International Journal of Informatics Engineering and Computing (IJIMATIC)
Vol. 1, No. 1, 2024

# IoT Intrusion Detection Model Using RNN Algorithm

Pierre Lacks[1]
[1] Université Paris Nanterre, France

**Abstract**
An Intrusion Detection System (IDS) is the method processed in an IoT system's network layer. Many techniques have been applied in the IDS and a higher performance in identifying IDS. The existing method involved in IDS tends to be ineffective due to the drawbacks of big data, centralization, and low privacy. The RNN model achieves the best accuracy (99.5%) and highest performance across metrics but has the highest variability (STD = 0.14). If stability is a priority, FPA-DT and FPA-RF would be the best choices, offering strong performance with lower variance. FPA optimization consistently improves traditional models, making it an effective enhancement strategy.

**Keywords:**
IoT, Intrusion Detection System, Machine Learning, RNN



## 1. Introduction

Intrusion Detection Systems (IDS) in IoT face significant challenges when relying on traditional methods due to the unique characteristics of IoT environments. Traditional IDS techniques, such as signature-based and anomaly-based detection, struggle with the diverse and resource-constrained nature of IoT devices. Signature-based IDS requires frequent updates to recognize new attack patterns, making it ineffective against zero-day attacks. Meanwhile, anomaly-based detection relies on predefined behavioral baselines, which are difficult to establish in highly dynamic IoT networks with heterogeneous devices. Additionally, the sheer volume of data generated by IoT devices creates scalability issues, as conventional IDS solutions may not efficiently process real-time traffic and identify threats with high accuracy [1][2].

Moreover, traditional IDS methods often suffer from high false positive and false negative rates, reducing their reliability in IoT security. The limited computational power and storage capabilities of many IoT devices make it impractical to deploy complex IDS algorithms, forcing reliance on centralized security solutions that introduce latency and potential single points of failure. Furthermore, IoT networks are highly distributed, increasing the attack surface and making it difficult to monitor traffic effectively using traditional IDS techniques. As cyber threats continue to evolve, traditional IDS approaches struggle to keep pace, highlighting the need for advanced, adaptive, and machine learning-driven solutions to enhance security in IoT environments [3][13].

Developing an IDS can combine dimension reduction techniques with Multi-class SVM to reduce dimensions and shorten training time. However, the ranking restriction issue is represented in the discriminant vector, which prevents the obtained information from being classified as harmful. To overcome the problems that occur in the existing model, it is proposed to describe a hybrid GWO-PSO, which shows the effective classification of attacks to binary and multi-class based on the NSL-KDD dataset. The paper presents contextual intrusion detection that uses runtime mode development for service interaction and functionality patterns. Analysis of the developed intrusion detection system shows

**Corresponding Author:** Pierre Lacks (pierre.lacks@gmail.com)

strange behavior in the network to detect intrusions. The contextual intrusion detection system was evaluated by generating several attacks in the BACnet protocol, and the results showed that the developed method had high performance in detecting attacks [6].

The IoT detection problem using machine learning involves identifying and classifying IoT devices within a network, often to enhance security, optimize performance, or enable automation. IoT devices, such as smart home appliances, wearables, and industrial sensors, generate vast amounts of data, but their diverse communication protocols, limited computational resources, and heterogeneous nature make detection challenging. ML models are employed to analyze network traffic patterns, device behavior, and communication signatures to distinguish IoT devices from non-IoT devices or to identify specific device types. For instance, supervised learning techniques like decision trees, support vector machines (SVM), and deep learning models can be trained on labeled datasets containing network traffic features (e.g., packet size, frequency, and protocol) to classify devices accurately. However, challenges such as imbalanced datasets, evolving device firmware, and adversarial attacks can hinder model performance, necessitating robust preprocessing, feature engineering, and continuous model updating [21].

Recent research has explored advanced deep learning models like CNN and RNN have been applied to raw network traffic data to automatically extract relevant features, reducing the need for manual feature engineering. Unsupervised learning methods, such as clustering and anomaly detection, are also used to identify unknown or rogue IoT devices by detecting deviations from normal behavior patterns. Additionally, federated learning has been proposed to train models across distributed IoT networks while preserving data privacy. Despite these advancements, the dynamic nature of IoT ecosystems and the increasing sophistication of cyber threats require ongoing innovation in ML-based detection methods. References to studies IoT on deep learning for IoT security highlight the importance of scalable and adaptive solutions in this domain [22][23].

