Iraqi Journal for Computer Science and Mathematics

Volume 5 | Issue 4 Article 12

2024

Investigating Intrusion Detection System Using Federated Learning for IoT Security Challenges

Mohammed Q. Mohammed
Medical Instrumentation Engineering Department ,Al-Esraa University, Baghdad, Iraq AND University of information Technology and Communications, Iraq, Baghdad, dr.mohammed@esraa.edu.iq

Zena Abd Alrahman
Medical Instrumentation Engineering Department ,Al-Esraa University, Baghdad, Iraq

Aouf R. Shehab
Medical Instrumentation Engineering Department ,Al-Esraa University, Baghdad, Iraq

Follow this and additional works at: https://ijcsm.researchcommons.org/ijcsm

 Part of the Computer Engineering Commons

Recommended Citation

Mohammed, Mohammed Q.; Alrahman, Zena Abd; and Shehab, Aouf R. (2024) "Investigating Intrusion Detection System Using Federated Learning for IoT Security Challenges," *Iraqi Journal for Computer Science and Mathematics*: Vol. 5: Iss. 4, Article 12.
DOI: https://doi.org/10.52866/2788-7421.1218
Available at: https://ijcsm.researchcommons.org/ijcsm/vol5/iss4/12

This Original Study is brought to you for free and open access by Iraqi Journal for Computer Science and Mathematics. It has been accepted for inclusion in Iraqi Journal for Computer Science and Mathematics by an authorized editor of Iraqi Journal for Computer Science and Mathematics. For more information, please contact mohammad.aljanabi@aliraqia.edu.iq.

IRAQI JOURNAL FOR COMPUTER SCIENCE AND MATHEMATICS 2024;5:131–143

Scan the QR to view
the full-text article on
the journal website

RESEARCH ARTICLE

Investigating Intrusion Detection System Using Federated Learning for IoT Security Challenges

Mohammed Q. Mohammed [a,b,*], Zena Abd Alrahman [a], Aouf R. Shehab [a]

[a] Medical Instrumentation Engineering Department, Al-Esraa University, Baghdad, Iraq
[b] University of information Technology and Communications, Baghdad, Iraq
[c] Department of Accounting, Al-Esraa University, Baghdad, Iraq

ABSTRACT

The Internet of Things (IoT) is a decentralized and ever-changing network, which poses challenges in terms of security. The input highlights the need for robust security measures to protect IoT devices and their data from potential threats. The study focuses on Federated Learning (FL) technology as a potential solution to enhance IoT security. FL models are designed to protect sensitive data while allowing its exchange with other systems, making it a promising approach for securing IoT environments. Additionally, the input suggests the implementation of intrusion detection systems (IDS) as an additional strategy to enhance overall IoT security. By combining FL and IDS, the aim is to develop a comprehensive solution to address the complex problem of protecting IoT settings. The input emphasizes the significance of exploring machine learning (ML) techniques to improve security protocols for IoT devices. It also highlights the importance of validating the effectiveness of FL technology in safeguarding and transferring confidential information within IoT systems. The integration of IDS is proposed as an extra measure to strengthen the security of IoT systems as a whole. Ultimately, the objective of this research is to provide comprehensive and effective solutions to address security challenges in the IoT, thereby increasing trust in the application of this technology across various domains.

Keywords: Internet of Things (IoT), Security Measures, ML, Federated Learning (FL), IDS

1. Introduction

In recent years, Federated Learning (FL) has experienced substantial advancements across multiple academic disciplines and enterprises in recent times. FL, an emerging methodology in ML training, is characterized as an on-device collaborative ML environment. It was introduced by Google in 2015 [1, 2]. After training the global model using the local device's data from each client, the trained local model and client data remain on the client edge device in FL. Clients notify a central server of their updated parameters after training a local model. The server then compiles these updates from all clients in the pool and employs them to modify the parameters of the global model. When the global model is revised following one round of FL training another cohort of clients is selected to participate in a second round of training utilising the updated global model. The modular nature of FL facilitates the establishment of a secure environment for collaborative, private machine learning [3]. To ensure the security of both the trained global model and the client data, a few secure computations can be incorporated during the fusion of the client's local parameter updates.

There exist three primary methodologies for educating machine learning. The first is the traditional approach based on a central server. The subsequent approach is the contemporary distributed method, which necessitates training process parallelization. Lastly, there is the less prevalent decentralised method, which typically operates under the

Received 1 November 2023; accepted 5 July 2024.
Available online 25 November 2024

* Corresponding author.
E-mail address: dr.mohammed@esraa.edu.iq (M. Q. Mohammed).

https://doi.org/10.52866/2788-7421.1218
2788-7421/© 2024 The Author(s). This is an open-access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

assumption that the datasets of all clients are independent and identically distributed (IID) [4, 5].

In addition, improvements in the personalisation of client models are achievable in FL's collaborative learning environment by utilising unique user data to update the global model during the local training process. To enhance user privacy, FL restricts communication to aggregated model updates that have effectively undergone numerous secure computations, as opposed to exposing specific user information. The majority of recent research on the expansion of FL has concentrated on technical aspects such as framework architectures and their practical applications. In addition to system concerns such as resource allocation, privacy and security, and communication expenses. [6].

The existing body of literature primarily examines technical aspects and obstacles related to FL. However, more recent studies have fallen short of providing a comprehensive analysis of FL's current state and prospective developments in terms of markets and applications [6]. Hence, the utilisation of ML or FL models enables the extraction of valuable insights or data trends from security data. Using this procedure, machines can be programmed to identify preliminary indications of potential hazards. Integration of applications with the Internet of Things (IoT) may be protected considerably more effectively by intrusion detection systems (IDS) that employ artificial intelligence (AI) technology. This is especially beneficial when contemplating the Internet of Things as a strategy to overcome the constraints imposed by [7].

2. Literature review

This section offers a comprehensive analysis of Federated Learning, with a particular focus on IDS. Federated Learning, often known as FedAvg, is a parallelism technique that was devised by [8]. The technique involves training statistical methods directly on devices. Because users are concerned about data privacy as well as the increasing processing capabilities of devices, Florida exemplifies the overarching principle of "using code to access data rather than data accessing code" and has made substantial progress, with several methodologies proposed to address its challenges.

In the study [9] presented FedProx as a comprehensive and restructured adaptation of FedAvg, aiming to tackle the issue of data heterogeneity in FL. FedProx improves stability and precision in networks with different characteristics by introducing a slight adjustment to FedAvg. This improvement involves including a regulation term in every local impartial function.

The success of federated learning (FL) has resulted in its implementation in intrusion detection systems, where many studies have suggested FL-based approaches for identifying abnormal activities in Internet of Things (IoT) networks. In [10] presented DÏoT, a distributed system that utilizes federated learning (FL) to effectively combine behaviour profiles in a self-learning manner. The solution, comprising a Security Gateway as well as an IoT Security Service (SG and SS), exhibited exceptional precision in identifying abnormalities in actual IoT devices infected with malicious software.

[11] introduced an intrusion detection system for IoT devices that use federated learning. The approach focuses on promoting knowledge exchange among peers while maintaining anonymity. Their assessment of the NSL-KDD dataset demonstrated that FL surpassed on-device learning and achieved comparable performance to centralized learning.

