Wireless Personal Communications
https://doi.org/10.1007/s11277-023-10852-z

RESEARCH

An Evolutionary Computation-Based Federated Learning for Host Intrusion Detection in Real-Time Traffic Analysis

A. Suresh[1] · B. Dwarakanath[2] · Ashok Kumar Nanda[3] · P. Santhosh Kumar[2] · S. Sankar[4] · Sreevardhan Cheerla[5]

Accepted: 26 December 2023

Abstract
Important information needs to be sent over the Internet safely. Real-time data is mixed with harmful information, lowering the quality of communication and the system's overall performance. A network intruder detection system is software that examines all incoming and outgoing network packets to detect malicious events. Federated learning (FL) is a way to put artificial intelligence at the cutting edge. It is a way to solve problems that is not centralized and lets people learn from significant amounts of data. Deep learning (DL) methods have often been used to find harmful data in host intrusion detection systems (HIDS) that look for unusual behavior. The FL architecture allows multiple users to train a global model while respecting the privacy of each user's data, making DL-based methods more useful. But there has yet to be a complete analysis of how well FL-based HIDSs protect against known privacy threats with the already in-place defenses. To solve this problem, we offer two privacy assessment measures for FL-based HIDSs, including a privacy score that rates how close the original and restored traffic attributes are. The CICIDS2017 dataset, which includes several attacks from the present day, was used to make the real-time model. In addition, an adaptive threshold-correlation algorithm (ATCA) is presented to enhance detection accuracy by dynamically adjusting threshold values according to traffic patterns and intrusion behaviors. The FL-HIDS framework was created and tested using a realistic network dataset. Experiment results show that the suggested technique outperforms existing intrusion detection systems regarding detection precision and scalability. The federated learning strategy effectively leverages the collective intelligence of network devices, enabling continuous learning and adaptation to emergent attack strategies. Furthermore, the adaptive threshold technique considerably reduces the rate of false positive and false negative detection, boosting the intrusion detection system's overall effectiveness. The proposed architecture solves previous centralized solutions' shortcomings by providing a scalable, privacy-preserving method for defending network environments against expanding invasion threats.

Keywords Evolutionary computation · Deep learning (DL) · Federated learning (FL) · Host intrusion detection · Real-time traffic analysis

Extended author information available on the last page of the article

Published online: 04 February 2024

1 Introduction

The Internet of Things (IoT) technology, which has applications in vital infrastructure, has revolutionized the creation of intelligent information systems. These systems improve the effectiveness of technology and business process management by automating tasks like equipment monitoring, lifecycle management, energy consumption control, and facilitating the best linkages between enterprises and service users [1]. Various industries, including smart grids, transportation systems, smart cities, and smart homes, are adopting IoT-based services, applications, and information systems. But along with its many advantages, the growth of IoT also raises new issues and worries about information security. Due to the diversity of connected devices with varying capabilities and communication protocols, the Internet of Things (IoT) introduces intricate security challenges, amplifying privacy concerns amidst the massive data influx. The shift to edge computing necessitates a reevaluation of security models. IoT devices present challenges for timely updates because of their extended lifespan and deployment in exposed environments, which necessitate heightened attention to physical security. A comprehensive cybersecurity strategy that includes secure device design, encryption, access controls, and stakeholder collaboration is required to address these complexities. It is essential to establish a robust security framework to effectively navigate the constantly changing IoT landscape. IoT devices, constrained by processing limitations and inadequate security, are highly vulnerable due to widespread interconnection. Ensuring IoT security requires prioritizing device-level protection, implementing encryption, and skillfully managing interconnected systems to minimize risks. The increasing number of networked gadgets raises many possible concerns that could endanger people's privacy and general security [2].

Considering the data in reference [3], it is possible to divide the security concerns connected to any IoT system into four major classes. These risk categories cover risks relating to device and user authentication, physical effects on system components, breaches of integrity, and availability, confidentiality, and risks relating to the handling of sensitive data, such as personally identifiable information. Personal data security and other sensitive information security are more critical than ever in the current environment [4]. This includes many categories of data, including biometric data, health records, and other kinds of information. Some components fall under the definition of personal data stated by the General Data Protection Regulation of the European Union [5]. Using these components to locate the people they belong to is possible. As a result, information security management systems frequently use these figures. Because of this, processing personal data under information security management obligations in IoT-based systems is more dangerous [6]. In addition, any unauthorized access to the information security management system puts at risk the privacy of user data and the security of essential information for IoT devices [7].

Cybercriminals use security system flaws to access confidential information, obstruct services, stop operations, steal data, and damage computer systems. In addition to having a negative impact on industrial production, these illicit acts continually endanger national security. The seriousness of such calamities was recently highlighted by a significant Distributed Denial of Service attack in Ukraine that caused substantial disruptions and raised public concern. In many data network contexts, accurate network attack detection can aid in locating security issues and enhancing security defense. Network attack detection is becoming a more widespread activity in considerable data traffic. A comprehensive strategy that includes packet filtering, behavioral analysis, deep packet inspection, honeypots, deception technology, NetFlow analysis, SIEM, threat intelligence feeds, real-time incident

response systems, encryption analysis, and signature-based and anomaly-based detection is needed to detect network attacks in data traffic. To provide a secure environment for all users, a comprehensive strategy concentrates on ongoing monitoring, updates, and a proactive response plan for strong network security. These systems monitor network activity to identify potential threats. Signature-based IDS relies on predefined patterns, while anomaly-based IDS detects deviations from normal behavior. Machine learning enhances threat detection by analyzing user and network entity behavior for anomalies and attack patterns. Thus, it is crucial to create and use cutting-edge computational techniques developed expressly for identifying network risks in large-scale data settings [8].

The traditional ways for recognizing network assaults, such as standard statistical approaches, behavioral techniques, and Deep Packet Inspection (DPI), present major computational hurdles in big data application scenarios with a vast scale, numerous devices, and massive traffic. These methods demand message parsing and constant monitoring of the health of all networks and devices [9]. But in the domains of deep learning, evolutionary computing, and machine learning, where the emphasis is on creating behavior-based intrusion detection systems, tackling this problem has emerged as a critical issue. It is now possible to automatically identify the behavioral traits of network traffic and the underlying patterns by utilizing these cutting-edge technologies [10, 11]. Based on the characteristics and laws they have learned, they may classify the behaviors in anomalous traffic. Deep learning techniques employ various neural network models to fit highly complex nonlinear functions. Deep learning captures complex nonlinear functions using a variety of neural network models, including FNN, MLP, CNN, and RNN. While RBFN employs radial basis functions for interpolation, GAN generates synthetic data, and autoencoders concentrate on unsupervised learning. Transformers, originally designed for natural language processing, exhibit adaptability through self-attention mechanisms. These models support task-specific adaptation and are updated frequently to reflect new deep-learning architectures and advancements. As a result, they can accurately recognize the complexities of network attack patterns.

A fresh approach to distributed machine learning is federated machine learning. The fact that learning is done locally or on data collection devices is a key component of this strategy. As a result, local models are developed first; then, they are combined to create a global model. This federated learning capacity makes the development of systems that safeguard the confidentiality of sensitive data and personal information possible. A host intrusion detection system (host IDS) is one of these systems. It can monitor and analyze data obtained from and related to a host to find intrusions or other policy violations on that host. It is usually placed on the target host, although some of its capabilities can be centralized if host information is available. Unfortunately, this centralization drastically restricts its capabilities because only a few functions (such as gathering and analyzing system logs) can be done thus without risk. As a result, it can be strictly essential to deploy a host IDS on the target host based on the features and capabilities requested. These devices' integration with embedded industrial machinery is challenging because of their constrained processing capabilities and strict operational constraints.

The following is the paper's primary contribution:

- To address this issue, this article provides two privacy assessment metrics for FL-based HIDSs, including a privacy score that reflects how similar the original and restored traffic attributes are.
- The real-time model was built using the CICIDS2017 dataset, which includes numerous recent attacks.

- An adaptive threshold-correlation algorithm (ATCA) is also introduced to improve detection accuracy by dynamically modifying threshold values based on traffic patterns and intrusion behaviors.
- The FL-HIDS framework is created and tested using a realistic network dataset. Federated Learning focuses on privacy and security in training models on decentralized devices with sensitive data. It requires efficient aggregation and secure communication protocols. Choosing the right architecture for host-based intrusion detection systems (HIDS) on edge devices is crucial in federated learning.
- The federated learning technique successfully uses network device collective intelligence, allowing for continuous learning and adaptation to evolving attack strategies.

This article's remaining sections are organized as surveys: the second section examines and comments on some of the essential connected works. Section 3 explains the proposed strategy. We conduct an empirical analysis and experimental setup in Sect. 4. Finally, Sect. 5 brings this essay to a close.

2 Literature Survey

To detect such attacks, the approach proposed by Khatri et al. [12] examines the transmitted packet properties, judges the models, and visualizes the outcomes. A dataset with more than 80 columns and 10,48,575 rows is used in this investigation. Several machine learning algorithms, including random forest (RF), decision tree (DT), Ada boost classifier (ADA), ridge classifier (ridge), logistic regression (LR), SVM-linear kernel (SVM), naive Bayes (NB), and quadratic discriminant analysis (QDA), were used in the process of building and testing the models. The results reveal an exceptional accuracy rate. When autoML was used, it generated a best-fit algorithm with 99% accuracy and produced specific metrics for each data file. Next, a prediction was made using these saves, and an output CSV was created. The Autoviz package was then used to map the CSV into several displays.

