Adaptive Intrusion Detection in Heterogeneous IoT Environments Using Federated Graph Neural Networks

R.V. Umaselvi[1] and T. R. Nisha Dayana[2]

[1, 2]Department of Computer Science
Vels Institute of Science and Technology and Advanced Studies
Chennai, Tamil Nadu, India
[1]umaselviravichandran120873@gmail.com, [2]nisha.dayana1984@gmail.com

Abstract. In the fast-growing Internet of Things (IoT) environment, intrusion detection is an essential security problem due to the heterogeneity and decentralized characteristic of connected devices. This research introduces a new intrusion detection system (IDS) based on the synergetic combination of Federated Learning (FL) and Graph Neural Networks (GNNs) to solve privacy, accuracy, and resource limitations in IoT environments. The envisioned system allows every IoT device to locally train light-weight GNN models on graph-structured representations of its local network traffic. The local models are then collectively aggregated using FL methods without exposing raw data, thus maintaining privacy. Sophisticated graph construction and temporal encoding techniques are utilized to capture changing attack patterns, and continual learning mechanisms are employed to provide adaptability to novel threats. Performance is measured in terms of detection accuracy, latency, and energy usage. Compared with traditional signature-based (Snort), anomaly-based (K-Means), and deep learning-based (CNN) IDS schemes, the developed GNN+FL mechanism exhibits higher detection accuracy (96%), low latency (75ms), and energy efficiency (1W) and is extremely well-suited for low-power, real-time applications. Apart from enhancing real-time detection, this framework also supports scalable and privacy-aware security in distributed IoT environments..

Keywords:. ： Intrusion Detection Systems, Internet of Things, Federated Learning, Graph Neural Networks

1 Introduction

Network security is still a major concern in the world of information technology, and conventional security solutions have proven to be inadequate due to the quick growth of the technology and the growing complexity of network threats. A significant challenge nowadays is the effective identification and mitigation of cyberattacks, particularly in the developing field of federated learning (FL) Network intrusions can be found with the use of Intrusion Detection Systems (IDS)[1]. It is able to differentiate between the host data and the traits of intrusion behaviour and intrusion operations.

But in real-world situations, an organization's or enterprise's network system only produces lower-quality attack sample data for training.

Effective defences against the growing cybersecurity threats in the Artificial Intelligence of Things (AIoT) space have been demonstrated by the integration of intrusion detection systems (IDSs) with machine learning (ML) techniques. Federated learning (FL) has emerged as a potential approach for AIoT intrusion detection due to privacy issues[2].

Conventional IDS techniques are incapacitated in distributed IoT settings by their dependency on centralized data collection, which is privacy-intrusive and introduces latency. Furthermore, IoT devices are typically resource-constrained, and therefore it is difficult to use cumbersome deep learning models or real-time monitoring. Therefore, there is a requirement for an IDS that is scalable, light, and privacy-preserving but can learn collectively across devices.

Federated learning (FL) is a distributed machine learning model that allows IoT devices to cooperatively build a common ML model without sharing local data. As a result, FL in IDSs contributes to reducing the processing load on central processing servers while maintaining data privacy. There are three popular FL architectures: semi-decentralized, decentralized, and centralized FL (client-server). Each device in the centralized FL architecture trains the local model using its own data, and the parameters and weights of each device are then combined to create a global model that is managed by a central entity, such as a coordinator or server[3].

This work suggests a federated learning-based hybrid intrusion detection system with Graph Neural Networks (GNNs) to deal with these issues. Each device learns a local GNN-based model on its traffic data and exchanges only updates of the model with a central server, maintaining privacy with collaborative learning. The work presents a new method of modeling network traffic as dynamic graphs, reflecting both structural and temporal interactions. Lightweight optimization is utilized to minimize energy usage and computational expense, allowing the system to be deployed in real-time on IoT devices. The system also allows for continuous learning to learn from new threats. The objectives of the study are as follows

- To create a decentralised, privacy-preserving intrusion detection system that effectively detects intrusions in Internet of Things environments by leveraging Graph Neural Networks (GNNs) and Federated Learning (FL).
- To use lightweight model optimization approaches, the system will be scalable and energy efficient, making it appropriate for deployment on IoT devices with limited resources.

