system: A taxonomy and survey, 2017, arXiv preprint arXiv:1701.02145.

[9] F. Tang, B. Mao, Z.M. Fadlullah, N. Kato, On a novel deep-learning-based intelligent partially overlapping channel assignment in SDN-IoT, IEEE Commun. Mag. 56 (9) (2018) 80–86.

[10] V. Mothukuri, R.M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, G. Srivastava, A survey on security and privacy of federated learning, Future Gener. Comput. Syst. 115 (2021) 619–640.

[11] R. Taheri, M. Shojafar, M. Alazab, R. Tafazolli, FED-IIoT: A robust federated malware detection architecture in industrial IoT, IEEE Trans. Ind. Inf. (2020).

[12] Y. Cheng, Y. Liu, T. Chen, Q. Yang, Federated learning for privacy-preserving AI, Commun. ACM 63 (12) (2020) 33–36.

[13] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, Y. Gao, A survey on federated learning, Knowl.-Based Syst. 216 (2021) 106775.

[14] V. Mothukuri, P. Khare, R.M. Parizi, S. Pouriyeh, A. Dehghantanha, G. Srivastava, Federated learning-based anomaly detection for IoT security attacks, IEEE Internet Things J. (2021).

[15] M. Gong, Y. Xie, K. Pan, K. Feng, A.K. Qin, A survey on differentially private machine learning, IEEE Comput. Intell. Mag. 15 (2) (2020) 49–64.

[16] T.R. Gadekallu, Q.-V. Pham, T. Huynh-The, S. Bhattacharya, P.K.R. Maddikunta, M. Liyanage, Federated learning for big data: A survey on opportunities, applications, and future directions, 2021, arXiv preprint arXiv:2110.04160.

[17] M. Alazab, S.P. RM, M. Parimala, P. Reddy, T.R. Gadekallu, Q.-V. Pham, Federated learning for cybersecurity: Concepts, challenges and future directions, IEEE Trans. Ind. Inf. (2021).

[18] W. Wang, M.H. Fida, Z. Lian, Z. Yin, Q.-V. Pham, T.R. Gadekallu, K. Dev, C. Su, Secure-enhanced federated learning for AI-empowered electric vehicle energy prediction, IEEE Consum. Electron. Mag. (2021).

[19] Y. Liu, Y. Kang, C. Xing, T. Chen, Q. Yang, A secure federated transfer learning framework, IEEE Intell. Syst. 35 (4) (2020) 70–82.

[20] Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: Concept and applications, ACM Trans. Intell. Syst. Technol. (TIST) 10 (2) (2019) 1–19.

[21] W.Y.B. Lim, N.C. Luong, D.T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, C. Miao, Federated learning in mobile edge networks: A comprehensive survey, IEEE Commun. Surv. Tutor. 22 (3) (2020) 2031–2063.

[22] S. Niknam, H.S. Dhillon, J.H. Reed, Federated learning for wireless communications: Motivation, opportunities, and challenges, IEEE Commun. Mag. 58 (6) (2020) 46–51.

[23] D. Kwon, H. Kim, J. Kim, S.C. Suh, I. Kim, K.J. Kim, A survey of deep learning-based network anomaly detection, Cluster Comput. 22 (1) (2019) 949–961.

[24] P.A.A. Resende, A.C. Drummond, A survey of random forest based methods for intrusion detection systems, ACM Comput. Surv. 51 (3) (2018) 1–36.

[25] N. Sultana, N. Chilamkurti, W. Peng, R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches", Peer-To-Peer Netw. Appl. 12 (2) (2019) 493–501.

[26] A.A. Aburomman, M.B.I. Reaz, A survey of intrusion detection systems based on ensemble and hybrid classifiers, Comput. Secur. 65 (2017) 135–152.

[27] J. Schmidhuber, Deep learning in neural networks: An overview, Neural Netw. 61 (2015) 85–117.

[28] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M.P. Reyes, M.-L. Shyu, S.-C. Chen, S. Iyengar, A survey on deep learning: Algorithms, techniques, and applications, ACM Comput. Surv. 51 (5) (2018) 1–36.