Tang et al. [13] provide a federated learning-based network intrusion detection method in their study. This ground-breaking method enables several ISPs or businesses to collaborate on deep learning training while safeguarding the privacy of their local data. Doing this increases the model's detection accuracy while preserving network communication's anonymity. In this study's trials, network intrusion detection data were collected using the CICIDS2017 system. These studies' findings show that federated learning participants acquire higher detection accuracy. Furthermore, federated learning exhibits performance equivalent to centralized deep learning models when accuracy and other metrics are considered.

In their study, Zczepanik et al. [14] revealed the core ideas of a heuristic algorithm that uses a neural network. Before implementing the machine learning phase, they used a specific dataset and offered details on data processing and any necessary adjustments. It demonstrated how to modify the hyperparameters of artificial neural networks and how they learn. The models' accuracy of more than 98% and their F2-score of more than 95% showed how precise and highly effective this method was in the results. It makes intrusion detection incredibly effective.

According to Jeune et al. [15], internet technology has advanced significantly. Security concerns increased as the number of users increased. The organization must ensure data security. Network traffic packets are examined by anomaly-based intrusion detection

systems (IDS) to look for assaults. Threats were carefully identified using a neural network with an indicator variable utilising rough set for attribute reduction. This approach identifies potential dangers by carefully examining the attributes and applying a rough set theory.

A thorough analysis of network intrusion detection datasets is presented by Ring et al. [16], focusing on 15 distinct features. Their work offers a detailed analysis of these characteristics and how they apply to network intrusion detection. As they provide both a condensed overview and a substantial in-depth table and discussion, their work can be used as a guide when choosing appropriate public datasets for a specific objective. Finally, they arrive at a few conclusions that apply to all future studies that use NIDS datasets. For instance, they talk about how unlikely it is to have a flawless dataset and suggest combining many datasets for evaluation. Thus, their examination does produce helpful information even when it does not offer insight into the performance of specific algorithms or techniques.

A comprehensive intrusion detection system (IDS) was investigated by Ferrag et al. [17] in their work to combat distributed denial-of-service (DDoS) attacks. The suggested IDS employed three alternative models: convolutional neural networks, deep neural networks, and recurrent neural networks. Using the two brand-new real-world traffic datasets, TON_IoT, and CIC-DDoS2019 the researchers assessed the performance of these models. These datasets covered a wide range of DDoS attacks, enabling a thorough evaluation of the model's performance for the binary and multiclass classification types.

Aashmi et al. [18] have developed a novel intrusion detection system that takes advantage of federated learning in the context of jungle computing. Scalability, availability, fault tolerance, and the ability to handle many computing paradigms simultaneously are just some of the benefits that jungle computing provides. We add meta-tags to the job descriptions to highlight frequently completed jobs. This real computing jungle's code is built to function effectively on various hardware, including desktop and laptop computers, clusters, grids, the cloud, low-cost gadgets, and cell phones. The computer system architecture of the Jungle can easily be changed by just changing the meta-tags given to jobs. Frameworks for federated learning with privacy enhancements are more resistant to models of anomaly detection. The federated learning architecture is vulnerable to attacks from hostile parties. Hence, a more resilient paradigm was used in jungle computing by using Java parallel processing in various application scenarios.

Liu and Shi [19] first suggested a hybrid intrusion detection system (IDS) using genetic algorithms (GA). The results of a random forest (RF) classifier were used to generate a new fitness function. In each iteration, the chromosome with the highest value replaced the one with the lowest value. According to test results, the model consistently achieves more than 90% accuracy when applied to the reference datasets NSL-KDD and UNSW-NB15.

The benchmark datasets Bot-IoT, UNSW-NB15, and CIRA-CIC-DOHBrw-2020 were used by Halim et al. [20] to investigate their suggested GA-based feature selection approach (FBFS). The fitness function chooses the characteristics with the fewest correlations for categorization in their experiment using unsupervised GA. Their recommended method, which combines the three classification models XGBoost, k-nearest neighbor (KNN), and support vector machine (SVM), has a 99.80% success rate.

2.1 Limitations of Existing System

- Federated learning relies on several parties or hosts contributing local data for model training. However, acquiring real-time traffic data for host intrusion detection can be

difficult. Because not all hosts are ready to reveal sensitive traffic information, data availability may be limited, and models may be biased.

- Frequent communication between the central server and participant hosts is required in federated learning. Real-time traffic monitoring necessitates low-latency processing, and the increased connection overhead might cause delays, reducing the system's overall performance and responsiveness.
- Hosts' hardware capabilities, network circumstances, and data distributions might differ dramatically in a federated learning environment. This heterogeneity can make reaching uniform convergence during model training difficult. Different hosts may have varying levels of accuracy, resulting in potential variances in intrusion detection system performance.
- Federated learning seeks to protect the confidentiality of regional data stored on participating hosts. Since real-time traffic analysis involves sensitive information regarding network activity, protecting data security and privacy is critical. Data integrity must be protected, and potential assaults on the federated learning process, such as model poisoning or inference attacks, must be avoided.
- Real-time traffic analysis frequently deals with a high volume of data supplied by multiple hosts simultaneously. Scaling federated learning to accommodate an increasing number of servers while processing enormous amounts of traffic data can be difficult. The system must manage growing computing and communication demands while being

2.2 Problem Identification of Existing System

- Traditional methods for detecting host intrusions in real-time traffic analysis frequently rely on centralized data collection, in which all network traffic data is provided to a central server for analysis. However, this strategy has significant drawbacks, including privacy concerns, bandwidth constraints, and the possibility of single points of failure. Federated learning addresses these challenges by allowing collaborative analysis while keeping data dispersed. On the other hand, implementing federated learning for host intrusion detection in real-time traffic analysis presents its own set of obstacles.
- In real-time traffic analysis, networks can be highly heterogeneous, containing many devices, operating systems, and network topologies. This variety presents a difficulty for federated learning since the models must be trained in various network conditions while assuring compatibility and accuracy across all participating servers. For federated learning to be most successful and efficient in host intrusion detection, network heterogeneity problems must be identified and fixed.
- Hosts involved in federated learning for intrusion detection may have varied computational capabilities and resource constraints. Some hosts may need more processing capacity, memory, or energy, limiting their ability to actively engage in the learning process. Designing federated learning algorithms that can accept such resource-constrained hosts is a considerable problem while retaining the system's overall efficiency and accuracy.
- Host intrusion detection involves sensitive network traffic data and potential security vulnerabilities. Data remains on the hosts with federated learning, decreasing privacy risks. However, maintaining data privacy and security while aggregating and updating models across dispersed servers becomes crucial. Potential vulnerabilities must be addressed, such as model poisoning attacks, information leakage during communication, or malicious behavior from collaborating hosts.

- Real-time traffic analysis necessitates the discovery and response to intrusions on time. Due to the necessity for communication, coordination, and aggregation of model changes over numerous hosts, federated learning imposes a significant delay. Balancing the trade-off between real-time analysis capabilities and precise intrusion detection via federated learning is challenging.

3 Proposed System

In host intrusion detection systems (HIDS) that seek odd behavior, deep learning (DL) approaches are often employed to discover damaging data. Deep learning strategies for detecting damaged data have included autoencoders, variational autoencoders (VAEs), generative adversarial networks (GANs), outlier detection models such as Isolation Forests and one-class SVM, residual analysis, and data augmentation and denoising techniques. These techniques assess reconstruction errors, generate synthetic samples, detect deviations from normal data distribution, and improve model resilience to damage. The overall efficacy of identifying damaged data can be increased by combining techniques or using hybrid approaches. Autoencoders, along with anomaly detection techniques like variational autoencoders and outlier detection algorithms, are extensively used. Autoencoders reconstruct input data, while anomaly detection models identify deviations from normal patterns, indicating potential data damage. The federated learning (FL) architecture makes DL-based methods more valuable, which protects user data while training a global model. By decentralizing the training process, the federated learning (FL) architecture revolutionizes deep learning (DL) methods, allowing models to be trained on local data without threatening user privacy. FL iteratively improves a global model by sending model updates rather than raw data to a central server, ensuring that sensitive user information remains secure on individual devices. This approach not only protects user data but also increases the value of DL-based methods, particularly in applications where privacy is important. FL strikes a balance between the power of deep learning and the need to protect individual privacy by enabling collaborative learning across distributed devices, ultimately establishing a global model that has learned from diverse, decentralized data sources. However, FL-based HIDSs' privacy protection has yet to be fully assessed. To tackle this challenge, we offer two privacy assessment metrics for FL-based HIDSs, including a privacy score that rates how close the original and restored traffic attributes are. The real-time model was created using the CICIDS2017 dataset, which includes numerous recent attacks. An Adaptive Threshold- correlation Algorithm (ATCA) dynamically adjusts threshold values based on traffic patterns and incursion behavior to improve detection accuracy. ATCA is designed to enhance detection accuracy by dynamically adjusting threshold values based on real-time observations of traffic patterns and incursion behavior. The process involves continuous monitoring, initial threshold setting, real-time analysis, dynamic threshold adjustment, correlation with known anomalies, and a feedback loop for continuous improvement. By adapting to changing conditions and learning from detections, ATCA provides a flexible and responsive approach to identifying both known and emerging threats in a given system. A realistic network dataset is used to design and test FL-HIDS. Experimental results show that the suggested technique is more precise and scalable than standard intrusion detection systems. The federated learning strategy uses network devices' collective intelligence to learn and adapt to emerging attack strategies continuously.