At [12] we presented an approach to anomaly detection in IoT systems by asynchronous federated learning. By addressing the gradient delay problem and utilising a denoising autoencoder model outperformed previous techniques in terms of (Acc%, recall, precision, as well as F1-score), to achieve enhanced convergence.

Although FL offers advantages in the field of intrusion detection, its vulnerability to hostile attack was assessed at [13]. Upon examining FL's application in malware detection for IoT devices, the researchers discovered that it suffered a reduction in precision when maliciously attacked. This underscores the necessity of employing more resilient approaches to address this issue.

Furthermore, Numerous studies have investigated the issues surrounding network intrusion detection systems that utilise FL. To address the problem of limited data, MTDNN-FL employs a multi-task learning approach to tackle multiple FL framework tasks concurrently [14]. A recent study showcased the impressive ability of FL to effectively and reliably detect abnormalities in network intrusion detection systems. The study focused on applying FL to the task of identifying unusual occurrences within these systems [15]. Scientists have developed an intrusion detection system (IDS) for networks that utilises FL to analyse large volumes of data and identify potentially suspicious behaviour [16]. The team introduced a fresh federated learning (FL) architecture that prioritises privacy. This approach demonstrated superior accuracy and convergence rate compared to previous FL-based intrusion detection systems (IDS) [17], across multiple datasets.

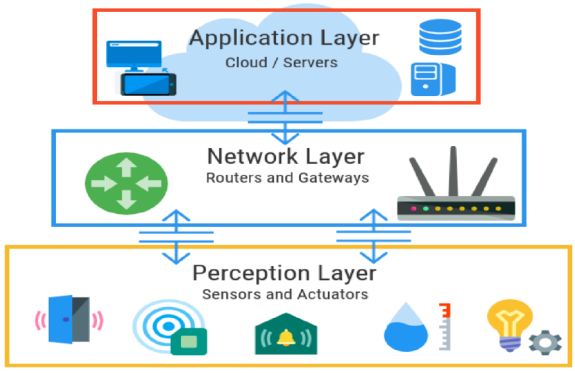


Fig. 1. Internet of Things architecture consists of three layers.

2.1. Internet of Things (IoT) concept

The three elements of the IoT architecture are depicted in Fig. 1: the Perception Layer, the Network Layer, and the Application Layer. The Perception Layer is comprised of physical devices that interact with or acquire data from the environment. Communication and data transmission between devices in the Perception Layer and higher layers are facilitated by the Network Layer. Using the Application Layer, users or applications engage with the Internet of Things (IoT) system using an interface that processes and analyses data acquired from the Perception Layer. Together, these layers facilitate a wide range of Internet of Things applications and use cases [18]. The input further specifies that the physical layer comprises real-time data collection sensors and actuators, the network layer employs networking protocols to establish encrypted communication between devices, and the application layer utilises machine learning algorithms to deliver tailored services while safeguarding the integrity and reliability of the IoT network.

Through device gateways, the Internet of Things architecture connects sensors and actuators to the application. The application subsequently uses a rule engine to process the data from these devices. Any apparatus capable of wireless or wired communication with sensors is referred to as a device. Gateways serve to enable communication between systems and devices, even when it is not possible to establish direct connections. A gateway functions as a conduit that facilitates the transmission and analysis of data between various devices and components. One can effortlessly generate straightforward processing rules in the IoT by utilizing the rule engine, eliminating the need for scripting. The user possesses the capability to establish foundational principles that govern the actions of the system in response to particular situations.

2.2. Security assault on the internet of things

The incorporation of IoT into external environments facilitates intelligent and automated communication between devices and their surroundings. Generally, IoT devices exchange physical words to accomplish various duties. Nevertheless, a comprehensive examination of the characteristics of these devices as well as their actions in both virtual and physical settings is necessary to ensure their security [19, 20]. As previously mentioned, the development of a resilient security framework to detect various cyber-assaults in the Internet of Things is a formidable undertaking. The complexity of this issue may increase when considering the security of wireless networks. Because the majority of IoT devices operate in an autonomous, centralized, and open environment, eavesdropping on these devices to obtain sensitive and confidential data becomes a simple task. Furthermore, IoT devices are distinguished by their restricted computational capabilities and substantial resource usage, which exacerbates preexisting difficulties and increases the likelihood of potential hazards [21]. A threat can be described as an action that takes advantage of vulnerabilities in a system's security to cause damage to it. In essence, hazards can be classified as either active or passive [22]. Denial of service (DoS) assault, Sybil assault, malware analysis, device deception, and man-in-the-middle assault are all examples of active threats. In contrast, passive threats consist of phishing assaults, surveillance, and so forth. This assault significantly compromises the reliability and effectiveness of the Internet of Things system.

The following are the security characteristics that are taken into account when developing a prospective IoT security framework:

1. Confidentiality is important in IoT systems to prevent unauthorized access to critical information [23].
2. Integrity of device information is crucial to identify transmission modifications and prevent malicious threats [24].
3. Authentication is necessary to establish the identity of users or devices before any operation [25].
4. Authorization schemes protect sensitive information by allowing only authorized individuals to access data [26].
5. Availability of data is important for authorized parties to retrieve their specific information resources [27].
6. Non-repudiation ensures the trustworthiness and reliability of exchanged data by providing

Fig. 2. Security and privacy issues on the Internet of Things.

evidence of its provenance, dependability, and integrity [28].

7. Key Management: Ensuring the secure creation, distribution, safekeeping, and cancellation of cryptographic keys [29].
8. Privacy: Refers to the safeguarding of personal information to ensure its confidentiality and integrity [30]. As shown in Fig. 2.

3. Federated learning

Federated learning, a method that ensures data confidentiality while enabling on-device training for decentralized systems, has been demonstrated to be an effective solution [31]. FL has become increasingly popular as a widely used approach to guarantee the security, accuracy, and fast transmission of data [32–36]. Federated learning shines by efficiently overcoming the constraints of centralized paradigms and surpassing typical machine learning methods in terms of preserving data privacy while sharing knowledge with other systems. Federated learning models utilize a remarkable approach that allows them to disseminate a learned machine-learning model among numerous devices. The taught machine learning model uses the computational resources at its disposal to help these gadgets gain knowledge about their environment. Federated learning offers several advantages due to its special characteristics and operational principles:

1. FL enhances privacy by not requiring the use of unprocessed data, making it suitable for IoT security.
2. FL reduces communication latency and resource utilization by eliminating the need to transmit IoT data to a server.
3. FL improves convergence rate and learning quality compared to traditional machine learning techniques.
4. FL is widely implemented in various IoT applications but lacks specialized research on its implementation in IoT security.
5. The study aims to emphasize the implementation of federated learning for IoT security and provides a taxonomy of federated learning models for IoT network security.