A paper investigates critical IoT healthcare security issues, for which a new security framework based on RNNs was used to investigate enhancements in threat detection and response. This approach modeled network traffic and device behavior sequentially for anomaly and potential breach detection using RNNs. Hence, we introduce the RNN-based model combined with an inclusive security architecture, including data encryption, mechanisms of authentication, and monitoring tools in real-time. Experimental results prove that our RNN-based framework significantly improves malicious activity detection and reduces false positives compared to traditional security solutions. The proposed model would provide a strong, scalable, and adaptable security solution tailored to the IoT healthcare environment dynamics. These findings could indicate how RNNs can enhance security in IoTs and provide new ways in which better and more secure healthcare systems can be developed [7][8][12].

In this study, we explore the capabilities of RNNs to capture temporal dependencies and sequential patterns in the data that are particularly well-suited for processing sequential data, such as network traffic or time-series data generated by IoT devices. RNNs are designed to capture dependencies and patterns over time, making them highly effective for analyzing data where the sequence of events carries significant information. This capability enables the model to identify intricate and subtle patterns that might be overlooked or difficult to capture through manual feature engineering. This automated feature extraction not only reduces the reliance on domain-specific knowledge but also enhances the model's ability to generalize across different datasets and scenarios, thereby improving the overall accuracy and robustness of anomaly detection in IoT healthcare systems.

## 2. Related Works

A paper classified IDS into two approaches, anomaly detection and malicious traffic signatures [4]. Traditional techniques implemented data display techniques to detect attacks using multi-view attack information. The authors proposed a semi-supervised co-training method using the multi-display attack trait. The attack behavior will be preserved across multiple views, and attack detection will be performed using predictions made by the ML model from multiple attack views [8]. Another paper utilized techniques to develop a Collaborative Blockchain-Based Detection of DDoS Attacks Based on IoT Botnets. The developed model overcame the problem above from the current plans by installing lightweight agents at distinct multiple IoT installations. However, the developed model failed to prove the solution's applicability and stressed-out limitations that may emerge [5][7].

Machine Learning is a popular architecture to address detection issues including IDS detection [10][16][17][18]. A paper proposed an IDS detection using a two-level dimensional reduction engine and a two-level classification engine. The dimension reduction engine consists of component analysis and linear discrimination analysis units. In contrast, the classification engine consists of Naïve Bayes and the certainty factor version of the cascade K-nearest neighbor (CF-KNN) unit. The Naive Bayes classifier is used to classify attack records which, in turn, is refined by the CF-KNN classifier as a second filtering layer. Using the NSLKDD dataset [9], the proposed model achieves competitive detection performance for elusive attacks, namely U2R and R2L classes [6].

The current paper applies deep learning as an effective solution to this problem to enable the preprocessing phase, which can significantly affect the accuracy of an algorithm [9][12]. A paper employed a two-stage deep learning technique including a stacked auto-encoder and a soft-max classifier. The first stage detects normal or abnormal packets, and in the second stage, the classification method between standard and other attack classes. The authors report good classification accuracy on the KDD99 and UNSW-NB15 datasets [8]. The paper proposed a semi-supervised IDS with a generative model using the NSL-KDD dataset. The first module consists of an encoder-decoder network, while the second module consists of a fully connected neural network followed by a SoftMax classifier [3]. On the other hand, the testing phase uses a trained neural network generated from the training phase where KDDTest+ with detection accuracy of 91.39%[11]. Another paper also presents a deep learning-based intrusion detection system 3 for SDN-based IoT architecture, where SDN modeling is used to improve IoT security, scalability, and resilience. In contrast, the Restricted Boltzmann Machine (RBM) is used as the engine for intrusion detection, which achieved a competitive performance higher than 94% in terms of precision and accuracy [12].

Modern IDS techniques explored DL architectures to improve the normalized data to leverage detection capability [7][9][19][20]. RNN is an algorithm to stores information from the past to repeat the architecture, which automatically holds information from the past in the form of sequences and lists. [14][15]. A paper introduces a 2D anomaly detection method for network intrusion detection. The proposed 2D anomaly detection method requires less computational power than the LSTM or RNN model but performs comparably. The proposed methods detected multiple packets with UNSW-NB15 and achieved 99.51%, 97.84%, and 97.88% accuracy on each binary, gray, original method [24]. IoT networks generate vast amounts of time-series data. RNN model can enable to identification of complex attack patterns, such as distributed denial-of-service (DDoS) attacks or stealthy intrusions. Unlike traditional methods that rely on manual feature engineering, RNNs can automatically learn relevant features from raw data, reducing the need for domain expertise and improving adaptability to new or evolving threats.