Federated learning-host intrusion detection systems (FL-HIDS) development and testing necessitate careful consideration of network topology, communication latency, security, privacy of data, and diversity of hosts and data. It is critical to ensure efficient model synchronization as well as manage network topology and communication latency. The variety of hosts and data, including operating systems, applications, and network behaviors, makes robust intrusion detection difficult. For accurate and resilient FL-HIDS, decisions on model architecture, hyperparameters, and handling adversarial attacks are critical. Other critical factors are data distribution, imbalance, scalability, and regulatory standard compliance. A realistic network dataset, which includes a variety of traffic patterns and attack scenarios, is required for training and testing. The FL-HIDS plan is depicted in a block diagram in Fig. 1. Users give the data firewall information. A firewall is a constant filter for data attempting to enter a network, screening it for potential threats and blocking those that do. Before reaching the firewall, the attacker attacks the user data. This prevents data from being written to a dataset. Data is gathered from a dataset. Techniques for pre-processing collected data. Data transformation and pre-processing to remove outliers and unnecessary instances. The datasets used contain values that are symbolic, continuous, and binary. The deployment of federated learning for host intrusion detection calls for sharing privacy concerns among organizations to safeguard user privacy and expand the knowledge base of learning models. Host intrusion detection requires analyzing sensitive data, emphasizing the importance of collaborative discussions among organizations. Open dialogue fosters trust, transparency, and a commitment to ethical practices, aiding in the identification and strategy development to minimize privacy-related incidents. Introducing a more extensive

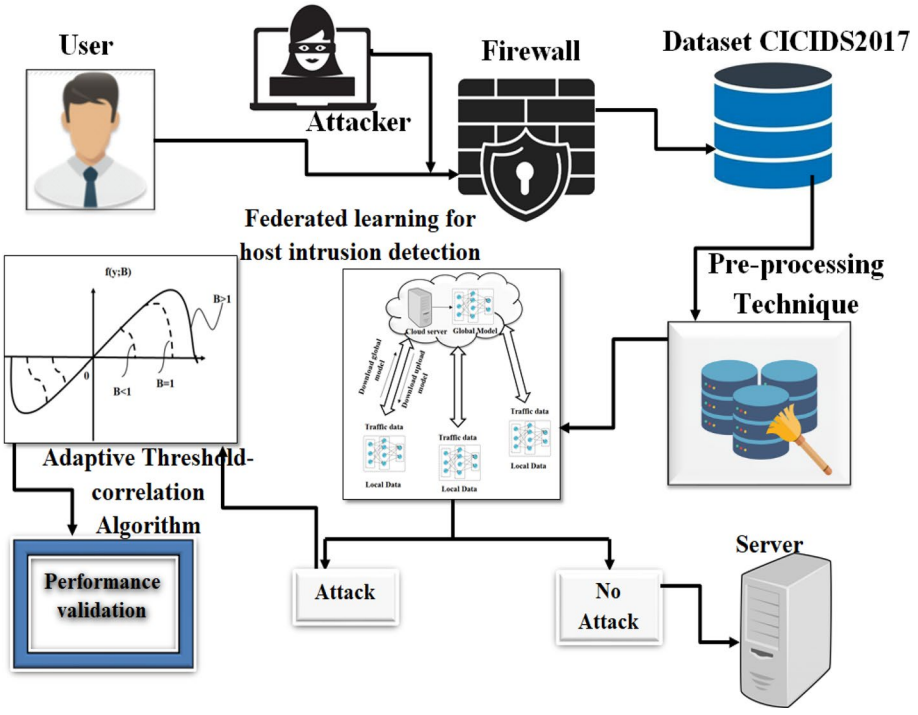

Fig. 1 Proposed method of FL-HIDS

variety of attacks and eliminating any attack variants within the learning model is crucial to guaranteeing the accurate detection accuracy for previously unrecognized traffic within an organization. It boosts the model's robustness, enhancing its capability to recognize and defend against a broader range of threats. This strengthens overall security and adaptability, resulting in more effective and reliable performance across diverse scenarios. The Adaptive Threshold-correlation Algorithm (ATCA) can improve detection accuracy by dynamically altering threshold levels based on traffic patterns and intrusion behaviors if an attack is detected. Without any attacks, the information is transmitted to the server.

3.1 Dataset Description

The Canadian Institute for cyber security debuted the CICIDS 2017 in 2017 [21]. The dataset includes several benign attacks that are further divided into 15 subcategories and seven primary categories of recent attacks [22]. Two million eight hundred thirty thousand one hundred eight occurrences total in the sample, with 83.3% benign and 16.7% malicious traffic. Most of the most recent real-world attacks are included in this dataset, which has been rationalized. From a five-day data collection campaign from July 3 to July 7, 2017, a program called CICFlowmeter was used to extract various flow-based, time-based, content-based, host-specific, and additional auxiliary features and their labels. Some attacks include brute force SSH, brute force FTP, Heartbleed, DoS, infiltration, botnet, web assault, and DDoS. The abstract behavior of 25 people was built utilizing a range of network protocols, including FTP, HTTPS, SSH, HTTP, and email protocols, to give the dataset more realism. Several CSVs are also included in the dataset for usage in deep learning and machine learning applications, along with the entire packet payload in PCAP format for real-time implementation. Table 1 has a thorough list of complete signatures for several attacks and the relevant types.

3.2 Data Pre-processing

In the world of data mining, the preprocessing phase of the data is both significant and time-consuming. Dealing with accurate data frequently gathered from various systems that may be unpredictable, redundant, incomplete, or inconsistent is a requirement. Accurate data with unexpected patterns or outliers can lead to unpredictable model behavior. Redundant data increases the risk of overfitting and reduces generalization. Incomplete data introduces bias, hindering pattern extraction and impacting predictions. It is essential to transform raw data into a format appropriate for analysis and information discovery. The transformation of raw data for analysis is crucial due to the inherent complexity and lack of structure in raw data, often containing errors and inconsistencies. This process involves cleaning and pre-processing to enhance data quality and enable integration from various sources. Normalization and standardization play a role in achieving consistent presentation, while feature engineering enriches the dataset. Dimensionality reduction simplifies the data, enhancing analytical techniques and models. Then, the transformed data aligns with governance standards, assists compliance efforts, and facilitates exploration,

Table 1 The CICIDS2017 dataset provides descriptions of numerous attacks and their kinds

Name of the attack	Subcategories of attack	Attack objectives
Brute Force	SSH-Patator, FTP-Patator	To access an account, practically all possible combinations of login information are tried
Botnet	Bot	Malicious software known as a "bot" infiltrates a computer system intending to carry out instructions sent by a remote attacker
DOS/DDOS	DoS GoldenEye, DoSslowloris, Heartbleed, DoSSlowhttptest, DoSHulk	Distributed denial of service (DDoS) and denial of service (DoS) techniques can obstruct a server, service, or network's normal traffic flow. They entail flooding the target's network or infrastructure with a large amount of Internet traffic to block access for authorized users

visualization, and interpretation. Data manipulation and eliminating redundant and outlier events were part of the preparation procedure in this investigation. Symbolic, continuous, and binary values can be found in the datasets [23]. Data preparation involves transforming raw data for effective processing and analysis. Key steps include collecting, cleaning, and labeling data for machine learning algorithms, followed by exploration and visualization to ensure suitability for analysis.

3.3 Federated Learning for Host Intrusion Detection

Organizations should employ a federated learning strategy to share information about privacy issues to broaden the information base of learning models and prioritize user privacy. The learning model is exposed to various benign and attack variations to accomplish consistent detection accuracy through beforehand undetected organizational traffic. This exposure gives the model a chance to become accustomed to a broader variety of good and bad events, improving adaptability and ensuring consistent performance. Federated learning is a machine learning technique that allows for model training on decentralized devices or servers while maintaining data privacy by keeping data local. It enables diverse data sources, reduced latency through edge computing, and adaptability to dynamic environments in real-time traffic analysis. Federated learning enables local threat detection, privacy-preserving collaboration, adaptive threat models, reduced false positives, and scalability through a distributed architecture for intrusion detection. By fostering robustness and resilience, this approach enhances the model's ability to generalize across diverse situations. It enables the model to learn effectively from both successful and challenging experiences, equipping it to handle real-world complexities, unforeseen scenarios, and noisy data. The proposed architecture facilitates collaboration between organizations through the sharing of cyber intelligence and insights. Additionally, enterprises that cannot now amass and keep enough network traffic for training learning models by partnering with other businesses can