3.1. Federated Learning Internet of Things

The implementation of FL in Internet of Things (IoT) environments with limited resources presents a multitude of obstacles. To resolve these concerns, [37] proposed CoLearn, an architecture built upon the open-source Manufacturer Usage Description (MUD) implementation (osMUD) and the FL framework PySyft. The General Gradient Sparification (GGS) framework was expressly designed for FL in Edge Computing environments by the authors of reference [38]. In addition, experiments were conducted on LeNet-5, CifarNet, DenseNet-121, and AlexNet utilizing adaptive optimizers. A PerFit framework for personalized FL was introduced by the authors of the article [39]. This framework effectively mitigates the heterogeneity that exists among IoT applications' devices, statistical data, and models. In addition, a case study of Internet of Things-based human activity recognition was provided to demonstrate the effectiveness of personalized FL for intelligent IoT applications. The current focus of the majority of research in the domain of IoT pertains to the integration of additional sensors for data collection. A framework for the development of smartphone-integrated sensors that aid in decision-making across diverse domains was proposed by the authors of the reference [40]. As per its underlying concept, every device within this framework operates as a decentralized decision-making structure.

The authors proposed a method in [41] for integrating FL-based distributed Deep Learning into networks of Internet of Underwater Things (IoT) devices by employing a concise iterations algorithm based on MADDOG. When compared to reinforcement learning that was executed utilizing JCARA methods, the MADDOG-based algorithm demonstrated superior efficacy. By differential, A real-time data processing architecture for multi-robot environments was proposed by FL and its authors [42]. The proposed architecture obtains data for designated objectives

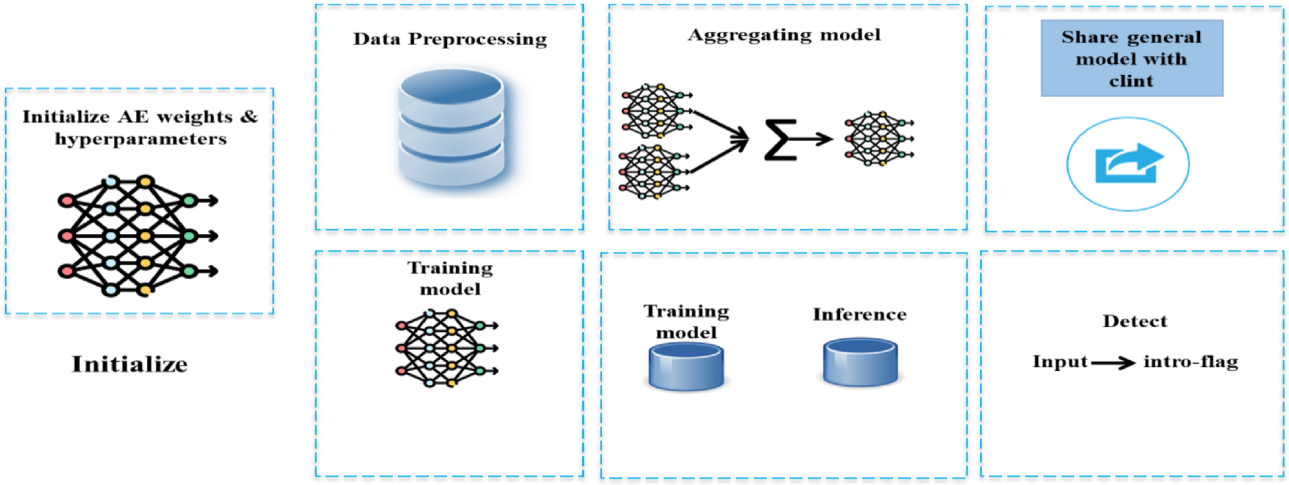


Fig. 3. The architecture of FL for anomaly-based-IDS consists of four primary parts, initializing model, training model, aggregating model, and disseminating model.

while safeguarding data privacy and enabling real-time processing. FL faces obstacles with centralized optimization, which is predicated on a centralized server. This can impede the scalability of the network and lead to a single point of failure. The authors of the paper [43] devised a fully distributed FL algorithm that generates data functionalities throughout the network, thereby eliminating the necessity for a centralized server and singular points of failure.

The authors in [45] examined the problem of JPRA, or joint power and resource allocation, within the framework of vehicle networks with low-latency communication. A framework for combined transmit power and resource allocation based on FL was put in place to address the issue of delayed in-vehicle communication. As well as in [46], the authors go into the industrial IoT space to find out how to process and analyse data for computer environments that use deep learning to predict the health and life of parts. This is achieved through the utilization of a methodology based on RUL predictions. The authors referenced [47] a comprehensive analysis of opportunities and solutions for FL in vehicular networks. Moreover, the authors of [48] presented an overview of FL in the context of IoT. The six main areas of discussion in their work are as follows: FL fundamentals and developments, FL technical challenges and solutions in wireless IoT environments, FL benefits and technical issues in vehicular IoT, and FL future research. Further studies suggest that the implementation of FL in the training of IoT applications can improve the overall user experience [46].

4. Methodology

We present a method for network intrusion detection using anomaly-based techniques, specifically utilizing federated learning (FL) and autoencoders. Fig. 3 illustrates the comprehensive structure of Federated Learning (FL) for detecting network intrusions based on anomalies. The structure comprises four primary elements:

1. Initializing model,
2. Training model,
3. Aggregating model, and
4. Disseminating model.

In the context of the paper, various notations are employed to describe key aspects and parameters throughout the remaining sections. As described in Table 1, serves as a reference guide for these symbols and abbreviations.

4.1. Initializing model

We assume that we are a decentralized learning system component. Before the learning process begins, specific hyperparameters, including learning rates, momentum, and the value of **O** (for FedProx) [9], are initialized by the server of central. These constitute the general model's weights. The general model may be implemented using an Autoencoder (AE), Variational Autoencoder (VAE), or Adversarial Autoencoder (AAE). The server is additionally tasked with ascertaining the initial distribution (a) in the AAE model. The distributes of server the model as

Table 1. Describe the abbreviation.

Abbreviation	Describe	Detailed explanation
K	The entire network's clients and entities	Represents all clients and entities within the network.
E	Count of all comm. rounds	Indicates the total number of communication rounds.
I	All the local iterations added together	Denotes the cumulative count of all local iterations.
N	Size for mini-batch	Refers to the number of data points in a mini-batch.
IS	Score of Intrusion	Represents the score assigned to intrusion detection.
thr	Threshold for detecting anomalies	The threshold value is used to identify anomalies.
δ	The FedProx algorithm's penalty term	Penalty term utilized in the FedProx algorithm.
DK	Customer k's local dataset	The dataset is specific to customer k within the network.
α	The rate at which Decoder Learns	The rate at which the decoder learns during training.
β	Decoder Learns	Another rate at which the decoder learns during training.
γ	The Discriminator's Learning Rate	The rate at which the discriminator learns during training.
$p(z)$	The prior distribution of AAE	The prior distribution is utilized in the Adversarial Autoencoder (AAE).
θt, ϕt, and χt,	Three parameters are defined for the encoder, decoder, and discriminator at round t:	Parameters are specific to the encoder, decoder, and discriminator at round t.
$\mathscr{B}$	Sizing of the local mini-batch	Indicates the size of the mini-batch used locally.

well as hyperparameters to a client (C) that was chosen in an initial round, once initialization is complete.

4.2. Training model

To develop a comprehensive model for detecting network intrusions, we analyze a group of K clients working together in collaboration. The clients serve two primary purposes, namely data preparation as well as local training. Table 2 describes these two presses.

The data transformation pipeline, depicted in Fig. 4, is an essential component in the preparation of unprocessed data for modelling and analysis. The pipeline comprises several stages, namely data transformation, feature extraction or selection, data normalisation or scaling, and data cleansing. Every phase is designed to enhance the data's integrity and render it appropriate for applications such as machine learning or data analysis. The figure probably illustrates flowcharts or visual components that depict the data transformations performed at each stage.