[29] L. Deng, A tutorial survey of architectures, algorithms, and applications for deep learning, APSIPA Trans. Signal Inf. Process. 3 (2014).

[30] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, M.S. Lew, Deep learning for visual understanding: A review, Neurocomputing 187 (2016) 27–48.

[31] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M.P. Reyes, M.-L. Shyu, S.-C. Chen, S. Iyengar, A survey on deep learning: Algorithms, techniques, and applications, ACM Comput. Surv. 51 (5) (2018) 1–36.

[32] M. Aledhari, R. Razzak, R.M. Parizi, F. Saeed, Federated learning: A survey on enabling technologies, protocols, and applications, IEEE Access 8 (2020) 140699–140725.

[33] Q.-V. Pham, K. Dev, P.K.R. Maddikunta, T.R. Gadekallu, T. Huynh-The, et al., Fusion of federated learning and industrial Internet of Things: A survey, 2021, arXiv preprint arXiv:2101.00798.

[34] T. Li, A.K. Sahu, A. Talwalkar, V. Smith, Federated learning: Challenges, methods, and future directions, IEEE Signal Process. Mag. 37 (3) (2020) 50–60.

[35] V. Mothukuri, R.M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, G. Srivastava, A survey on security and privacy of federated learning, Future Gener. Comput. Syst. 115 (2021) 619–640.

[36] O.A. Wahab, A. Mourad, H. Otrok, T. Taleb, Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems, IEEE Commun. Surv. Tutor. (2021).

[37] M. Mohammadi, T.A. Rashid, S.H.T. Karim, A.H.M. Aldalwie, Q.T. Tho, M. Bidaki, A.M. Rahmani, M. Hoseinzadeh, A comprehensive survey and taxonomy of the SVM-based intrusion detection systems, J. Netw. Comput. Appl. (2021) 102983.

[38] Z. Wu, J. Wang, L. Hu, Z. Zhang, H. Wu, A network intrusion detection method based on semantic re-encoding and deep learning, J. Netw. Comput. Appl. 164 (2020) 102688.

[39] S. Bhattacharya, P.K.R. Maddikunta, R. Kaluri, S. Singh, T.R. Gadekallu, M. Alazab, U. Tariq, et al., A novel PCA-firefly based xgboost classification model for intrusion detection in networks using GPU, Electronics 9 (2) (2020) 219.

[40] M. Alazab, R. Broadhurst, Spam and criminal activity, Trends Issues Crime Crim. Justice (526) (2016) 1–20.

[41] H. Hindy, D. Brosset, E. Bayne, A.K. Seeam, C. Tachtatzis, R. Atkinson, X. Bellekens, A taxonomy of network threats and the effect of current datasets on intrusion detection systems, IEEE Access 8 (2020) 104650–104675, http://dx.doi.org/10.1109/ACCESS.2020.3000179.

[42] S.-W. Lee, M. Mohammadi, S. Rashidi, A.M. Rahmani, M. Masdari, M. Hosseinzadeh, et al., Towards secure intrusion detection systems using deep learning techniques: Comprehensive analysis and review, J. Netw. Comput. Appl. (2021) 103111.

[43] I.F. Kilincer, F. Ertam, A. Sengur, Machine learning methods for cyber security intrusion detection: Datasets and comparative study, Comput. Netw. 188 (2021) 107840, http://dx.doi.org/10.1016/j.comnet.2021.107840, URL https://www.sciencedirect.com/science/article/pii/S1389128621000141.

[44] B. Gao, B. Bu, W. Zhang, X. Li, An intrusion detection method based on machine learning and state observer for train-ground communication systems, IEEE Trans. Intell. Transp. Syst. (2021) 1–13, http://dx.doi.org/10.1109/TITS.2021.3058553.

[45] C.-C. Sun, D.J. Sebastian Cardenas, A. Hahn, C.-C. Liu, Intrusion detection for cybersecurity of smart meters, IEEE Trans. Smart Grid 12 (1) (2021) 612–622, http://dx.doi.org/10.1109/TSG.2020.3010230.