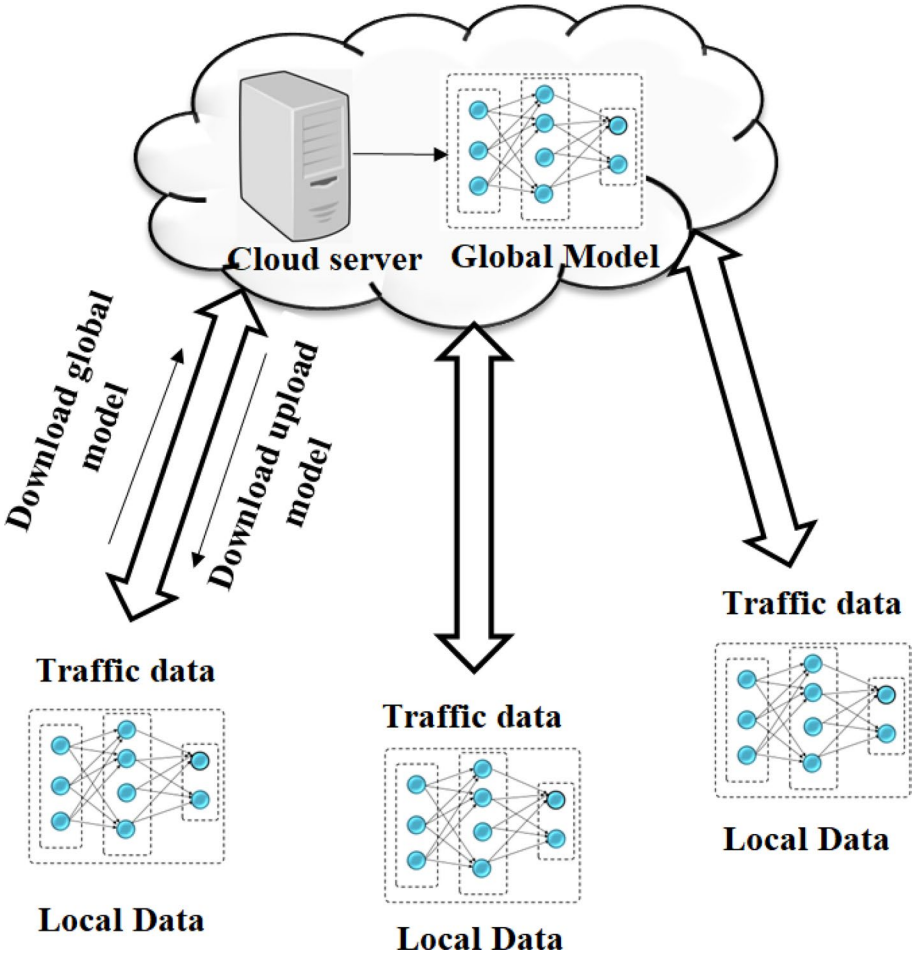

Fig. 2 Federated learning's organizational structure

create efficient deep learning (DL) systems. Our method tackles the issue of data scarcity and enables the creation of DL-based HIDS without the need for a sizable training data set. Each member provides a small number of data samples, which make up the system's success. The framework for federated learning is shown in Fig. 2.

Furthermore, federated learning assures the confidentiality and accuracy of sensitive individuals' network information by distributing local network data samples across numerous companies. A central server oversees and manages the autonomous local model training in this federated learning setup. Despite being housed inside one of the partner organizations, the project's global server is designed to be hosted externally by a responsible third party, such as cloud computing. Selecting an external hosting solution for the global server in federated learning depends on scalability and operational efficiency. Prioritize strong encryption, strict access controls, regulatory compliance, and thorough vetting of hosting providers to safeguard sensitive data. Each member organization must consistently log its local network data traffic according to the framework's primary requirements. References [24, 25] provide a comprehensive

list of advantages to using a uniform feature set. Federated learning is more effective since the global model may identify significant patterns within a predetermined range of data attributes. The global model's parameters and organizational structure were created to comply with the industry-recognized network logging format.

The whole procedure is distinct by Algorithm 1, where P is the member organizations indexed by p, w is the initialized regular of limitations, t is the federated learning round, and m is the average learning rate across all member organizations. The letters E stand for the local learning rate, the regional training set, the number of local epochs, the prediction loss (xi, yi), and the local prediction loss. The regional training batch's size is B., similar to traditional federated learning approaches;

Step 1 An ML model with a set architecture and parameters is launched by a global server to begin the process.

Step 2 To each participant, the model is sent.

Step 3 The framework is refined and trained nearby using internal network data examples.

Step 4 The weights are modified before returning to the primary server.

Step 5 The system employed is FedAvg [26], in which the server pools the weights supplied by other organizations to create a more effective intrusion detection model. This updated model considers the characteristics of each participant's network using a more accurate set of variables.

The FedAvg approach is defined as

$$v_{t+1} \leftarrow \sum_{l=1}^{L} \frac{m_l}{m} v_{t+1}^{l} \tag{1}$$

To enhance detection performance in all network scenarios, these five phases can be repeated to build a single federated learning round.

Global Server Executes

Initialize v_0
for each federated learning round t=1,2,….:do
  $S_t \leftarrow$ (Set of K organizations)
  for each organizations l € S_t : do
    $v^{l}_{t+1} \leftarrow Local\ organization\ Update(l, v_t)$
    $v_{t+1} \leftarrow \sum_{l=1}^{L} \frac{m_l}{m} v^{l}_{t+1}$
  end
end
Local organization Update(l, v)
Run on organization l
$\aleph \leftarrow (split\ \Re_l \text{ int } o\ batches\ of\ size\ N)$
for each local epoch I from 1 to E:do
  for batch $n \in \aleph n : do$
    $v \leftarrow v - n\nabla 1(v; n)$
    return v to server
  end
end

This work broadens the application of federated learning by treating each local client as an autonomous entity with a distinct selection of varied data examples. Creating a robust information security program is crucial for businesses to minimize the impact of data breaches, safeguard data privacy, and detect/prevent threats. This includes managing authorized user access, maintaining integrity with approved changes, and ensuring confidentiality through effective access control. The direct result is the creation of a comprehensive DL-based HIDS through inter-organizational collaboration without the requirement to disclose participant data to protect participant privacy. The final model's capacity to identify threats from a more extensive range of sources is essential for an organizational defensive system. The defensive system is an important component of organizational security, which protects IT infrastructure from unauthorized access and threats using measures such as access controls and encryption. It incorporates firewalls and intrusion prevention systems to monitor and control network traffic, while regular assessments, incident response planning, and user awareness initiatives improve its effectiveness. Integration with intrusion detection systems enables timely detection and response to malicious activities. The system's continuous improvement, adherence to compliance, and adaptability to emerging threats all contribute to a resilient security posture, making it critical for proactive intrusion detection and prevention. Creating a strong information security program for businesses is vital. It helps identify and prevent risks, protects data privacy, and minimizes the impact of potential data breaches. This includes ensuring confidentiality through access control, maintaining integrity with authorized changes, and ensuring availability for authorized users. As a result, a solid learning model is developed that can discern between good and bad heterogeneous traffic thanks to its extensive knowledge and insights. When there are modifications to the regular traffic patterns due to upgrades to the Standard Operating Environments (SOE), sophisticated models like these can potentially reduce the incidence of false warnings. Federated learning allows personalized model adjustments on local devices, reducing centralized data transfer. This improves accuracy in anomaly detection by adapting to local conditions, enhances privacy by keeping data localized, and minimizes false warnings through collaborative learning in dynamic environments. This improvement is because the models have become more adaptable to variations in benign traffic distribution due to learning from secure usage across multiple networks. Extracting harmful patterns from a broader range of attacks that target numerous organizational networks also promotes a better incidence of sophisticated and zero-day attack detection.

3.4 Adaptive Threshold-Correlation Algorithm (ATCA)

Look at Fig. 3's general nonlinear function first.

The function $f(y;B)$ must be peculiar and limited. $f(y;B)$ can change its shape by inserting a parameter B known as the "threshold," as seen in the picture.

Assume f(x; A) is written as

$$f(y;B) = B.\widetilde{f}(y/B) \tag{2}$$

where $\widetilde{f}()$ is a fundamental function with attributes.

$$\text{(a)}\ \ \widetilde{f}(u) \to u\ as\ |u| \to 0; \tag{3}$$

$$\text{(b)}\ \ \widetilde{f}(u) \to A.\text{sgn}(u)\ as\ |u| \to \infty,\ \ with\ A \geq 0 \tag{4}$$

It is worth noting that $f(y:1)$ equivalents are the basic function $\widetilde{f}(y)$.

The nonlinear correlation function, designated as f(x; A), is incorporated into the adaptation procedure's implementation. The following is a presentation of the equation for updating the tap weights.

$$d^{(n+1)} = d^{(n)} + \alpha_c f(e_n + v_n; B^{(n)}) b^{(n)} \tag{5}$$

where $b^{(n)} = [b_n, b_{n-1}, ...b_{n-N+1}]^T$ stands for the tap weight vector at time n (T for transposition), $d^{(n)} = [d_0^{(n)}, d_1^{(n)}, ...d_{N-1}^{(n)}]^T$ for the tap input reference signal vector at time n, en for the tap error signal, n for the tap additive noise, N for the tap count, and $\theta^{(n)} = d_{opt} - d^{(d)}$ for the step size. If the definition of "tap error vector" is (n) = $\mathrm{d_{opt}}$

$$\theta^{(n+1)} = \theta^{(n)} - \alpha_c f(_n + v_n; B^{(n)}) b^{(n)} \tag{6}$$

and

$$e_n = b^{(n)T} \theta^{(n)} \tag{7}$$

The name of the proposed method suggests an adaptive control of the threshold $B^{(n)}$. The long-term root mean square value $e_n + v_n$ is used to determine how proportional BB should be.