In contrast, a federated algorithm for an anomaly intrusion detection system is illustrated in Fig. 5. The purpose of this algorithm is to identify malicious activities or unauthorised access in computer networks. The methodology employs a distributed structure in which the detection model is collectively trained on numerous clients or devices, eliminating the need for centralised aggregation of sensitive data. By utilising

Table 2. Describes data preparation and local training presses.

Data preparation	Local training
Each client prepares and ensures the quality of their data before feeding it to the autoencoder-based model.	From the server, the client retrieves the autoencoder's current general state.
The publicly available PCAP3 files are used for each dataset.	For AE to learn how to encode and reconstruct typical behaviour, it is trained using solely typical trials of the client's data.
The PCAP files are repaired using the pcapfix4 tool to fix any corruption or damage.	MSE is mini among the input & its reconstruction.
The repaired files are then sorted by timestamp using the reordercap program.	Isotropic Gaussian distribution is used as the prior for training the discriminator of the AAE.
This stage holds significance for files that were generated through the fusion of frames from various sources, disregarding the order of time.	When updating local models, FL anomaly IDS usage the FedProx [9].
The resultant files are corrected and arranged PCAPs that are operational.	In order to update local tech., a regularization is incorporated into each client's loss function.
Flow construction, labelling, and attack simulation are all insufficient in current NIDS datasets.	
To convert PCAP files into CSV files, the suggested tool extracts 87 statistical flow features.	
To correctly name flows in CIC-IDS2017 and CSE-CICIDS2018,	
Numeric and categorical features are both extracted, with character data being transformed into numerical values by feature encoding.	
min-max scaling is executed.	

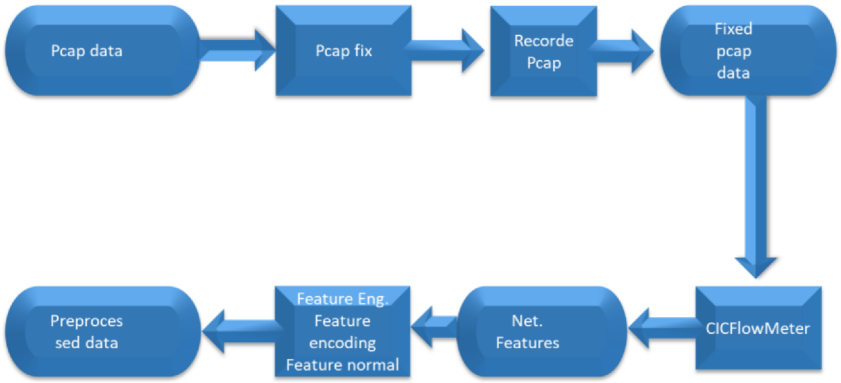


Fig. 4. Data preprocessing pipeline.

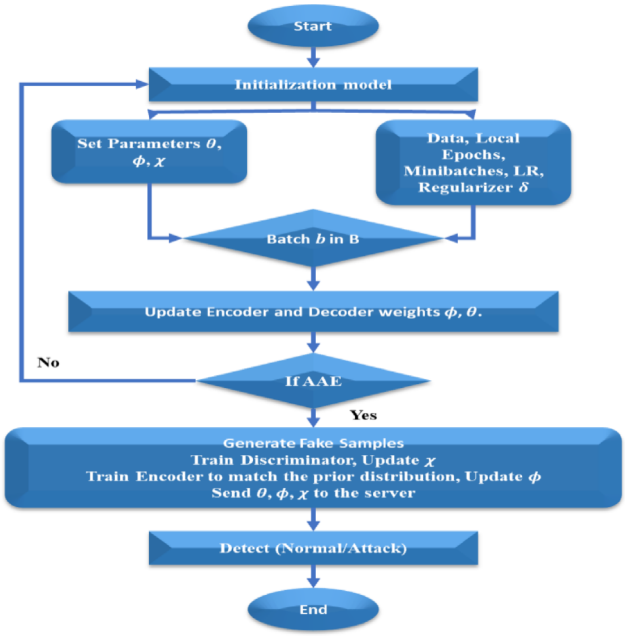


Fig. 5. Federated algorithm for anomaly intrusion detection system.

federated learning methodologies, the algorithm enables every client to train a local detection model with its own set of data, while also facilitating the periodic exchange of model updates with a central server. These updates are compiled by the central server to enhance the detection model as a whole, all the while protecting privacy and security. The depicted figure probably illustrates the activities of the clients participating in the federated algorithm, including but not limited to local model training, parameter updates, communication with the central server, and collaborative learning with other clients.

4.3. Aggregating model

An essential part of FL is model aggregation. It safeguards sensitive data while allowing for the merging of local models learned on different devices into one model. Therefore:

1. Local models trained on separate devices are submitted to the server after each communication round.
2. The server calculates new general weight parameters by considering all modifications and applying a weighted average.
3. Convergence is achieved by repeating this process.
4. The general model is updated by taking the weighted average.

4.4. Disseminating model (Threshold selection)

After the completion of the training procedure, it is necessary to calculate a score threshold for the IDS phase. To achieve this objective, we establish the distinct validation set (Val-set) on which we ascertain an appropriate *thr* for the whole model. It is important to note that the performance of the model is highly dependent on the threshold value. Increasing the threshold would reduce the occurrence of false alarms, but it may also classify more assaults as regular occurrences. Conversely, a lower value would result in an increased number of false alarms and classify more regular occurrences as assaults. We suggest calculating the threshold *thr* using the Eq. (1). In other words, the threshold is determined by adding the total as well as the standard deviation *MSE* over the Val-set, as described by [9].

$$thr = (MSE(Dval, \phi t)) + s(MSE(Dval, \phi t)) \quad (1)$$

Subsequently, the established threshold is employed during the inference phase. We evaluate *IS*, which represents the loss of reconstruction, by comparing it to a pre-calculated threshold. An occurrence is classified as an intrusion if its intrusion score

exceeds the threshold; otherwise, it is classified as benign.

5. Dissemination of model parameters

Upon completion of the training process and model convergence, the most recent iteration of the general model is distributed to all consumers inside the network. Cap*IS* is calculated upon the addition of a new instance *Xnew* to the network through the computation of the reconstruction loss. If the IS (Intrusion Score) of instance *Xnew* exceeds a specified threshold value, the instance is classified as an intrusion; otherwise, it is classified as a typical instance.

$$Xnew = \begin{matrix} Anomaly, & if\ IS > thr \\ Normal, & otherwise \end{matrix} \quad (2)$$

6. Experimental results

We assess the effectiveness of the suggested Fed-anomaly-IDS approach by analyzing its performance on several widely recognized datasets, namely USTC-TFC2016, CICIDS2017, and CSE-CIC-IDS2018. Our primary emphasis is on addressing the following three aspects:

First: What is the performance of intrusion detection systems and federated learning-based anomaly detection compared to other baselines, such as (GAN and BiGAN), in terms of different intrusion detection metrics?

Second: Which distributed algorithm, FedProx [9] or FedAvg, is the most efficient?

Third: How does the Fed-anomaly-IDS perform when trained on data from different contexts, and how does it perform when tested on new, unseen data?