The rest of the paper is structured as follows: Section 2 discusses the related works and Section 3 describes local GNN training and federated learning aggregation; Section 4 provides performance analysis and comparisons with current IDS methods; Section 5 concludes the paper with results and sets directions for future work.

2 Related Works

Osa et al., (2024) developed a Deep Neural Network model for intrusion detection on the CICIDS 2017 dataset and addressed some data imbalance issues with SMOTE and Random Sampling. The study was performed in Google Colaboratory, and it achieved a model accuracy of 99.68 and a loss of 0.0102. The results achieved demonstrated the performance was outstanding in detecting zero-day cyber-attacks as well as known attacks[4].

Mao et al., (2025) suggested the label-aware federated graph contrastive learning framework FeCoGraph for few-shot intrusion detection. In order to directly process flow embeddings that work with a variety of GNNs, the line graph is presented. Additionally, in order to efficiently utilize label information, we develop a graph contrastive learning task that permits intra-class embeddings to be more compact than inter-class embeddings. Researchers use FL to increase NIDS's scalability and defend data privacy by covering more attack scenarios[5].

Alsaleh et al., (2024) provide a thorough analysis of FL for IDSs in an Internet of Things setting with devices that have limited resources. We look into the previous research on FL in three different architectures: semi-decentralized, decentralized, and centralized (client-server). The results of the study show that the FL framework needs to be improved in order to better fit IoT networks. This improvement is essential, especially for tackling two major issues: having a design aggregation algorithm that can efficiently manage the heterogeneity and resource limitations present in IoT devices, and the requirement to lightweight FL client models to account for the resource constraints of IoT devices[6].

Jianping et al., (2024) suggests a GNN with an attentional foundation for identifying cross-level and cross-departmental network attacks. This technique protects data privacy on dispersed devices while facilitating collaborative model training. The accuracy of network attack detection is improved by creating a graph structure based on log density and arranging network traffic data chronologically. Results from experiments show that the approach prioritizes data security and privacy while achieving accuracy and robustness that are comparable to those of conventional detection techniques[7].

Arya et al., (2023) outlines a method for detecting intrusions in smart cities that uses distributed FL of heterogeneous neural networks. It uses the most effective intrusion detection method, saving time and money. First, cars create local, deep learning-based IDS classifiers for VANET data streams using a FL technique[8].

Kaur et al., (2024) suggest a network intrusion detection model based on FL for IIoT applications that only exchange learnt parameters with the central server and save data in its original format on local servers. During FL training, gated recurrent units (GRUs) assist the suggested model in learning the temporal relationships of network traffic attacks, which improves the accuracy of intrusion detection[9].

SAjid et al., (2024) describes a hybrid intrusion detection model that integrates XGBoost, CNN, and LSTM to enhance detection accuracy on four existing benchmark datasets. This model, through superior feature selection along with deep learning classification, demonstrates high detection rates with low false acceptance rates for identifying multiple cyber threats[10]. Table 1 lists the related works pertaining to the present study.

Table 1 Related works – IDS in IoT

Author(s) & Year	Method	Purpose	Strengths	Limitations
Mao et al., (2025)	FeCoGraph – label-aware federated graph contrastive learning	Few-shot intrusion detection using line graphs and GNN-compatible flow embeddings	Utilizes label info efficiently; scalable; improves intra-class compactness	Few-shot learning still poses challenges in real-world large-scale deployment
Alsaleh et al., (2024)	Review of FL architectures (centralized, decentralized, semi-decentralized)	Analyze FL in IoT-focused IDS under resource limitations	Identifies key architectural challenges; highlights need for lightweight models	Lacks a concrete solution; more of a theoretical/framework analysis
Jianping et al., (2024)	GNN with attention and log density-based graph structuring	Detect cross-departmental/level attacks while preserving privacy	Maintains data privacy; achieves robustness and accuracy comparable to traditional methods	Performance may depend on quality of log data and chronological ordering
Arya et al., (2023)	FL of heterogeneous neural networks in VANETs	Distributed intrusion detection for smart city environments	Reduces costs; supports real-time VANET stream processing	Heterogeneity of models may impact consistency of global model updates
Kaur et al., (2024)	FL using GRUs for temporal	Improve IIoT NIDS by capturing temporal	Accurate detection via tem-	GRU models may still be computationally heavy for ex-

	pattern detection	network traffic patterns	poral learning; preserves data locality	tremely constrained devices