Where M is a coefficient ranging from around 1–2, and $(\sigma_L^{(n)})^2$ is the

$$B^{(n)} = M\sigma_L^{(n)} \tag{8}$$

and

$$(\sigma_L^{(n)})^2 = (1 - \beta_L)(e_n + v_n)^2 + \beta_L(\sigma_L^{(n-1)})^2 \tag{9}$$

The computation uses the long-term average of $(e_n + v_n)^2$ and include β_L s the leak factor inside the leaky accumulator to compute (9).

The filter convergence follows the following trend when the adaptive threshold constrained algorithm (ATCA) is used for adaptive filtering. The advanced telecommunications computing architecture (ATCA) is a standardized framework for signal processing and adaptive filtering because of its robustness and scalability. Its integrated

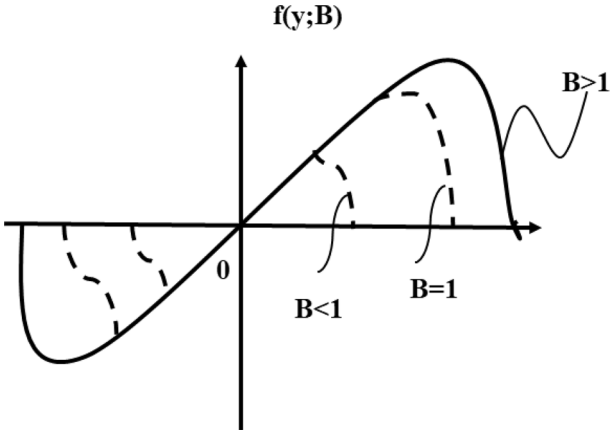

Fig. 3 Adaptive "threshold" parameter for a nonlinear function

management tools improve dependability by monitoring in real-time and optimizing parameters. ATCA prioritizes reliability with redundant components and fault tolerance, making it a preferred choice in these fields. Its adaptability enables parameter optimization for efficient adaptive filtering and signal processing. The error signal, along with $(\sigma_L^{(n)})^2$ $B^{(n)}$ and, tends to be significant during the early stages. Because the leak factor is so close to unity, BB remains considerable even as the filter gradually converges and the erroneous signal's strength decreases. When the threshold parameter B surpasses the total size of the error and noise signal, its behavior resembles that of the least mean square algorithm (LMSA). In machine learning and signal processing, one common adaptive filtering method is the least mean square algorithm (LMSA). The least mean square algorithm (LMSA) is a widely used adaptive filter algorithm for minimizing mean squared error in real-time applications such as noise reduction and system identification. It iteratively adjusts filter weights based on the instantaneous error and input data using stochastic gradient descent. The update rule, convergence influenced by the learning rate, and applications in adaptive filtering and machine learning are important considerations. Its adaptability is increased by regularization and variations like Normalized LMSA. LMSA is a computationally efficient algorithm with adaptive step size options for dynamic adjustments. Its adaptability and simplicity make it popular for real-time tasks. Its capacity to iteratively modify filter coefficients, minimizing mean square error in changing system parameters, makes it popular for real-time applications.

The error signal becomes negligible, possibly even smaller than the power of the background noise, after convergence and achieving the ultimate mean squared value of en + n. As a result, in the steady state, the noise power is a few dB lower than the threshold $B^{(n)}$.

As a result, the lower threshold effectively "clips" the peak of the impulse and preserves the majority of the residual error even if a high-power stimulation is applied to the system after convergence Fig. 3.

4 Experiments, Results, and Discussion

This section presents our proposed method's extensive performance summary and experimental data.

4.1 Experimental Setup

A federated learning system based on evolutionary computation has been created to put the findings into practice for host intrusion detection in real-time traffic analysis. To detect intrusions on individual hosts within a network, the system analyzes network traffic in real-time. The practical application of these findings demonstrates the system's ability to address real-world security challenges. Empowers real-time analysis of network traffic for swift responses to potential intrusions, bolstering overall security. Safeguards data privacy by localizing sensitive information on individual hosts, minimizing the risk of exposure. Emphasizes host-level detection of anomalies and potential security threats. The Raspberry Pi 3 Model B + is used for testing the proposed FL-HIDS. This Raspberry Pi model includes 1 GB of LPDDR2 SDRAM, a potent 1.4 GHz 64-bit quad-core processor, and compatibility with Bluetooth 4.2 as well as 2.4 GHz and 5 GHz IEEE 802.11b/g/n/ac wireless LAN. It also provides several other contemporary features crucial for the testing

procedure. It is an edge computing tool and may be used with numerous ML models. The Raspbian operating system and many Linux variants power this gadget. Python, a high-level, multi-paradigm programming language, coupled with a few Python modules and TensorFlow (an open-source software library), is used to generate the ML source code. Two alternatives are provided for writing Python code: SSH and Mu editor. The development of an edge tensor processing unit for embedded and mobile devices to speed up TensorFlow calculations. The performance on edge devices is achieved by offloading machine learning inference tasks to specialized hardware designed for tensor operations in neural networks. This dedicated hardware, proficient in matrix multiplications, reduces the main processor workload, leading to faster TensorFlow computations at the edge. The system is optimized for effective performance and security using a 64-bit Ubuntu Linux operating system, an Intel(R) Core (TM) i5-8250U processor, 8.00 GB (7.89 GB useable), and the designated FL-HIDS.

4.2 Evaluation Metrics

A few measuring criteria are used to assess the suggested edge-based IDS. The performance of the model is evaluated using the subsequent measures. Precision, false alarm rate (FAR), recall, accuracy, attack detection rate (ADR), and F-score. TP, TN, FP, and FN are the four different measuring factors that must be included in the computation. In Table 2, each aspect is fully explained. Table 3 shows the evaluation metric's complete description and computation equation.

4.2.1 Precision Analysis

In Fig. 4 and Table 4, the precision of the FL-HIDS method is compared with that of existing techniques. The graph demonstrates how the DL approach has increased efficiency with precision. For instance, the XGBoost, ADA, QDA, and LR models' respective precision values for 100 data are 76.12%, 87.67%, 81.11%, and 91.45%, respectively, as opposed to the FL-HIDS model's precision with 94.55%. However, the FL-HIDS model has performed best with various data. Similarly, under 500 data, the FL-HIDS has a precision of 96.12%, while the corresponding precision values for XGBoost, ADA, QDA, and LR are 80.45%, 90.56%, 86.45%, and 93.11%.

4.2.2 Recall Analysis

In Fig. 5 and Table 5, the recall of the FL-HIDS methodology is compared with that of other methods. The graph demonstrates how the DL approach has increased efficiency with recall. For instance, the XGBoost, ADA, QDA, and LR models' respective recall values for 100 data are 67.12%, 72.44%, 79.34%, and 85.12%, respectively, as opposed to the FL-HIDS model's recall value of 92.18%. However, the FL-HIDS method has been shown to perform best with various data. Similar to this, under 500 data, the FL-HIDS has a recall value of 97.45%, while the corresponding recall values for XGBoost, ADA, QDA, and LR are 71.43%, 76.66%, 83.44%, and 89.23%.

Table 2 Measuring variables

Factors	Description
False positive rate (FP)	The total number of incidents that were misclassified as routine attacks
False positive rate (FP)	The total number of incidents that were labeled attacks, even though they were routine
True positive rate (TP)	The total number of situations that were mislabeled as usual yet were, in fact, typical
True negative rate (TN)	The total number of attacks that were incorrectly classified as attacks

4.2.3 F-Score Analysis

In Fig. 6 and Table 6, the f-score of the FL-HIDS method is compared with that of other methods. The graph demonstrates how the DL approach has increased efficiency with f-score. For instance, the XGBoost, ADA, QDA, and LR models' respective f-score values for 100 data are 80.13%, 88.45%, 84.87%, and 90.67%, respectively, as opposed to the FL-HIDS model's f-score value of 94.19%. However, the FL-HIDS method has been shown to perform best with various data. Like this, under 500 data, FL-HIDS has an f-score value of 97.12%, while the corresponding f-score values for XGBoost, ADA, QDA, and LR are 83.76%, 90.13%, 87.12%, and 93.87%.

4.2.4 Accuracy Analysis

In Fig. 7 and Table 7, the accuracy of the FL-HIDS method is compared with that of other methods. The graph demonstrates how the DL approach has increased efficiency with accuracy. For instance, the XGBoost, ADA, QDA, and LR models' respective accuracy values for 100 data are 82.67%, 88.12%, 85.19%, and 91.43%, respectively, as opposed to the FL-HIDS model's accuracy of 96.98%. However, the FL-HIDS method has been shown to perform best with various data. Similar to this, under 500 data, the FL-HIDS has an accuracy of 99.90%, while the corresponding accuracy values for XGBoost, ADA, QDA, and LR are 84.45%, 90.34%, 87.98%, and 95.19%.

4.2.5 Attack Detection Rate Analysis

In Fig. 8 and Table 8, the attack detection rate of the FL-HIDS method is compared with that of other methods. The graph shows how the DL approach has increased efficiency with attack detection rate. For instance, the XGBoost, ADA, QDA, and LR models' respective attack detection rate values for 100 data are 74.12%, 82.78%, 79.13%, and 87.34%, respectively, as opposed to the FL-HIDS model's attack detection rate of 92.87%. However, the FL-HIDS method has been shown to perform best with various data. Similar to this, under 500 data, the FL-HIDS has an attack detection rate of 96.78%, while the corresponding attack detection rate values for XGBoost, ADA, QDA, and LR are 78.88%, 86.12%, 81.87%, and 91.45%.