6.1. Datasets

The provided input outlines three reputable datasets that are frequently employed in assessments of intrusion detection systems and federated learning-based anomaly detection performance. The primary dataset utilized in this study is USTC-TFC2016. It consists of malware traffic acquired from publicly accessible websites as well as regular traffic gathered via IXIA BPS. A subset of the dataset is designated for testing, validation, and training purposes. The second dataset is CIC-IDS2017, which contains classified “Safe” and “Malicious” genuine network traffic data. The dataset is partitioned based on the days of the week into (training, Val., as well as test sets). CSECICIDS-2018, the third dataset, is an enhanced version of CICIDS-2017. It comprises seven categories of contemporary attacks. By dividing the dataset into training, validation, and test sets over two weeks, the dataset is partitioned. In this investigation, the training and validation sets solely comprise the standard samples, whereas the test set comprises the assault samples exclusively. For training purposes, the training set is distributed equitably and arbitrarily among all customers in the system to guarantee an even distribution of typical data. The dataset distribution of safe and malicious for each type is illustrated in Fig. 6.

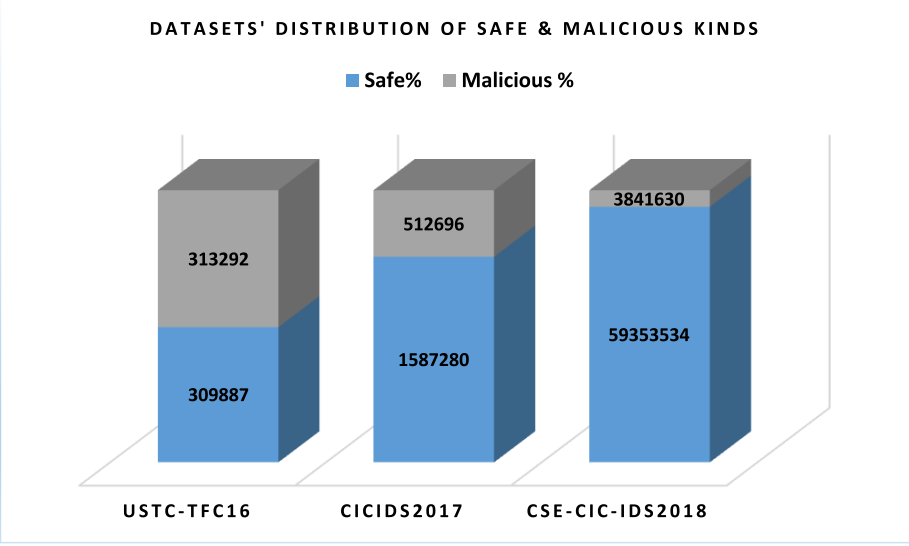


Fig. 6. Datasets distribution of safe & malicious kinds.

Table 3. Experimental Settings and Architectural Configurations for IDS using Federated Learning and Autoencoder Variants.

Setting	Value/Description
Number of Clients (K)	10
Client Fraction (C)	0.5
Local Iterations (I)	10
General Comm Rounds (E)	The first and second types of datasets = 30 epochs, and the third one = 10 epochs
Regularization Parameter (FedProx)	Candidate set: {0.1, 0.01, 0.001, 0.0001, 0.00001, 0.000001}
Learning Rate (AE & VAE)	Range: {0.01, 0.001, 0.0001, 0.00001}
Weight Decay (AE & VAE)	Range: {0.0001, 0.00001, 0.000001}
AE Architecture	Encoder: (87-64-32), Decoder: (32-64-87), ReLU activation function
VAE Architecture	Encoder: (87-64-64-32), Decoder: (32-64-64-87)
AAE Architecture	Encoder: (87-16-4-2), Decoder: (2-4-16-87), Discriminator: (16-4-2), LeakyReLU activation
Evaluation Metrics	F1-score, Acc, and FDR
Programming Languages & Libraries	Python, PyTorch, Numpy, Pandas

6.2. Settings of experimental

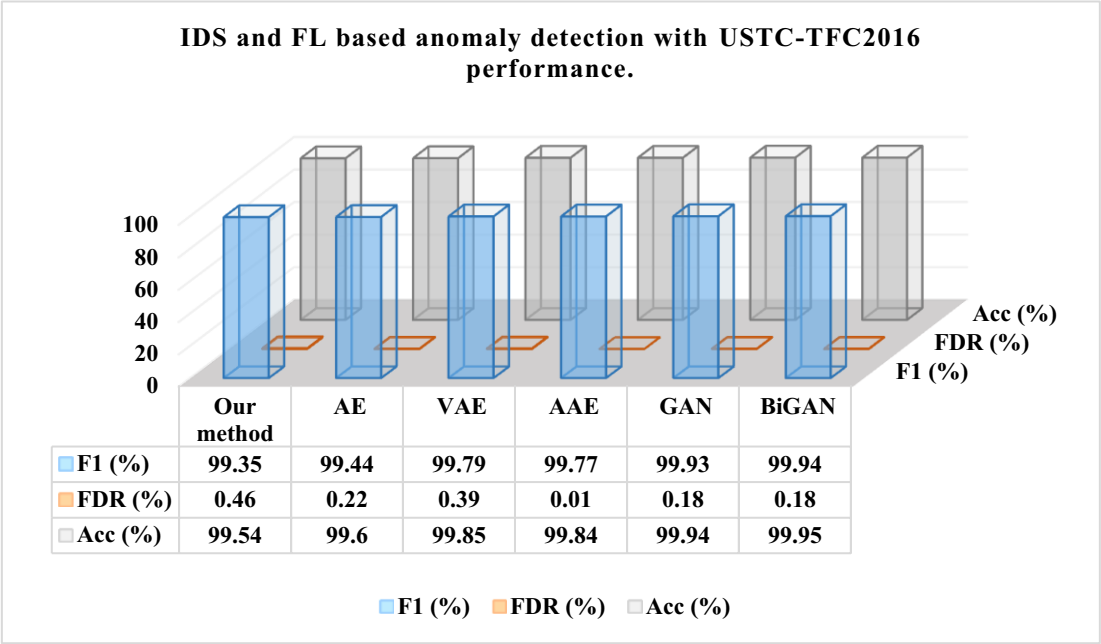
The experimental settings, in Table 3, outline key parameters and configurations used in the study. It involves details such as the number of clients (K), client fraction (C), local iterations (I), and general communication rounds (E) for different datasets. Hyperparameter choices for regularization, learning rates, and weight decay are specified. The Autoencoder (AE), Variational Autoencoder (VAE), and Adversarial Autoencoder (AAE) are broken down in terms of their architecture, showing how many neurons are in each layer and how they are activated. The evaluation metrics include F1-score, ACC, and FDR.