The studied research globally goes in favor of the increasing adoption of federated learning (FL) as a cornerstone to architect scalable and privacy-constraining network intrusion detection systems (NIDS) for IoT and IIoT use cases. Different methodologies such as unsupervised clustering, label-aware graph contrastive learning, and attention-based graph neural networks (GNNs) suggest how FL can efficiently be framed to tackle heterogeneity, data segregation, and sparsity scenarios. In addition, the use of temporal models such as gated recurrent units (GRUs) and distributed learning in vehicular networks mark efforts to maximize accuracy and efficiency in terms of resources in real-time and mobile settings. Nevertheless, despite these advances, the research collectively acknowledges model heterogeneity, computational resource constraint, and the need for advanced aggregation algorithms. In total, the direction of research leans towards more special, light-weight, and privacy-aware NIDS models for various and resource-restricted environments

3 Methodology

The method combines Federated Learning (FL) and Graph Neural Networks (GNNs) to detect intrusions across a distributed IoT ecosystem without centralized data sharing. Each node (e.g., a smart home device or sensor) trains a local GNN-based anomaly detector using its own network traffic and shares only the learned model parameters (not the data) with a central server. The workflow for the proposed systems is shown in figure 1.

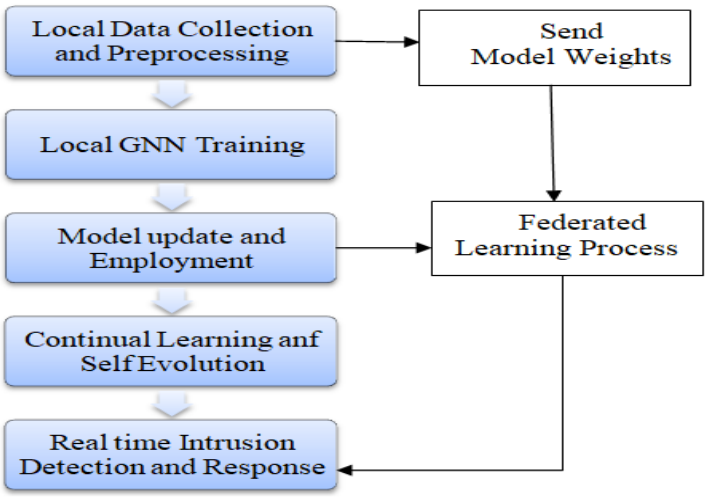

Figure 1 Workflow of the proposed system

3.1 Data Preprocessing and Graph Construction

In the proposed system, data preprocessing and graph construction form the foundation for effective intrusion detection. Each IoT device independently gathers raw network traffic data, such as packet flows, communication logs, and metadata. This data is then preprocessed to remove noise and standardized for uniform representation.The purified data are converted into graph representations, wherein nodes are entities that communicate with each other (e.g., devices or services) and edges represent communication sessions between them.Each edge is annotated with context information including protocol type, session length, and packet size, and node and edge features are learnt to reveal behavioral traits and interaction tendencies. To support temporal learning and capture changing threats, the system utilizes a sliding time window to create dynamic graphs showing evolution of network behavior across time. Graph-based representation enables more expressive modelling of intricate IoT interactions as input to downstream GNN-based analysis.

3.2 Local Graph Neural Network (GNN) Training

During the local training stage, the individual IoT devices train a light-weight Graph Neural Network (GNN)-based classifier on its graph-structured traffic data individually. The GNN structure used could be Graph Convolutional Networks (GCNs) or Graph Attention Networks (GATs) due to their capability to efficiently capture the structural and relational information embedded in network traffic. To encode temporal dynamics and evolving communication patterns, temporal encoding mechanisms are integrated into the model. The classifier is trained to perform binary or multi-class classification tasks. The classifier distinguishes normal from malicious behavior such as Denial of Service (DoS), probing, or other attack behaviors. Training employs a cross-entropy loss function in conjunction with regularization techniques to provide generalization and stability, especially in the scenario where the limited and heterogeneous dataset available on resource-constrained IoT devices is utilized. Decentralized learning methodology provides each device the capability of intrusion detection autonomously while retaining local context[11].