## 3. Proposed Method

In this study, the dataset consists of network traffic features such as packet sizes, inter-arrival times, protocol types, and payload information. These raw features are fed into the model, where convolution layers are applied to extract local patterns and spatial relationships within the data. For example, convolution operations can identify recurring byte sequences or protocol-specific signatures in the network packets. The output of the convolution layers is then passed to the RNN layers, which analyze the sequential nature of the data. RNNs, with their recurrent connections, can model long-term dependencies and temporal dynamics, making them effective for tasks like detecting periodic communication patterns or identifying anomalies in device behavior.

RNN is used to learn the feature representations of highly imbalanced network traffic data for discriminative classification. The learning is done $\mathbf{X}$ in batches as three-dimensional tensors such that $\mathbf{X} \in \mathrm{R}b \times t \times j$, where $b$ is the batch size and $t$ is the timestep. Information about previously seen network traffic data is stored in a hidden state, $\mathbf{h}$. A list of tensors is used to produce an initial hidden state, $\mathbf{h}init$. RNN and SRNN processes present network traffic features, $\mathbf{x}$, jointly with the initial hidden state, $\mathbf{h}init$, to produce a new hidden state, $\mathbf{h}1k$ given by Eq:

$$\mathbf{h}_{1\mathrm{k}} = \sigma_h \left( \mathbf{W}_{\mathrm{xh}} \mathbf{x}_k + \mathbf{W}_{\mathrm{hh}} \mathbf{h}_{\mathrm{init}} + \mathbf{b}_{\mathrm{h}} \right)$$

where $\mathbf{W}_{xh}$ is the kernel weight matrix used for linear transformation of the input vector, $\mathbf{x}_k$ ; $\mathbf{W}_{hh}$ is the recurrent kernel weight matrix used for linear transformation of the recurrent state, $\mathbf{h}_{init}$; $\mathbf{b}_h$ is the bias vector, and $\sigma_h$ is a Rectified Linear Unit (ReLU).

**Algorithm 1:** RNN Algorithm

**Input:** $\mathbf{X}$
**Target:** $\mathbf{y}$
**Output:** $\tilde{\mathbf{y}}$

**for** $e = 1$ *to* $u$ **do**
  **for** $k = 1$ *to* $n$ **do**
    $\mathbf{h}_0 = \mathbf{h}_{init}$
    $\mathbf{h}_{1k} = \sigma_h \left( \mathbf{W}_{xh} \mathbf{x}_k + \mathbf{W}_{hh} \mathbf{h}_0 + \mathbf{b}_h \right)$
    $\tilde{\mathbf{y}}_k = \sigma_y \left( \mathbf{W}_{hv} \mathbf{h}_{1k} + \mathbf{b}_v \right)$
    $L_k = \theta(\mathbf{y}_k, \tilde{\mathbf{y}}_k)$
  **end**
  $L = \sum_{k=1}^{n} \theta(\mathbf{y}_k, \tilde{\mathbf{y}}_k)$
  $\mathbf{W}'_{(\cdot)}, \mathbf{b}'_{(\cdot)} = \psi \left( \mathbf{W}_{(\cdot)}, \mathbf{b}_{(\cdot)} \right)$
**end**

A fully-connected dense output layer was used to classify the output of the RNN layer in RNN based on Eq:

$$\tilde{\mathbf{y}}_k = \sigma_y \left( \mathbf{W}_{hv} \mathbf{h}_{1k} + \mathbf{b}_v \right)$$

Finally, RNN is trained to perform binary and multi-class classification tasks based on the Backpropagation Through Time (BPTT) algorithm. Training loss ($L$) in RNN is minimized using the cross-entropy loss function ($\theta$). Binary and multi-class classification performance of RNN is optimized using the Adam optimization method given by Eq :

$$\mathbf{W}'_{(\cdot)} , \mathbf{b}'_{(\cdot)} = \psi \left( \mathbf{W}_{(\cdot)} , \mathbf{b}_{(\cdot)} \right.$$

where $\psi$ is the optimization function; $\mathbf{W}_{(\cdot)}$ and $\mathbf{W}'_{(\cdot)}$ are the old and new Weight matrices respectively; $\mathbf{b}_{(\cdot)}$ and $\mathbf{b}'_{(\cdot)}$ are the old and new bias vectors respectively. The optimization process is performed for $u$ epochs as presented in the Algorithm.