[46] V.K. Kukkala, S.V. Thiruloga, S. Pasricha, INDRA: Intrusion detection using recurrent autoencoders in automotive embedded systems, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 39 (11) (2020) 3698–3710, http://dx.doi.org/10.1109/TCAD.2020.3012749.

[47] D. Upadhyay, J. Manero, M. Zaman, S. Sampalli, Gradient boosting feature selection with machine learning classifiers for intrusion detection on power grids, IEEE Trans. Netw. Serv. Manag. 18 (1) (2021) 1104–1116, http://dx.doi.org/10.1109/TNSM.2020.3032618.

[48] Z. Tian, C. Luo, J. Qiu, X. Du, M. Guizani, A distributed deep learning system for web attack detection on edge devices, IEEE Trans. Ind. Inf. 16 (3) (2020) 1963–1971, http://dx.doi.org/10.1109/TII.2019.2938778.

[49] F. Farivar, M.S. Haghighi, A. Jolfaei, M. Alazab, Artificial intelligence for detection, estimation, and compensation of malicious attacks in nonlinear cyber-physical systems and industrial IoT, IEEE Trans. Ind. Inf. 16 (4) (2020) 2716–2725, http://dx.doi.org/10.1109/TII.2019.2956474.

[50] A. Rehman, S.U. Rehman, M. Khan, M. Alazab, T. Reddy, Canintelliids: Detecting in-vehicle intrusion attacks on a controller area network using cnn and attention-based gru, IEEE Trans. Netw. Sci. Eng. (2021).

[51] G. Srivastava, N. Deepa, B. Prabadevi, P.K. Reddy M, An ensemble model for intrusion detection in the internet of softwarized things, in: Adjunct Proceedings of the 2021 International Conference on Distributed Computing and Networking, 2021, pp. 25–30.

[52] B.T. Devi, S. Shitharth, M. Jabbar, An appraisal over intrusion detection systems in cloud computing security attacks, in: 2020 2nd International Conference on Innovative Mechanisms for Industry Applications, ICIMIA, IEEE, 2020, pp. 722–727.

[53] V. Vishal, R. Johari, SOAiCE: simulation of attacks in cloud computing environment, in: 2018 8th International Conference on Cloud Computing, Data Science & Engineering, Confluence, IEEE, 2018, pp. 14–15.

[54] X. Wang, S. Garg, H. Lin, J. Hu, G. Kaddoum, M.J. Piran, M.S. Hossain, Towards accurate anomaly detection in industrial Internet-of-Things using hierarchical federated learning, IEEE Internet Things J. (2021).

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

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

[57] B. Li, Y. Wu, J. Song, R. Lu, T. Li, L. Zhao, DeepFed: Federated deep learning for intrusion detection in industrial cyber-physical systems, IEEE Trans. Ind. Inf. (2020).

[58] V. Rey, P.M.S. Sánchez, A.H. Celdrán, G. Bovet, M. Jaggi, Federated learning for malware detection in IoT devices, 2021, arXiv preprint arXiv:2104.09994.

[59] S.A. Rahman, H. Tout, C. Talhi, A. Mourad, Internet of things intrusion detection: Centralized, on-device, or federated learning? IEEE Netw. 34 (6) (2020) 310–317.

[60] R. Samrin, D. Vasumathi, Review on anomaly based network intrusion detection system, in: 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques, ICEECCOT, 2017, pp. 141–147, http://dx.doi.org/10.1109/ICEECCOT.2017.8284655.

[61] K. Li, H. Zhou, Z. Tu, W. Wang, H. Zhang, Distributed network intrusion detection system in satellite-terrestrial integrated networks using federated learning, IEEE Access 8 (2020) 214852–214865.

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

[63] Y. Sun, H. Esaki, H. Ochiai, Adaptive intrusion detection in the networking of large-scale LANs with segmented federated learning, IEEE Open J. Commun. Soc. (2020).

[64] W. Li, W. Meng, M.H. Au, Enhancing collaborative intrusion detection via disagreement-based semi-supervised learning in IoT environments, J. Netw. Comput. Appl. 161 (2020) 102631.

[65] C. Briggs, Z. Fan, P. Andras, Federated learning with hierarchical clustering of local updates to improve training on non-IID data, in: 2020 International Joint Conference on Neural Networks, IJCNN, IEEE, 2020, pp. 1–9.