3.3 Federated Learning and Aggregation

After local training within a given set of epochs, all IoT devices engage in rounds of federated learning to improve the global model together. Instead of sharing data, devices only upload trained model weights to a central aggregator, thereby ensuring data security and privacy.The aggregator subsequently applies model merging through Federated Averaging (FedAvg) or more flexible algorithms such as FedProx, which are more appropriate for handling non-IID (non-independent and identically distributed) data that is normally encountered in heterogeneous IoT environments. After aggregation, the globally updated model is returned to all devices that are involved. This cyclical mechanism allows for continuous learning and updating of the network intru-

sion detection model, without sacrificing the privacy and autonomy of specific IoT devices[12].

3.4. Continual Learning & Self-Evolution

In order to effectively adapt to new and unexpected threats, the system contains a mechanism of continuous learning and self-enhancement embedded in each IoT device. A confidence scoring module evaluates the confidence of each prediction, supplemented by peer agreement metrics where multiple devices are observing similar trends. High-confidence unlabeled data get automatically labeled via a consensus voting method and are subsequently injected in the subsequent local training cycle to enable the model to learn emerging patterns without being manually intervened with. In addition, the system actively looks for outliers or outlier traffic pattern deviances. On detection of these, local fine-tuning of the GNN model is started in order to ensure responsiveness to new attack methods. This continuous, adaptive learning cycle allows the intrusion detection system to remain effective in the long term, including against sophisticated and dynamic threats[13].

3.4 Lightweight Optimization

The system utilizes a variety of light-weight optimization methods to limit computational and memory overhead to allow efficient operation on low-resource IoT devices. Model quantization is used to convert trained GNN models into lower-precision representations, e.g., 8-bit, that reduce storage requirements and inference latency by a significant amount without compromising accuracy.Pruning methods are used to remove redundant neurons and graph edges, further simplifying the model and lowering computational requirements. Additionally, attention mechanisms are integrated into the GNN architecture to rank nodes and edges dynamically based on importance, allowing the model to focus on the most significant parts of the network graph while filtering out less significant information. These optimizations combined enable the deployment of effective intrusion detection capabilities even on low-power IoT devices, hence the system is scalable and deployable in real-world scenarios[14].

3.5 Intrusion Detection and Response

Intrusion Detection and Response (IDR) systems are central to network infrastructure protection through real-time monitoring of live traffic across devices. Each device in the network is coded to classify traffic via a Graph Neural Network (GNN), which intelligently looks out for patterns to identify potential threats or anomalies. After detecting anomalies in the traffic pattern, the system can automatically alert or respond automatically, such as by quarantining the affected node or diverting traffic to secure paths. Active response enables network security through immediate responses to potential intrusions, mitigating the impact of real-time cyber attacks[15].

4. Results and Findings

The system utilizes a variety of light-weight optimization strategies to limit computation and memory overhead in a bid to facilitate effective operation on low-resource IoT devices. Data sets like NSL-KDD are well-suited for such application since they support a high degree of network traffic diversity, both normal traffic as well as various attacks like DoS (Denial of Service), probing, and user-to-root attacks With these datasets, you can make sure that the IDS models are tested under a full range of traffic patterns and attack types so that their performance can be well evaluated. Moreover, testing each IDS under various configurations is also necessary to mimic various network environments and attack levels.This can involve different levels of traffic, attack rates, and data types (e.g., enterprise networks versus IoT devices). Running IDS on multiple test cases will determine their robustness, capacity to generalize to unknown attacks, and performance in dealing with different traffic patterns. This will guarantee that the IDS solutions not only perform well but are also scalable and flexible to different real-world network scenarios.

Metric	GNN + FL (Proposed System)	Snort (Traditional Signature-Based IDS)	K-means (Anomaly-Based IDS)	CNN (Deep Learning-Based IDS)
Detection Accuracy	96%	85%	90	94%
Latency	75 ms	350ms	200ms	600ms
Energy Consumption	1W	3W	3W	7W

Three key factors Detection Accuracy, Latency, and Energy Consumption are employed in Table 2 to quantify the performance of proposed systems against three varied Intrusion Detection Systems (IDS).