Table 3 Performance indicators

Measurement metric	Description	Formula
Precision	To calculate precision, divide the value of each true positive by the total number of true positives and false positives	$\Pr ecision = \frac{T_p}{T_p+F_p}$
Recall	The proportion of true positive divided by the sum of true positive and false negative is known as the recall	$\text{Re}call = \frac{T_p}{T_p+F_N}$
F-score	The harmonic mean of precision and recall is the F-score	$F-Score = 2 * \frac{R*P}{R+P}$
Accuracy	The accuracy measure describes the percentage of successfully categorizing the test data. It is derived by dividing the dataset's total number of records by the accurate classification of each class	$Accuracy = \frac{T_p+T_N}{T_p+T_N+F_p+F_N}$
Attack detection rate	The attack detection rate serves as a gauge of how well a model detects attacks	$ADR = \frac{\sum_{i=2}^{C} T_{p_i}}{\sum_{i=2}^{C} T_{p_i}+FP_i}$
False alarm rate	False alarm rate displays the classes that do not attack yet are labeled as such. Regular traffic is made feasible by this statistic, which also quantifies the FN	$FAR = \frac{F_{N_i}}{T_{p_i}+F_{N_i}}$

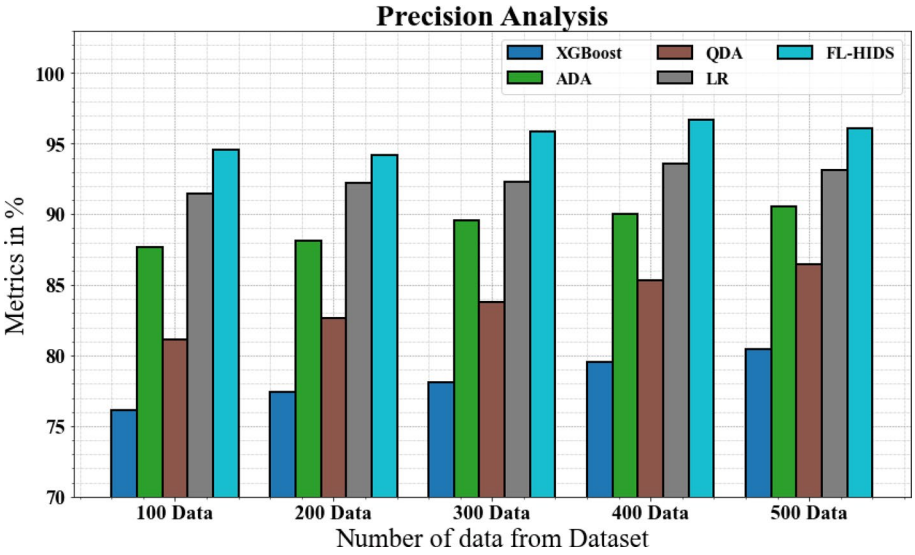

Fig. 4 Precision analysis for FL-HIDS method

Table 4 Precision Analysis for FL-HIDS method

Number of data	XGBoost	ADA	QDA	LR	FL-HIDS
100	76.12	87.67	81.11	91.45	94.55
200	77.45	88.12	82.67	92.19	94.19
300	78.12	89.56	83.77	92.33	95.87
400	79.56	89.99	85.34	93.56	96.66
500	80.45	90.56	86.45	93.11	96.12

4.2.6 False Alarm Rate Analysis

Figure 9 and Table 9 present a comparative analysis of the FL-HIDS method's False Alarm Rate with other current methodologies. The DL approach has produced better performance with a lower False Alarm Rate, as the figure illustrates. The False Alarm Rate for FL-HIDS, for instance, is 23.19% with 100 data, whereas the XGBoost, ADA, QDA, and LR models have somewhat higher False Alarm Rates of 43.98%, 39.45%, 34.19%, and 29.76%, respectively. On the other hand, for various data numbers and low False Alarm Rate values, the FL-HIDS model has demonstrated maximum performance. Comparably, under 500 data, FL-HIDS's False Alarm Rate is 27.77%, but the corresponding values for the XGBoost, ADA, QDA, and LR models are 47.87%, 42.87%, 38.88%, and 33.87%.

5 Conclusion

This study explored deep learning (DL) methods frequently used in host intrusion detection systems (HIDS) that look for unusual activity to identify malicious data. The efficacy of DL-based techniques is increased by using the federated learning architecture, which

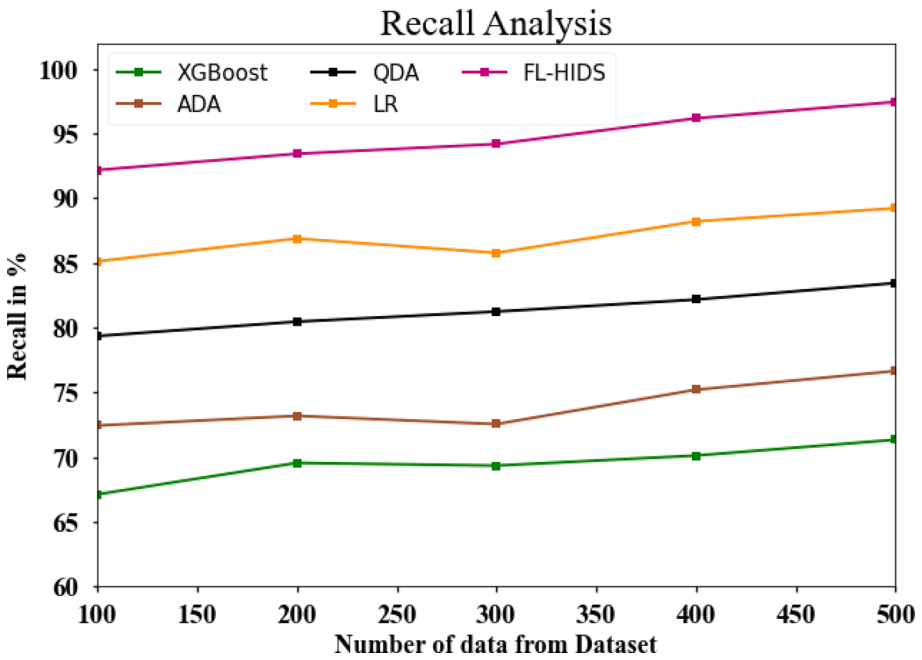

Fig. 5 Recall analysis for FL-HIDS method

Table 5 Recall analysis for FL-HIDS method

Number of data	XGBoost	ADA	QDA	LR	FL-HIDS
100	67.12	72.44	79.34	85.12	92.18
200	69.55	73.19	80.45	86.88	93.44
300	69.34	72.55	81.23	85.77	94.19
400	70.12	75.21	82.16	88.21	96.18
500	71.43	76.66	83.44	89.23	97.45

enables several users to train a global model while protecting the privacy of each user's data. However, a comprehensive investigation of how well FL-based HIDSs protect against recognized privacy concerns using existing protections has yet to be conducted. To address this issue, we provide two privacy assessment metrics for FL-based HIDSs, including a privacy score that quantifies how similar the original and restored traffic attributes are. The real-time model was built using the CICIDS2017 dataset, which includes numerous recent attacks. An Adaptive Threshold-correlation Algorithm (ATCA) is also introduced to improve detection accuracy by dynamically altering threshold values based on traffic patterns and incursion behaviors. The FL-HIDS framework was created and tested using a realistic network dataset. Studies show that the suggested method outperforms the most cutting-edge intrusion detection systems regarding detection accuracy and scalability. The federated learning technique successfully harnesses the network device's collective intelligence, enabling continuous learning and adaptation to change attack strategies. In terms of system performance, the suggested method beat five currently used techniques, including simultaneous XGBoost, Ada boost classifier (ADA), quadratic discriminant analysis

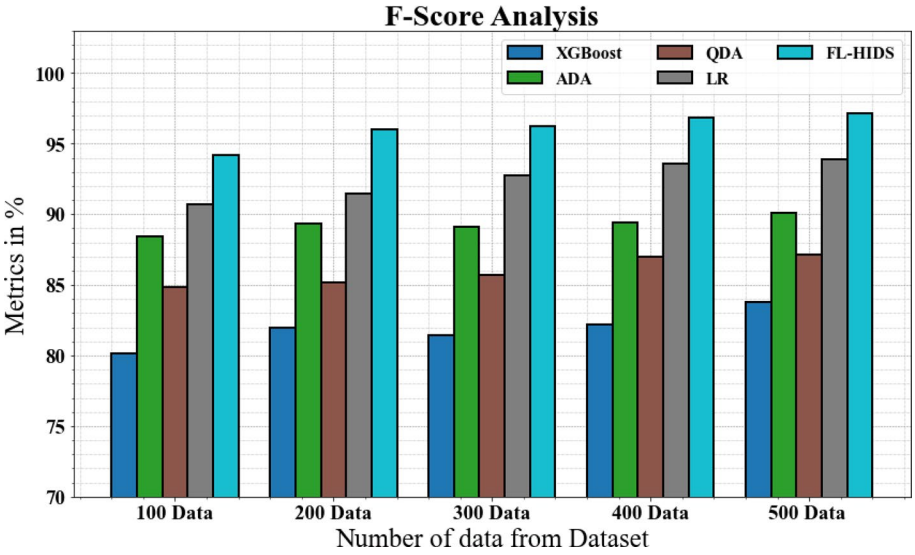

Fig. 6 F-score analysis for FL-HIDS method

Table 6 F-score analysis for FL-HIDS method

Number of data	XGBoost	ADA	QDA	LR	FL-HIDS
100	80.13	88.45	84.87	90.67	94.19
200	81.98	89.34	85.18	91.45	95.98
300	81.45	89.12	85.67	92.78	96.23
400	82.17	89.44	86.98	93.55	96.87
500	83.76	90.13	87.12	93.87	97.12