6.3. Evaluation of performance evaluation

To ensure the credibility of our evaluations, we conduct a comparative analysis of IDS and FL-based anomaly detection with other studies in the same domain. An intrusion detection system utilizing anomaly detection (AD) and fuzzy logic (FL) was proposed in reference [47]. The recommended approach relies on Geometric ANNs, which are extensively utilized in various sectors like computer vision as well as anomaly detection [48]. We followed our recommended protocol and employed the USTC-TFC2016 dataset, the CICIDS2017 dataset, as well as CSECICIDS-2018 dataset to evaluate the effectiveness of fundamental models GAN and BiGAN [49]. In tasks such as anomaly detection and unsupervised learning, the latter can obtain complex and comprehensive representations [50]. Autoencoders and GANs are examples of unsupervised learning methodologies that exhibit potential in the detection of network intrusions. To replace the FedAvg FL technique utilized in [47], we employ the FedProx federated learning technique. Furthermore, we employ refined iterations of flow characteristics derived from datasets for all assessments. Conversely, [47] employed images derived from datasets such as NSL-KDD, 10 KDD [51], and UNSW-NB15 [52] for their research. A FL network highly values the format of the dataset. Raw traffic image datasets are characterized by their higher size, more complexity in interpretation, and greater storage requirements compared to tabular datasets that encompass several parameters extracted from raw traffic data. We choose datasets that rely on flow features since edge entities in FL systems usually possess restricted resources.

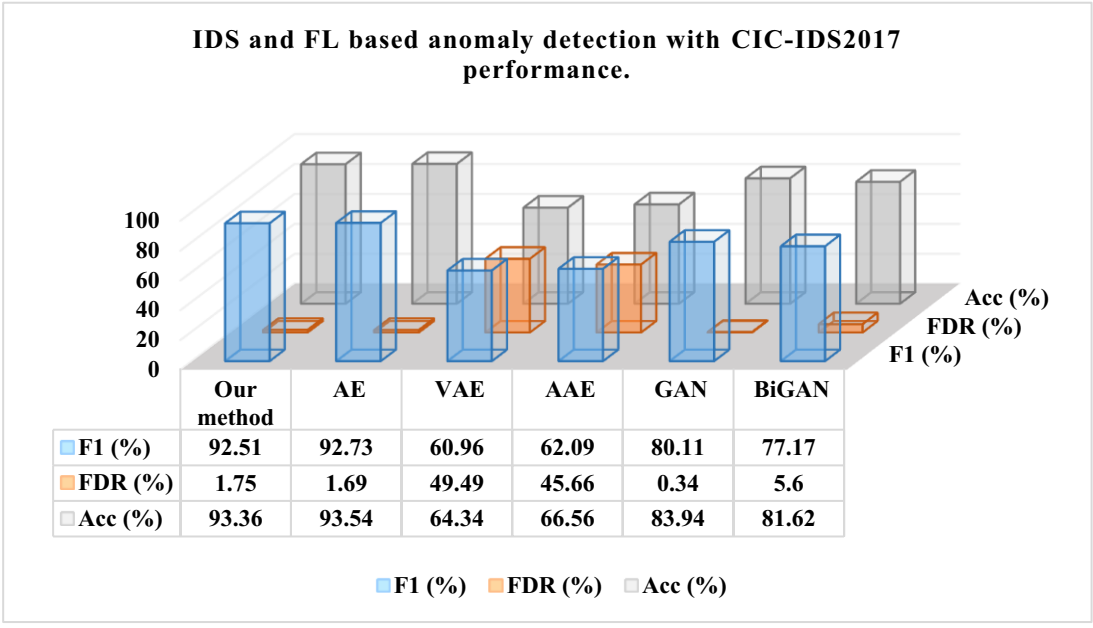
Fig. 7 presents a concise overview of the outcomes obtained from our assessment of the model using the USTC-TFC2016 dataset. AAE, when combined with the FedProx algorithm, exhibits exceptional performance, attaining a remarkable F1-score and Acc of 99 per cent approximately, as well as a negligible false discovery rate value of 18 per cent. The performance of learning rate and weight decline is likewise impressive. On the other hand, GAN and BiGAN demonstrate subpar outcomes, with BiGAN, in particular, displaying elevated (FDR, F1-score, and Acc) Furthermore, FedProx consistently achieves equal or superior performance compared to FedAvg across all models.

Transitioning to Fig. 8, directing attention to the assessment outcomes utilizing CIC-IDS2017, a basic Autoencoder (AE) trained with Federated Proximal (FedProx) [9] emerges as the highest-performing model. The accuracy achieved is greater than 93 percent and the false discovery rate (FDR) is at its lowest, measuring 1.693 percent. Finally, in Fig. 9, the evaluation is conducted utilizing the type of dataset (CSE-CIC-IDS20180 and it is observed that VAE demonstrates superior performance in terms of (F1-score as well as Acc). The basic AE model achieves an optimal false discovery rate (FDR) at 0.6 per cent, however, there is a minor decrease in both the F1 score as well as Acc. In general, the straightforward AE model proves to be the most efficient across all datasets.



	Our method	AE	VAE	AAE	GAN	BiGAN
F1 (%)	99.35	99.44	99.79	99.77	99.93	99.94
FDR (%)	0.46	0.22	0.39	0.01	0.18	0.18
Acc (%)	99.54	99.6	99.85	99.84	99.94	99.95

Fig. 7. IDS and FL-based anomaly detection with USTC-TFC2016 performance of evaluation.

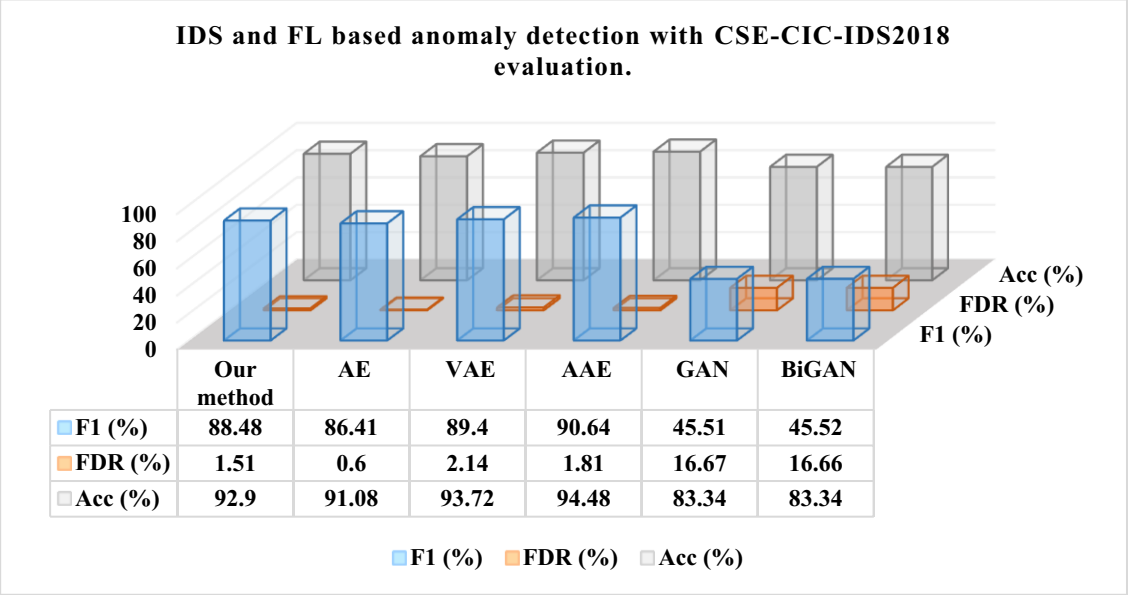


	Our method	AE	VAE	AAE	GAN	BiGAN
F1 (%)	92.51	92.73	60.96	62.09	80.11	77.17
FDR (%)	1.75	1.69	49.49	45.66	0.34	5.6
Acc (%)	93.36	93.54	64.34	66.56	83.94	81.62

Fig. 8. IDS and FL-based anomaly detection with CIC-IDS2017 performance of evaluation.