At 96% detection accuracy, the GNN + FL (Proposed System) performs the most accurate detection with superior detection capability for identifying known and unknown attack patterns. This is truly a huge strength, particularly within environments where there are probably likely to be shifting and dynamic threats. One of the limitations of signature-based techniques that are dependent upon pre-known attack signatures is that Snort can actually fight known attacks single-handedly (85%), but not new or unexpected threats. Its 90% K-means accuracy (Anomaly-Based IDS) indicates how well it can find exceptions from normal behaviors. It will generate false alarms, of course, but how well or how badly it does can be determined by whether the training data is representative of typical assault behavior. The 94% accuracy value of CNN (Deep Learning-Based IDS) showcases the capability of the CNN-based model to discover complex patterns of network traffic. It could, however, be chal-

lenged when it comes to new types of attacks that have limited representation within the training data and sometimes overfit the training data.

The proposed system, GNN + FL, has a quite low latency of 75 ms and is appropriate for real-time intrusion detection in resource-limited environments, such as Internet of Things networks. Reducing the attack impact requires timely detection. While Snort (Traditional Signature-Based IDS) is able to detect known threats, signature matching will take longer due to a relatively high 350 ms latency that could be harmful in circumstances where time is scarce. K-means (Anomaly-Based IDS) response time is faster than Snort but slower than the GNN+FL system. When deciding whether or not network traffic features are out of the ordinary, time is consumed in comparison and grouping. The computational load of deep learning models is justified by the very high CNN (Deep Learning-Based IDS) latency of 600 ms. CNNs are slower in infer speed and therefore less suitable for such systems as Internet of Things devices, which demand immediate response.

GNN + FL, is the most energy-efficient at 1W and optimal for low-power, resource-constrained devices like Internet of Things systems. With the sustainability of devices with short battery lives in mind, doing so is keeping that in mind.Having a relatively moderate 3W energy usage of Snort (Traditional Signature-Based IDS) means that Snort needs more processing power, particularly because it uses centralized processing and updates its signatures regularly.Since anomaly-based solutions such as K-means also need high processing capabilities for clustering and comparing real-time traffic data, their 3W energy usage is similar to that of Snort. Their peak energy consumption is 7W for CNN (Deep Learning-Based IDS), reflecting the large computational burdens of deep learning models, which are usually more processor-intensive to train and infer.

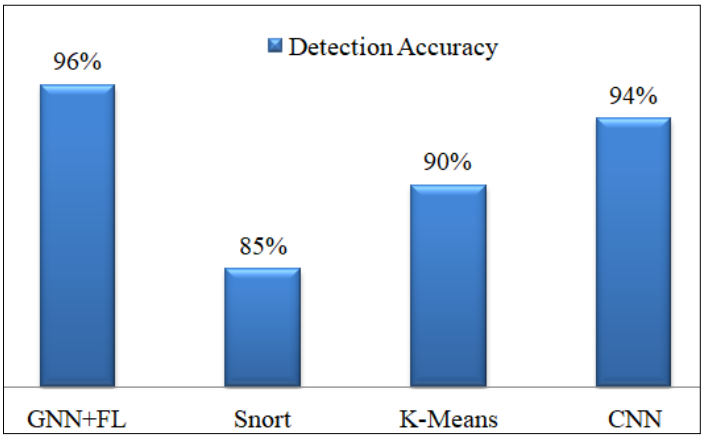

Figure 2 Proposed System-Detection Accuracy

The accuracy of the proposed system in identifying IoT intrusion is compared in Figure 2. With the 96% of detection accuracy , the GNN + FL is more precise than CNN (94%), K-Means (90%), and Snort (85%).This confirms that the suggested GNN + FL

method is better at correctly detecting intrusions compared to conventional, anomaly-based, and deep learning methods.