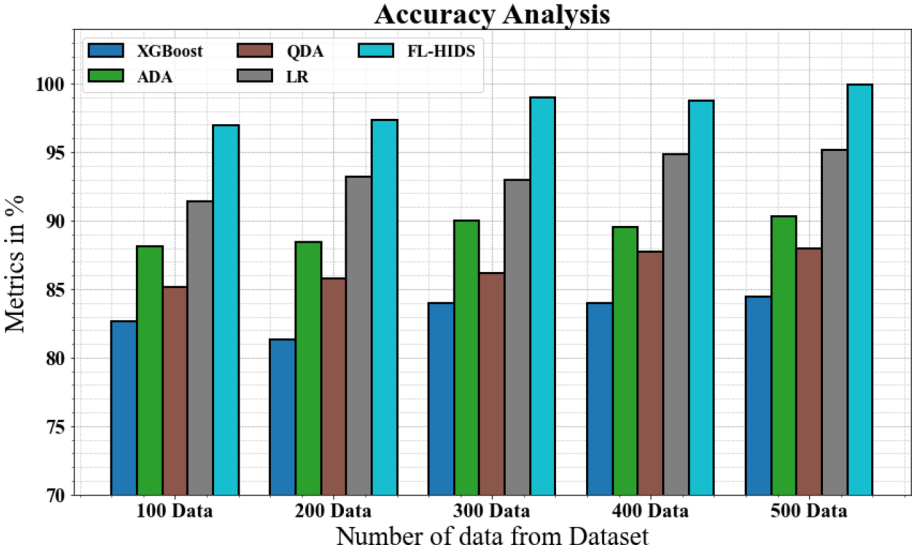

Fig. 7 Accuracy analysis for FL-HIDS method

Table 7 Accuracy analysis for FL-HIDS method

Number of data	XGBoost	ADA	QDA	LR	FL-HIDS
100	82.67	88.12	85.19	91.43	96.98
200	81.34	88.45	85.78	93.19	97.34
300	83.98	89.98	86.18	92.98	98.99
400	83.98	89.54	87.77	94.87	98.76
500	84.45	90.34	87.98	95.19	99.90

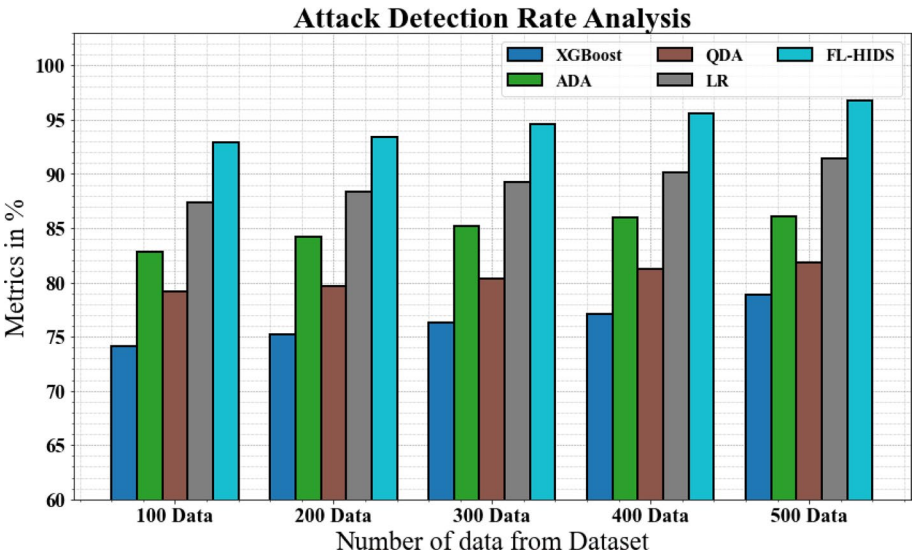

Fig. 8 Attack detection rate analysis for FL-HIDS method

Table 8 Attack detection rate analysis for FL-HIDS method

Number of data	XGBoost	ADA	QDA	LR	FL-HIDS
100	74.12	82.78	79.13	87.34	92.87
200	75.17	84.19	79.67	88.34	93.44
300	76.34	85.17	80.34	89.23	94.55
400	77.12	85.98	81.23	90.13	95.61
500	78.88	86.12	81.87	91.45	96.78

(QDA), and logistic regression (LR). According to research findings, the FL-HIDS ideal has average precision rates of 96.12%, recall rates of 97.45%, f-scores of 97.12%, accuracy rates of 99.90%, attack detection rates of 96.78%, false alarm rates of 27.77%. Future projects will encompass a range of activities. For network security operations, analyzing datasets suited for replicating vertically partitioned data is essential since it aids in assessing how well-federated learning algorithms function. The authors want to investigate approaches that make it easier to map various feature sets into a single feature space

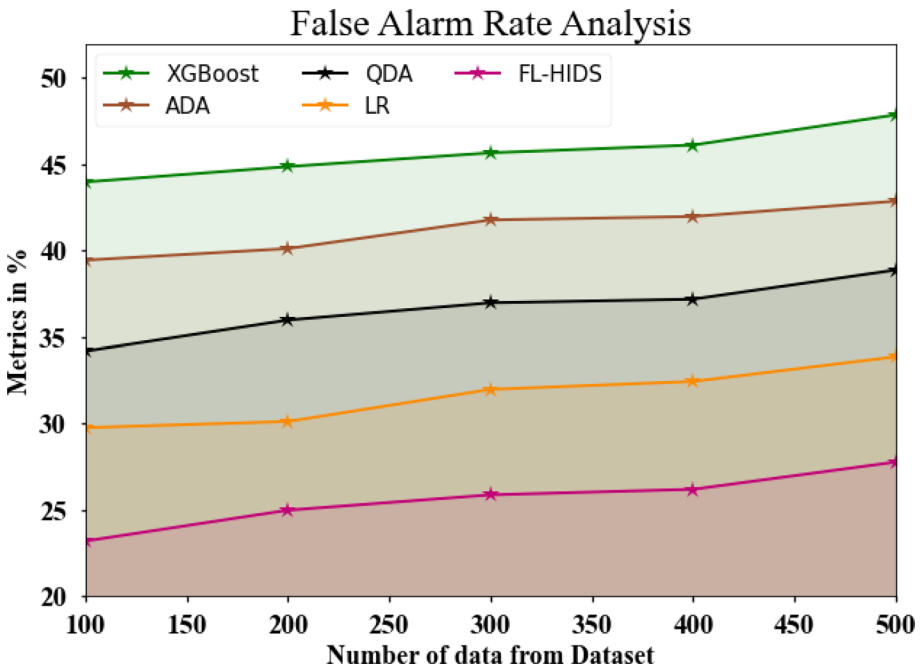

Fig. 9 False alarm rate analysis for FL-HIDS method

Table 9 False alarm rate analysis for FL-HIDS method

Number of data	XGBoost	ADA	QDA	LR	FL-HIDS
100	43.98	39.45	34.19	29.76	23.19
200	44.87	40.12	35.98	30.12	24.98
300	45.67	41.78	36.98	31.98	25.87
400	46.12	41.98	37.19	32.44	26.19
500	47.87	42.87	38.88	33.87	27.77

because of the tight relationship between this objective and another potential research direction.

Authors' Contributions AS, BD, and AKN, are responsible for designing the framework, analyzing the performance, validating the results, and writing the article. PSK, SS, and SC are responsible for collecting the information required for the framework, provision of software, critical review, and administering the process.

Funding The authors did not receive any funding.

Data Availability No datasets were generated or analyzed during the current study.

Code Availability Not applicable.

Declarations

Conflict of interest Authors do not have any conflicts.