[66] Y. Zhao, J. Chen, D. Wu, J. Teng, S. Yu, Multi-task network anomaly detection using federated learning, in: Proceedings of the Tenth International Symposium on Information and Communication Technology, 2019, pp. 273–279.

[67] J. Li, L. Lyu, X. Liu, X. Zhang, X. Lyu, FLEAM: A federated learning empowered architecture to mitigate DDoS in industrial IoT, 2020, arXiv preprint arXiv:2012.06150.

[68] T.D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, A.-R. Sadeghi, DÏoT: A federated self-learning anomaly detection system for IoT, in: 2019 IEEE 39th International Conference on Distributed Computing Systems, ICDCS, IEEE, 2019, pp. 756–767.

[69] A. Boulemtafes, A. Derhab, Y. Challal, A review of privacy-preserving techniques for deep learning, Neurocomputing 384 (2020) 21–45.

[70] E. Zisselman, A. Adler, M. Elad, Compressed learning for image classification: A deep neural network approach, in: Handbook of Numerical Analysis, Vol. 19, Elsevier, 2018, pp. 3–17.

[71] C. Gongye, Y. Fei, T. Wahl, Reverse-engineering deep neural networks using floating-point timing side-channels, in: 2020 57th ACM/IEEE Design Automation Conference, DAC, IEEE, 2020, pp. 1–6.

[72] S. Li, M. Xue, B. Zhao, H. Zhu, X. Zhang, Invisible backdoor attacks on deep neural networks via steganography and regularization, IEEE Trans. Dependable Secure Comput. (2020).

[73] Y. Liu, D. Dachman-Soled, A. Srivastava, Mitigating reverse engineering attacks on deep neural networks, in: 2019 IEEE Computer Society Annual Symposium on VLSI, ISVLSI, 2019, pp. 657–662, http://dx.doi.org/10.1109/ISVLSI.2019.00122.

[74] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, B.Y. Zhao, Neural cleanse: Identifying and mitigating backdoor attacks in neural networks, in: 2019 IEEE Symposium on Security and Privacy, SP, 2019, pp. 707–723, http://dx.doi.org/10.1109/SP.2019.00031.

[75] N.A.A.-A. Al-Marri, B.S. Ciftler, M.M. Abdallah, Federated mimic learning for privacy preserving intrusion detection, in: 2020 IEEE International Black Sea Conference on Communications and Networking, BlackSeaCom, IEEE, 2020 pp. 1–6.

[76] A. Ferdowsi, W. Saad, Generative adversarial networks for distributed intrusion detection in the internet of things, in: 2019 IEEE Global Communications Conference, GLOBECOM, IEEE, 2019, pp. 1–6.

[77] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the KDD cup 99 data set, in: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, IEEE, 2009, pp. 1–6.

[78] Y. Zhang, L. Wang, W. Sun, R.C. Green II, M. Alam, Distributed intrusion detection system in a multi-layer network architecture of smart grids, IEEE Trans. Smart Grid 2 (4) (2011) 796–808, http://dx.doi.org/10.1109/TSG.2011.2159818.

[79] H. Li, T. Han, An end-to-end encrypted neural network for gradient updates transmission in federated learning, 2019, arXiv preprint arXiv:1908.08340.

[80] S. Fathi-Kazerooni, R. Rojas-Cessa, GAN tunnel: Network traffic steganography by using GANs to counter internet traffic classifiers, IEEE Access 8 (2020) 125345–125359, http://dx.doi.org/10.1109/ACCESS.2020.3007577.

[81] P.J. Criscuolo, Distributed denial of service: Trin00, tribe flood network, tribe flood network 2000, and stacheldraht ciac-2319, Technical Report, California Univ Livermore Radiation Lab, 2000.

[82] N. Bouacida, P. Mohapatra, Vulnerabilities in federated learning, IEEE Access (2021).

[83] M. Asad, A. Moustafa, T. Ito, M. Aslam, Evaluating the communication efficiency in federated learning algorithms, in: 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design, CSCWD, IEEE, 2021, pp. 552–557.