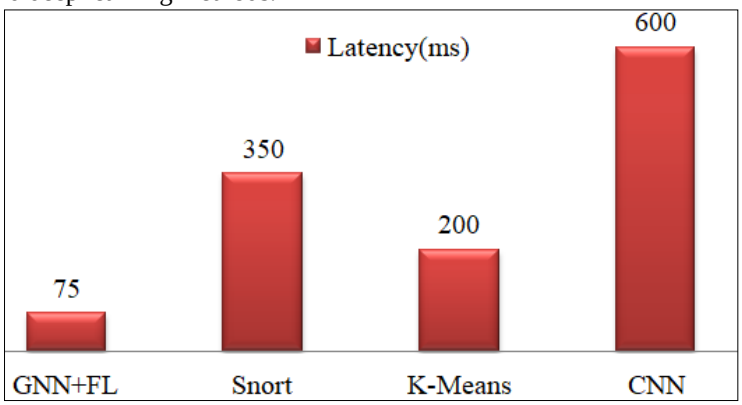

Figure 3 Proposed System- Latency

The proposed system's latency, or the time taken to respond, is compared in milliseconds in Figure 3. With 75 ms latency, GNN + FL is the quickest in real-time intrusion detection. CNN, however, less suits time-critical environments given that its highest latency (600 ms) is followed by Snort (350 ms) and K-Means (200 ms).

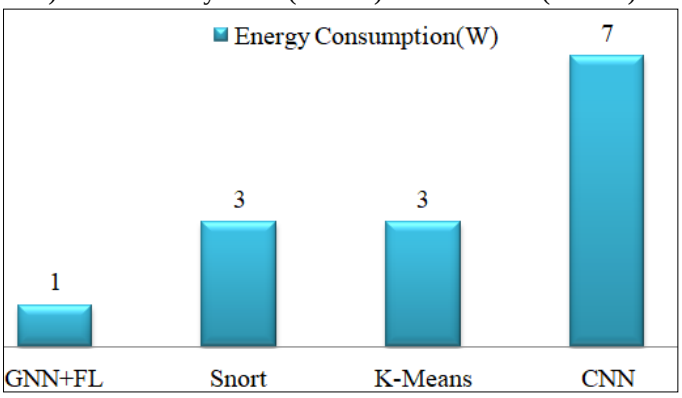

Figure 4 Proposed System- Energy Consumption

The energy usage of the proposed method, in watts (W), is shown in Figure 4.The GNN + FL system consumes the least amount of energy and is ideal for low-power Internet of Things devices. CNN consumes the least at 7W and is thus less suitable for low-resource environments compared to Snort and K-Means, which draw 3W each.

GNN + FL system achieves the optimal tradeoff among high detection accuracy, low latency, and low energy, which makes it most appropriate for real-time, resource-scarce settings like IoT networks. It is most suited to adaptive, changing attack scenarios in which other techniques will have trouble.The Snort is effective against known attacks but less so because its greater latency and medium accuracy do not make it best for novel attack detection in real time. Its energy usage is also greater than the suggested system. K-means provides good anomaly detection with average latency and energy usage but can be challenged with false positives and accuracy in comparison to more dedicated systems such as GNN+FL or CNN. CNN offers very

good accuracy (94%) but with the price of high latency and energy consumption. This makes it less practical for deployment on low-power devices or real-time environments where timely responses are required.

5. Conclusion

This study presents a privacy-preserving and energy-efficient intrusion detection framework that combines Graph Neural Networks with Federated Learning (GNN+FL) for securing distributed IoT infrastructures. Through local training and federated aggregation, the system gains high detection accuracy with preserved user data privacy. With graph-based traffic modeling and dynamic graph updating, fine-grained network behavior analysis can be performed to detect even subtle or evolving attack patterns. Lightweight optimization techniques such as pruning and quantization further improve the feasibility of the system on resource-limited devices. Compared to traditional IDS methods, the system described has a higher detection rate (96%), significantly reduced latency (75ms), and lower energy consumption (1W) and is thus ideal for today's IoT applications that need real-time responsiveness and low overhead. For future research, the framework would be expanded to enable cross-domain threat intelligence exchange via secure multi-party computation. Additionally, inclusion of reinforcement learning would make the adaptive response methods more effective for zero-day attack responses. Extending federated personalization where the global model learns from a user's behavior on individual devices while still having generalizability will also receive attention through research. Lastly, real-world experimentation and deployment within heterogeneous IoT deployments will be made to verify system robustness as well as scalability under real-world scenarios.