The performance of at least one AE-based technique utilizing FedProx is consistently superior to that of GAN depending on models across all datasets. Particularly in the domain of NIDS, our findings indicate that autoencoders are more effective than GAN-based methods at detecting potential hazards. Hence, AE is deemed more suitable for practical network intrusion detection situations due to its straightforwardness, low weight, and computational effectiveness.

FedAvg and FedProx denote the techniques of Federated Averaging and Federated Proximal, respectively. AE is for Autoencoder, VAE stands for Variational Autoencoder, AAE stands for Adversarial Autoencoder, GAN stands for Generative Adversarial



	Our method	AE	VAE	AAE	GAN	BiGAN
F1 (%)	88.48	86.41	89.4	90.64	45.51	45.52
FDR (%)	1.51	0.6	2.14	1.81	16.67	16.66
Acc (%)	92.9	91.08	93.72	94.48	83.34	83.34

Fig. 9. IDS and FL-based anomaly detection with CSE-CIC-IDS2018 performance of evaluation.

Network, and BiGAN stands for Bidirectional Generative Adversarial Network. The F1-score quantifies the trade-off between precision and recall, the FDR measures the rate of false discoveries, and Accuracy reflects the overall correctness of the model.

7. Conclusion

In the last part of this article, the main findings and contributions of the research are talked about, with a focus on how federated learning can be used to create a system for finding suspicious activity in distributed networks. The study presents a model that incorporates a variety of cutting-edge technologies and methods, such as hardware auto-encoders, federated learning, and anomaly detection. The objective of the model is to construct cognitive instruments that are capable of correct data manipulation and to guarantee the security of training processes. The research methodology being proposed is founded upon the Fed-Prox algorithm and utilises three distinct dataset categories for autoencoders. The outcomes of tests and evaluations indicate that the proposed approach effectively safeguards privacy and minimises false alarms while ensuring effective detection of network intrusions. The study also shows that federated learning anomaly identifiers work better than other frameworks that use generative adversarial networks (GANs). Additionally, the Fed-Prox algorithm consistently performs better than the Fed-Avg algorithm in federated learning anomaly intrusion detection systems. The findings of the research validate the efficacy of autoencoders in detecting extensive intrusions in distributed systems. They also suggest ways to improve future research and development in the areas of federated learning and attack detection, as well as ways to make intrusion detection more accurate and useful across a wide range of network areas. In the future, work in the field of federated learning and attack detection systems in distributed networks can focus on developing advanced models for federated learning. This will include studying the impact of different data sets, applying the research to real-life scenarios, improving data security, and developing advanced intrusion detection systems. These areas of research aim to enhance the efficiency, accuracy, and effectiveness of attack detection systems in distributed networks. By exploring these avenues, further improvements and innovations can be made in the fields of federated learning and attack detection.

Funding

None.

Acknowledgement

None.

Conflicts of interest

None.

References

1. U. M. Aïvodji, S. Gambs, and A. Martin, "IOTFLA: A secured and privacy-preserving smart home architecture implementing federated learning." In 2019 IEEE security and privacy workshops (SPW) pp. 175–180, 2019.
2. A. Al-Laith, M. Shahbaz, H. F. Alaskar, and A. Rehmat, "Arasencorpus: A semi-supervised approach for sentiment annotation of a large Arabic text corpus." *Applied Sciences*, vol. 11, no. 5, p. 2434, 2021.
3. S. Alawadi, V. R. Kebande, Y. Dong, J. Bugeja, J. A. Persson, and C. M. Olsson, "A federated interactive learning IoT-based health monitoring platform," *In European conference on advances in databases and information systems*. Springer, pp. 235–246, 2021.
4. M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated learning: A survey on enabling technologies, protocols, and applications." *IEEE Access*, vol. 8, pp. 140699–140725, 2020a.
5. M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated learning: A survey on enabling technologies, protocols, and applications." *IEEE Access*, vol. 8, pp. 140699–140725, 2020b. doi: 10.1109/ACCESS.2020.3013541.
6. S. Banabilah, *et al.* "Robust federated learning with noisy communication." *IEEE Transactions on Communications*, 2020.
7. A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, and A. Barbado, *et al.* "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities, and challenges toward responsible AI." *Information Fusion*, vol. 58, pp. 82–115, 2020.
8. N. T. Nguyen, T. V. Dinh, H. T. Nguyen, D. T. Nguyen, and Q. V. Pham, "Distributed intrusion detection system for IoT using federated learning," in 2019 IEEE Asia-Pacific Conference on Advanced System Integrated Circuits (APASIC).
9. Y. Wu, Y. Kang, J. Luo, Y. He, and Q. Yang, Fedcg: Leverage conditional gan for protecting privacy and maintaining competitive performance in federated learning. arXiv preprint arXiv:2111.08211. 2021.
10. H. Tian, H. Chen, L. Yu, and X. Liao, "DC-Adam: A Decentralized and Asynchronous Federated Learning Approach for Anomaly Detection in IoT," in IEEE Transactions on Industrial Informatics, 2021.
11. J. A. S. S. Rey, A. Huertas Celdrán, and S. Bovet, "Adversarial Threat to Federated Learning in IoT Intrusion Detection," in 2022 International Conference on Cyber-Physical Systems (ICCPS), 2022.
12. Y. Zhao, X. Chen, Y. Wu, X. Teng, and S. Yu, "Multi-Task Learning for Federated Learning Based Network Intrusion Detection Systems," in Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (SAC), 2019.
13. B. Ayed and C. Talhi, "Federated Learning for Anomaly Detection in Network Intrusion Detection Systems: A Comprehensive Review," in *Computers, Materials & Continua*, vol. 68, no. 3, pp. 3737–3760, 2021.
14. Z. Qin and D. Kondo, "Federated Learning for Large-Scale Network Anomaly Detection," in 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC), 2021.
15. S. Tabassum, G. H. Nguyen, Q. V. Pham, and C. S. Hong, "FEDGAN-IDS: A Federated Learning Framework for Privacy-Preserving Intrusion Detection System," in *Future Generation Computer Systems*, vol. 128, pp. 642–652, 2022.
16. M. Litoussi, N. Kannouf, K. El Makkaoui, A. Ezzati, and M. Fartitchou, "IoT security: Challenges and countermeasures." *Procedia Comput. Sci.*, vol. 177, pp. 503–508, 2020.
17. D. Ratasich, F. Khalid, F. Geissler, R. Grosu, M. Shafique, and E. Bartocci, A roadmap toward the resilient Internet of Things for cyber-physical systems IEEE Access, vol. 7, pp. 13260–13283, 2019.
18. M. Krishna, S. M. B. Chowdary, P. Nancy, and V. Arulkumar, A survey on multimedia analytics in security systems of cyber-physical systems and IoT 2021 2nd International Conference on Smart Electronics and Communication, ICOSEC, IEEE, pp. 1–7, 2021.
19. K. Tabassum, A. Ibrahim, and S. A. El Rahman, Security issues and challenges in IoT 2019 International Conference on Computer and Information Sciences, ICCIS, IEEE, pp. 1–5, 2019.
20. S. Bhatt and P. R. Ragiri, *et al.* "Security trends in Internet of Things: A survey," *SN Appl. Sci.*, vol. 3 no. 1, pp. 1–14, 2021.
21. J. Zhang, H. Chen, L. Gong, J. Cao, and Z. Gu, "The current research of IoT security" 2019 IEEE Fourth International Conference on Data Science in Cyberspace, DSC, IEEE, pp. 346–353, 2019.
22. M. Gowtham and H. Pramod, "Semantic query-featured ensemble learning model for SQL-injection attack detection in IoT-ecosystems," *IEEE Trans. Reliab.*, 2021.
23. J. Zhang, C. Shen, H. Su, M. T. Arafin, and G. Qu, "Voltage over-scaling-based lightweight authentication for IoT security." *IEEE Trans. Comput.*, vol. 71 no. 2, pp. 323–336, 2021.
24. S. Ravidas, P. Karkhanis, Y. Dajsuren, and N. Zannone, An authorization framework for cooperative intelligent transport systems International Workshop on Emerging Technologies for Authorization and Authentication, Springer, pp. 16–34, 2019.
25. A. Y. F. Alsahlani and A. Popa, "LMAAS-IoT: Lightweight multi-factor authentication and authorization scheme for real-time data access in IoT cloud-based environment" *J. Netw. Comput. Appl.*, vol. 192, Article 103177, 2021.
26. P. M. Chanal and M. S. Kakkasageri, "Security and privacy in IOT: A survey Wirel." *Pers. Commun.*, vol. 115, no. 2, pp. 1667–1693, 2020.
27. F. Chen, J. Wang, J. Li, Y. Xu, C. Zhang, and T. Xiang, "TrustBuilder: A non-repudiation scheme for IoT cloud applications," *Comput. Secure*, vol. 116, Article 102664, 2022.
28. F. Hussain, S. G. Abbas, M. Husnain, U. U. Fayyaz, F. Shahzad, and G. A. Shah, IoT DoS and DDoS attack detection using ResNet 2020 IEEE 23rd International Multitopic Conference, INMIC, IEEE, pp. 1–6, 2020.
29. P. Ahlawat and R. Bathla, "A survey on key management solutions for IoT security," In 2023 4th International Conference on Computing and Communication Systems (I3CS), pp. 1–6. IEEE, 2023.
30. R. Singh, A. D. Dwivedi, G. Srivastava, P. Chatterjee, and J. C. W. Lin, "A privacy-preserving Internet of Things smart healthcare financial system." *IEEE Internet of Things Journal*, 2023.
31. Q. Yang, *et al.* "Federated Learning with Privacy-preserving and Model IP-right-protection." *Machine Intelligence Research*, vol. 20, no. 1, pp. 19–37, 2023.
32. Y. Zeng, *et al.* "Adaptive federated learning with non-IID data." *The Computer Journal*, vol. 66, no. 11, pp. 2758–2772, 2023.
33. Y. M. Lin, Y. Gao, M. G. Gong, S. J. Zhang, Y. Q. Zhang, and Z. Y. Li, "Federated Learning on Multimodal Data: A Comprehensive Survey," *Machine Intelligence Research*, pp. 1–15, 2023.
34. A. Alsajri and A. Steiti, "Intrusion Detection System Based on Machine Learning Algorithms:(SVM and Genetic Algorithm)."