[84] J. Zhang, J. Chen, D. Wu, B. Chen, S. Yu, Poisoning attack in federated learning using generative adversarial nets, in: 2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering, TrustCom/BigDataSE, IEEE, 2019, pp. 374–380.

[85] T.D. Nguyen, P. Rieger, M. Miettinen, A.-R. Sadeghi, Poisoning attacks on federated learning-based IoT intrusion detection system, in: NDSS Workshop on Decentralized IoT Systems and Security, 2020.

[86] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, M. Guizani, Reliable federated learning for mobile networks, IEEE Wirel. Commun. 27 (2) (2020) 72–80.

[87] C. Ma, J. Li, M. Ding, K. Wei, W. Chen, H.V. Poor, Federated learning with unreliable clients: Performance analysis and mechanism design, IEEE Internet Things J. (2021).

[88] C. Dinh, N.H. Tran, M.N. Nguyen, C.S. Hong, W. Bao, A.Y. Zomaya, V. Gramoli, Federated learning over wireless networks: Convergence analysis and resource allocation, 2019, arXiv preprint arXiv:1910.13067.

[89] A.I. Sunny, A. Zhao, L. Li, S. Kanteh Sakiliba, Low-cost IoT-based sensor system: A case study on harsh environmental monitoring, Sensors 21 (1) (2021) 214.

[90] A. Warzyński, G. Kołaczek, Intrusion detection systems vulnerability on adversarial examples, in: 2018 Innovations in Intelligent Systems and Applications, INISTA, IEEE, 2018, pp. 1–4.

[91] A. Thakkar, R. Lohiya, A review on machine learning and deep learning perspectives of IDS for IoT: recent updates, security issues, and challenges, Arch. Comput. Methods Eng. 28 (4) (2021) 3211–3243.

[92] M. Begli, F. Derakhshan, A multiagent based framework secured with layered SVM-based IDS for remote healthcare systems, 2021, arXiv preprint arXiv:2104.06498.

[93] Q. Qin, K. Poularakis, K.K. Leung, L. Tassiulas, Line-speed and scalable intrusion detection at the network edge via federated learning, in: 2020 IFIP Networking Conference (Networking), IEEE, 2020, pp. 352–360.

[94] A. Gupta, R.K. Jha, P. Gandotra, S. Jain, Bandwidth spoofing and intrusion detection system for multistage 5g wireless communication network, IEEE Trans. Veh. Technol. 67 (1) (2017) 618–632.

[95] Y. Fan, Y. Li, M. Zhan, H. Cui, Y. Zhang, IoTdefender: A federated transfer learning intrusion detection framework for 5G IoT, in: 2020 IEEE 14th International Conference on Big Data Science and Engineering, BigDataSE, IEEE, 2020, pp. 88–95.

[96] Y. Chen, Y. Ning, M. Slawski, H. Rangwala, Asynchronous online federated learning for edge devices with non-IID data, in: 2020 IEEE International Conference on Big Data, Big Data, 2020, pp. 15–24, http://dx.doi.org/10.1109/BigData50022.2020.9378161.

[97] X. Lu, Y. Liao, P. Lio, P. Hui, Privacy-preserving asynchronous federated learning mechanism for edge network computing, IEEE Access 8 (2020) 48970–48981, http://dx.doi.org/10.1109/ACCESS.2020.2978082.

[98] E. Ozfatura, K. Ozfatura, D. Gunduz, Time-correlated sparsification for communication-efficient federated learning, 2021, arXiv preprint arXiv:2101.08837.

[99] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, Y. Liu, Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning, in: 2020 $\{USENIX\}$ Annual Technical Conference ($\{USENIX\}\{ATC\}$ 20), 2020, pp. 493–506.

[100] S. Hardy, W. Henecka, H. Ivey-Law, R. Nock, G. Patrini, G. Smith, B. Thorne, Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption, 2017, arXiv preprint arXiv:1711.10677.

[101] M. Naehrig, K. Lauter, V. Vaikuntanathan, Can homomorphic encryption be practical? in: Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop, 2011, pp. 113–124.