References

1. Farhana, Kaniz, Maqsudur Rahman, and Md Tofael Ahmed. "An intrusion detection system for packet and flow based networks using deep neural network approach." *International Journal of Electrical & Computer Engineering (2088-8708)* 10, no. 5 (2020).
2. Han, Weixiang, Jialiang Peng, Jiahua Yu, Jiawen Kang, Jiaxun Lu, and Dusit Niyato. "Heterogeneous data-aware federated learning for intrusion detection systems via meta-sampling in artificial intelligence of things." IEEE Internet of Things Journal 11, no. 8 (2023): 13340-13354.
3. E. M. Campos, P. F. Saura, A. González-Vidal, J. L. Hernández-Ramos, J. B. Bernabé, G. Baldini, et al., "Evaluating federated learning for intrusion detection in Internet of Things: Review and challenges", *Comput. Netw.*, vol. 203, Feb. 2022.
4. Osa, Edosa, Patience E. Orukpe, and Usiholo Iruansi. "Design and implementation of a deep neural network approach for intrusion detection systems." *e-Prime-Advances in Electrical Engineering, Electronics and Energy* 7 (2024): 100434.
5. Mao, Qinghua, Xi Lin, Wenchao Xu, Yuxin Qi, Xiu Su, Gaolei Li, and Jianhua Li. "FeCoGraph: Label-aware Federated Graph Contrastive Learning for Few-shot Network Intrusion Detection." IEEE Transactions on Information Forensics and Security (2025).

6. S. S. Alsaleh, M. El Bachir Menai and S. Al-Ahmadi, "Federated Learning-Based Model to Lightweight IDSs for Heterogeneous IoT Networks: State-of-the-Art, Challenges, and Future Directions," in *IEEE Access*, vol. 12, pp. 134256-134272, 2024, doi: 10.1109/ACCESS.2024.3460468.
7. Jianping, Wu, Qiu Guangqiu, Wu Chunming, Jiang Weiwei, and Jin Jiahe. "Federated learning for network attack detection using attention-based graph neural networks." Scientific Reports 14, no. 1 (2024): 19088.
8. Arya, Monika, Hanumat Sastry, Bhupesh Kumar Dewangan, Mohammad Khalid Imam Rahmani, Surbhi Bhatia, Abdul Wahab Muzaffar, and Mariyam Aysha Bivi. 2023. "Intruder Detection in VANET Data Streams Using Federated Learning for Smart City Environments" *Electronics* 12, no. 4: 894. https://doi.org/10.3390/electronics12040894
9. Kaur, Amandeep. "Intrusion Detection Approach for Industrial Internet of Things Traffic using Deep Recurrent Reinforcement Learning Assisted Federated Learning." IEEE Transactions on Artificial Intelligence (2024).
10. Sajid, Muhammad, Kaleem Razzaq Malik, Ahmad Almogren, Tauqeer Safdar Malik, Ali Haider Khan, Jawad Tanveer, and Ateeq Ur Rehman. "Enhancing intrusion detection: a hybrid machine and deep learning approach." *Journal of Cloud Computing* 13, no. 1 (2024): 123.
11. Zhong, Meihui, Mingwei Lin, Chao Zhang, and Zeshui Xu. "A survey on graph neural networks for intrusion detection systems: methods, trends and challenges." Computers & Security (2024): 103821.
12. Agrawal, Shaashwat, Sagnik Sarkar, Ons Aouedi, Gokul Yenduri, Kandaraj Piamrat, Mamoun Alazab, Sweta Bhattacharya, Praveen Kumar Reddy Maddikunta, and Thippa Reddy Gadekallu. "Federated learning for intrusion detection system: Concepts, challenges and future directions." Computer Communications 195 (2022): 346-361.
13. Prasath, Sai, Kamalakanta Sethi, Dinesh Mohanty, Padmalochan Bera, and Subhransu Ranjan Samantaray. "Analysis of continual learning models for intrusion detection system." IEEE Access 10 (2022): 121444-121464.
14. Pan, Jeng-Shyang, Fang Fan, Shu-Chuan Chu, Hui-Qi Zhao, and Gao-Yuan Liu. "A lightweight intelligent intrusion detection model for wireless sensor networks." Security and communication Networks 2021, no. 1 (2021): 5540895.
15. Aravamudhan, Parthiban. "A novel adaptive network intrusion detection system for internet of things." Plos one 18, no. 4 (2023): e0283725.