References

1. Elrawy, M., Awad, A., & Hamed, H. (2018). Intrusion detection systems for IoT-based smart environments: A survey. *Journal of Cloud Computing, 7*, 21.
2. Baseline Security Recommendations for IoT. ENISA Report. https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iot. Accessed 15 Feb 2022.
3. Liu, X., Zhao, M., Li, S., Zhang, F., & Trappe, W. (2017). A security framework for the Internet of things in the future Internet architecture. *Future Internet, 9*, 27.
4. General Data Protection Regulation (GDPR). https://gdpr-info.eu/. Accessed 15 Feb 2022.
5. Giechaskiel, I., Zhang, Y., Rasmussen, K.B. (2019). A Framework for evaluating security in the presence of signal injection attacks. In *Proceedings of the European symposium on research in computer security, Luxembourg, 23–27 September 2019*.
6. Hajjaji, Y., Boulila, W., Farah, I. R., Romdhani, I., & Hussain, A. (2021). Big data and IoT based applications in smart environments: A systematic review. *Computer Science Review, 39*, 100318.
7. Zhao, B., Fan, K., Yang, K., Wang, Z., Li, H., & Yang, Y. (2021). Anonymous and privacy-preserving federated learning with industrial big data. *IEEE Transactions on Industrial Informatics, 17*(9), 6314–6323. https://doi.org/10.1109/TII.2021.3052183
8. Awaysheh, F. M., Aladwan, M. N., Alazab, M., Alawadi, S., Cabaleiro, J. C., & Pena, T. F. (2021). Security by design for big data frameworks over cloud computing. *IEEE Transactions on Engineering Management, 69*, 1–18.
9. Lv, Z., Qiao, L., Hossain, M. S., & Choi, B. J. (2021). Analysis of using blockchain to protect the privacy of drone big data. *IEEE Network, 35*(1), 44–49.
10. Yang, H., Zeng, R., Xu, G., & Zhang, L. (2021). A network security situation assessment method based on adversarial deep learning. *Applied Soft Computing, 102*, 107096.
11. Chakraborty, S., Krishna, R., Ding, Y., & Ray, B. (2022). Deep learning-based vulnerability detection: Are we there yet? *IEEE Transactions on Software Engineering, 48*(9), 3280–3296. https://doi.org/10.1109/TSE.2021.3087402
12. Khatri, A. H., Gadag, V., Singh, S., Satapathy, S. K., & Mishra, S. (2023). Quantum data traffic analysis for intrusion detection system. *Evolution and Applications of Quantum Computing*. https://doi.org/10.1002/9781119905172.ch8
13. Tang, Z., Haiyang, Hu., & Chonghuan, Xu. (2022). A federated learning method for network intrusion detection. *Concurrency and Computation: Practice and Experience, 34*(10), e6812.
14. Szczepanik, W., & Niemiec, M. (2022). Heuristic intrusion detection based on traffic flow statistical analysis. *Energies, 15*(11), 3951.
15. Jeune, L. L., Goedemé, T., & Mentens, N. (2021). Machine learning for misuse-based network intrusion detection: Overview, unified evaluation, and feature choice comparison framework. *IEEE (Institute of Electrical and Electronics Engineers) Access, 9*, 63995–64015.
16. Ring, M., Wunderlich, S., Scheuring, D., Landes, D., & Hotho, A. (2019). A survey of network-based intrusion detection data sets. *Computers and Security, 86*, 147–167. https://doi.org/10.1016/j.cose.2019.06.005
17. Ferrag, M. A., Shu, L., Djallel, H., & Choo, K.-K.R. (2021). Deep learning-based intrusion detection for distributed denial of service attack in Agriculture 4.0. *Electronics, 10*, 1257.
18. Aashmi, R. S., & Jaya, T. (2023). Intrusion detection using federated learning for computing. *Computer Systems Science and Engineering, 45*(2), 1295.
19. Liu, Z., & Shi, Y. (2022). A hybrid IDS using a GA-based feature selection method and random forest. *International Journal of Machine Learning and Computing, 12*(2), 43.
20. Halim, Z., Yousaf, M. N., Waqas, M., Sulaiman, M., Abbas, G., Hussain, M., et al. (2021). An effective genetic algorithm-based feature selection method for intrusion detection systems. *Computers and Security, 110*, 102448.
21. Panigrahi, R., & Borah, S. (2018). A detailed analysis of the CICIDS2017 dataset for designing intrusion detection systems. *International Journal of Engineering and Technology, 7*(3.24), 479–482.
22. Singh Panwar, S., Raiwani, Y. P., & Panwar, L. S. (2019). Evaluation of network intrusion detection with features selection and machine learning algorithms on CICIDS-2017 dataset. In *International*

conference on advances in engineering science management and technology-2019. Dehradun, India: Uttaranchal University.
23. Salo, F., Nassif, A. B., & Essex, A. (2019). Dimensionality reduction with IG-PCA an ensemble classifier for network intrusion detection. *Computer Networks, 148*, 164–175.
24. Sarhan, M., Layeghy, S., Portmann, M. (2021). An explainable machine learning-based network intrusion detection system for enabling generalisability in securing IoT networks. arXiv:2104.07183.
25. Portmann, M. (2021). Netfow datasets for machine learning-based network intrusion detection systems. In *Big data technologies and applications: 10th EAI international conference, BDTA 2020 and 13th EAI international conference on wireless internet, WiCON 2020, virtual event, December 11, 2020: Proceedings* (vol. 371, p. 117). Springer Nature.
26. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics* (pp. 1273–1282). PMLR.

A. Suresh received his Master's and Ph.D. in the fields of computer science and engineering from the University of Anna University, Chennai, India, in 2008 and 2018, respectively. He is currently working as an associate professor in the School of Computer Science and Engineering at VIT University, Vellore, India. He has more than 20 years of experience in various academic and industrial fields in computer science. His main research interests lie in the areas of networking, soft computing, and data science.

Dr. B. Dwarakanath M.Tech., PhD is Associate Professor in the Department of Information Technology at SRM Institute of Science and Technology, Ramapuram, Chennai, Tamil Nadu. He has completed his Ph.D, in Machine Learning - Information Technology in 2019 from Hindustan Institute of Technology and Science, Chennai, Tamilnadu. He has done his M.Tech, CSE from Vellore Institute of Technology, Vellore in the year 2004. Dr. B. Dwarakanath has 20 years of teaching experience and has 30 publications in International Journals and Conferences. His research interests include Machine Learning, Image Processing, and Network Programming. He is an active member of ACM, IET, ISTE, CSI, and IEANG.

Dr. Ashok Kumar Nanda is working as Professor in the Department of CSE at the B V Raju Institute of Technology (BVRIT), Narsapur, Medak, Greater Hyderabad, Telangana, India. BVRIT is recognized by AICTE and UGC-Autonomous, NBA and NAAC-A accredited. He was awarded a Ph. D. (CSE) degree from NIT Hamirpur, Himachal Pradesh in May 2015; an M. Tech. (CSE) from GJU, Hisar, Haryana in 1999; and graduated engineering in Electrical Engineering from The IE (India), Calcutta in 1996. He has been teaching for more than 22 years. He has published more than 20 international journals and 15 international conference papers, and a book chapter with a total of 14 patents (including international patents) and Indian copy rights. His main research interests include lightweight cryptography, Information Security, IoT Security, Machine Learning, and Deep Learning.

Dr. P. Santhosh Kumar Working as an Associate Professor in the Department of Information Technology at SRM Institue of Science and Technology, Ramapuram, Chennai. He graduated in Computer Science and Engineering at Anna University, Chennai, Tamilnadu, India. He secured Master of Engineering in Computer Science and Engineering at Anna University, Chennai, India. He received his Ph.D. in the field of Cloud Computing at Sathyabama Institute of Science and Technology, Chennai, India. He is in teaching profession for more than 12 years. He has presented number of papers in National and International Journals, Conference and Symposiums. He is holding 4 International and 2 National patents as well. Moreover he is a member of ISTE, IAENG and ACM Professional Society Bodies. His main area of interest includes Cloud computing, Machine Learning, Network Security, Artificial Intelligence and Internet of Things.

Dr. S. Sankar received his PhD degree Ph.D. in Information Technology from Hindustan University (2018), an M.E. in Software Engineering from Periyar Maniammai College of Technology (2006), and a B.E. in Computer Science and Engineering from Arulmigu Kalasalingam College of Engineering (1992). Currently, a Professor at Saveetha Institute of Medical and Technical Sciences in Chennai, India, His expertise lies in Machine Learning, Image Processing, and Information Science. With over 20 years of teaching experience, he is a dedicated educator, passionate programmer, and active researcher who has supervised numerous theses, published articles, obtained patents, organized technical events, authored technical books, and contributed to various book chapters.

Dr. Sreevardhan Cheerla is currently working as an Associate Professor in the Department of ECE at KL University (KLEF), Andhra Pradesh, INDIA. He published a number of papers in preferred Journals and chapters in books. He also presented various academic and research-based papers at national and international conferences. His areas of specialization include Wireless Communications, Microstrip Patch Antennas, Bio-Medical Instrumentation, and IoT.

Authors and Affiliations

A. Suresh[1] · B. Dwarakanath[2] · Ashok Kumar Nanda[3] · P. Santhosh Kumar[2] · S. Sankar[4] · Sreevardhan Cheerla[5]

✉ P. Santhosh Kumar
santhosp3@srmist.edu.in; psanthosh_kumar45@outlook.com

A. Suresh
a.suresh@vit.ac.in

B. Dwarakanath
dwarakab@srmist.edu.in

Ashok Kumar Nanda
ashokkumarnanda@yahoo.com

S. Sankar
sankars.sse@saveetha.com

Sreevardhan Cheerla
sreevardhancheerla@kluniversity.in

[1] School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India

[2] Department of Information and Technology, Faculty of Engineering and Technology, SRM Institute of Science and Technology, Ramapuram, Chennai, India

[3] Department of Computer Science and Engineering, B V Raju Institute of Technology, Hyderabad, Telangana, India

[4] Department of Computer Science and Engineering, Saveetha School of Engineering, SIMATS, Chennai, India

[5] Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India