[102] Z. Yang, S. Hu, K. Chen, FPGA-based hardware accelerator of homomorphic encryption for efficient federated learning, 2020, arXiv preprint arXiv:2007.10560.

[103] Federated ai ecosystem, 2019, [Online]. Available https://fate.fedai.org/.

[104] Y. Liu, Y. Kang, C. Xing, T. Chen, Q. Yang, A secure federated transfer learning framework, IEEE Intell. Syst. 35 (4) (2020) 70–82.

[105] B. Prabadevi, N. Deepa, Q.-V. Pham, D.C. Nguyen, T. Reddy, P.N. Pathirana, O. Dobre, et al., Toward blockchain for edge-of-things: A new paradigm, opportunities, and future directions, IEEE Internet Things Mag. (2021).

[106] J. Ren, H. Wang, T. Hou, S. Zheng, C. Tang, Federated learning-based computation offloading optimization in edge computing-supported internet of things, IEEE Access 7 (2019) 69194–69201.

[107] R. Balakrishnan, M. Akdeniz, S. Dhakal, N. Himayat, Resource management and fairness for federated learning over wireless edge networks, in: 2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications, SPAWC, 2020, pp. 1–5, http://dx.doi.org/10.1109/SPAWC48557.2020.9154285.

[108] R. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. Martí nez-del Rincon, D. Siracusa, Lucid: A practical, lightweight deep learning solution for DDoS attack detection, IEEE Trans. Netw. Serv. Manag. 17 (2) (2020) 876–889, http://dx.doi.org/10.1109/TNSM.2020.2971776.

[109] S. Latif, Z. Zou, Z. Idrees, J. Ahmad, A novel attack detection scheme for the industrial internet of things using a lightweight random neural network, IEEE Access 8 (2020) 89337–89350.

[110] O. Alkadi, N. Moustafa, B. Turnbull, K.-K.R. Choo, A deep blockchain framework-enabled collaborative intrusion detection for protecting iot and cloud networks, IEEE Internet Things J. (2020).

[111] N. Moustafa, J. Slay, UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), in: 2015 Military Communications and Information Systems Conference, MilCIS, 2015, pp. 1–6, http://dx.doi.org/10.1109/MilCIS.2015.7348942.

[112] N. Moustafa, The Bot-IoT dataset, IEEE Dataport, 2019, http://dx.doi.org/10.21227/r7v2-x988.

[113] A.R. Short, H.C. Leligou, M. Papoutsidakis, E. Theocharis, Using blockchain technologies to improve security in federated learning systems, in: 2020 IEEE 44th Annual Computers, Software, and Applications Conference, COMPSAC, 2020, pp. 1183–1188, http://dx.doi.org/10.1109/COMPSAC48688.2020.00-96.

[114] A. Fuller, Z. Fan, C. Day, C. Barlow, Digital twin: Enabling technologies, challenges and open research, IEEE Access 8 (2020) 108952–108971.

[115] Q. Xu, S. Ali, T. Yue, Digital twin-based anomaly detection in cyber-physical systems, in: 2021 14th IEEE Conference on Software Testing, Verification and Validation, ICST, IEEE, 2021, pp. 205–216.

[116] H. Zhu, Y. Jin, Multi-objective evolutionary federated learning, IEEE Trans. Neural Netw. Learn. Syst. 31 (4) (2019) 1310–1322.

[117] S. Agrawal, S. Sarkar, M. Alazab, P.K.R. Maddikunta, T.R. Gadekallu, Q.-V. Pham, Genetic CFL: Optimization of hyper-parameters in clustered federated learning, 2021, arXiv preprint arXiv:2107.07233.

[118] H. Wang, Z. Kaplan, D. Niu, B. Li, Optimizing federated learning on non-iid data with reinforcement learning, in: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, IEEE, 2020, pp. 1698–1707.

[119] C. Yang, Q. Wang, M. Xu, S. Wang, K. Bian, X. Liu, Heterogeneity-aware federated learning, 2020, arXiv preprint arXiv:2006.06983.

[120] J. Pang, Y. Huang, Z. Xie, Q. Han, Z. Cai, Realizing the heterogeneity: A self-organized federated learning framework for iot, IEEE Internet Things J. (2020).