Babylonian Journal of Machine Learning, pp. 15–29, 2024. doi: https://doi.org/10.58496/BJML/2024/002.
35. J. F. Yonan and N. A. A. Zahra, "Node Intrusion Tendency Recognition Using Network Level Features Based Deep Learning Approach." *Babylonian Journal of Networking*, pp. 1–10, 2023. doi: https://doi.org/10.58496/BJN/2023/001.
36. Fernando *et al.*, "CoLearn: An Architecture for Federated Learning in Resource-Constrained Internet of Things Environments," 2020.
37. Sun *et al.*, "General Gradient Sparification (GGS) Framework for Federated Learning in Edge Computing Environments," 2020.
38. He Wu and Chen, "PerFit: Personalized Federated Learning for Addressing Heterogeneity in IoT Applications," 2020.
39. Imteaj and Amini, "Smartphones as Decision-Making Devices: A Framework for IoT Integration," 2019.
40. G. Amirthayogam, N. Kumaran, S. Gopalakrishnan, K. Brito, S. RaviChand, and S. B. Choubey, "Integrating Behavioral Analytics and Intrusion Detection Systems to Protect Critical Infrastructure and Smart Cities," *Babylonian Journal of Networking*, pp. 88–97, 2024. doi: 10.58496/BJN/2024/010.
41. S. H. Ahmed, "An Analytical Study on Improving Target Tracking Techniques in Wireless Sensor Networks Using Deep Learning and Energy Efficiency Models." *EDRAAK*, pp. 8–11, 2024. doi: 10.70470/EDRAAK/2024/002.
42. A. S. Bin Shibghatullah, "Mitigating Developed Persistent Threats (APTs) through Machine Learning-Based Intrusion Detection Systems: A Comprehensive Analysis." *SHIFRA*, pp. 1–10, 2023. doi: 10.70470/SHIFRA/2023/003.
43. B. Samarakoon, Saad, Debbah, "Joint Power and Resource Allocation in Vehicular Networks using Federated Learning," 2020.
44. S. Hsu and Wu, Chen, "Deep Learning for Predictive Maintenance in Industrial IoT: A Remaining Useful Life Prediction-Based Approach," 2020.
45. T. Posner, Aloqaily, and Jararweh, "Opportunities and Solutions for Federated Learning in Vehicular Networks: A Comprehensive Analysis," 2021.
46. Du *et al.*, "Federated Learning in IoT: Basics, Advancements, Challenges, and Future Directions," 2020.
47. N. Tabassum, M. Tahir, S. Ahmed, and S. Y. Hwang, "A Privacy-Preserving Federated Learning-Based Intrusion Detection System with Adversarial Networks." *Sensors*, vol. 22, no. 4, p. 1199, 2022.
48. H. J. K. AL Masoodi, "Evaluating the Effectiveness of Machine Learning-Based Intrusion Detection in Multi-Cloud Environments." *Babylonian Journal of Internet of Things*, pp. 94–105, 2024. doi: 10.58496/BJIoT/2024/012.
49. J. Donahue, P. Krähenbühl, and T. Darrell, Adversarial feature learning. 2016. arXiv preprint arXiv:1605.09782.
50. K. Rajora, and abdulhussein N. salih, "Reviews research on applying machine learning techniques to reduce false positives for network intrusion detection systems." *Babylonian Journal of Machine Learning*, pp. 26–30, 2023. doi: 10.58496/BJML/2023/005.
51. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set." In 2009 IEEE symposium on computational intelligence for security and defence applications, pp. 1–6, 2009. IEEE.
52. M. Aljanabi, "Safeguarding Connected Health: Leveraging Trustworthy AI Techniques to Harden Intrusion Detection Systems Against Data Poisoning Threats in IoMT Environments." *Babylonian Journal of Internet of Things*, pp. 31–37, 2023. doi: 10.58496/BJIoT/2023/005.