Before being fed into the proposed model, it is applied to the pre-processing stage. At first, the collected sequence data is formatted to a requested interval to fit the input format

of a short-term or very short-term forecast model. In the next, each formatted input data is normalized. In general, a photovoltaic system has the maximum generating electrical power capacity. Also, according to weather statistics, weather data generally vary within some range. The normalization of weather input data is computed as follows:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Where x is an input vector, the norm is the normalized input vector, xmax is the maximum value of x, and min is the minimum value of x. Input normalization helps to improve the accuracy and execution time of the training process. The proposed deep RNN model consists of multi-layer RNN with layer normalization and one fully-connected layer The input layer of RNN receives normalized input vectors computed in the pre-processing stage. To overcome the vanishing and the exploding problem of the RNN model, each RNN cell is configured with an LSTM cell introduced in Section.

## 4. Result and Analysis

The proposed FPA method is evaluated in IoT intrusion detection by comparing DT, RF, SVM, and RNN. Table 1 presents a comparative analysis of various machine learning methods and RNN.

| No | Methods | $LR^{(12)}$ | FPA-LR | $SVM^{(12)}$ | FPA-SVM | $DT^{(12)}$ | FPA-DT | $RF^{(12)}$ | FPA-RF | $ANN^{(12)}$ | RNN |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Accuracy | 98.3 | 98.7 | 98.2 | 98.5 | 99.4 | 99.5 | 99.4 | 99.5 | 99.4 | **99.5** |
| 2 | STD (+/-) | 0.0055 | 0.0052 | 0.0064 | 0.0058 | 0.016 | 0.012 | 0.014 | 0.12 | 0.021 | **0.14** |
| 3 | Precision | 98 | 98.4 | 98 | 98.45 | 99 | 99.2 | 99 | 99.2 | 99 | **99.1** |
| 4 | Recall | 98 | 98.6 | 98 | 98.58 | 99 | 99.2 | 99 | 99.2 | 99 | **99.1** |
| 5 | F1-Score | 98 | 98.4 | 98 | 98.5 | 99 | 99.2 | 99 | 99.2 | 99 | **99.1** |

The table presents a comparative analysis of various machine learning methods for a classification task, evaluating their performance using metrics such as accuracy, standard deviation (STD), precision, recall, and F1-score. The methods include Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), Artificial Neural Network (ANN), and Recurrent Neural Network (RNN). Each method is evaluated in two variants: a baseline version and an enhanced version using a Feature Processing Algorithm (FPA).

All methods demonstrate high accuracy, with values ranging from 98.2% to 99.5%. The FPA-enhanced versions generally show slight improvements over the baseline methods. For instance, FPA-LR achieves 98.7% accuracy compared to 98.3% for baseline LR. The STD values indicate the variability in performance. Lower STD values suggest more consistent results. FPA-enhanced methods typically have lower STD values, indicating improved stability. For example, FPA-DT has an STD of 0.012 compared to 0.016 for baseline DT.

Precision values are consistently high across all methods, with FPA-enhanced versions slightly outperforming the baselines. FPA-SVM achieves a precision of 98.45%, compared to 98% for baseline SVM. Recall values follow a similar trend to precision, with FPA-enhanced methods showing marginal improvements. FPA-RF has a recall of 99.2%, compared to 99% for baseline RF. The F1-score, which balances precision and recall, is also consistently high. FPA-enhanced methods generally achieve slightly higher F1 scores, with FPA-ANN and RNN both reaching 99.1%. Overall, the table highlights that while all methods perform well, the FPA-enhanced versions tend to offer slight improvements in accuracy, precision, recall, and F1-score, along with reduced variability in performance.

This suggests that incorporating a Feature Processing Algorithm can enhance the robustness and effectiveness of machine learning models for classification tasks.

If stability is a priority, the FPA-DT (99.5%) and FPA-RF (99.5%) models emerge as more favorable choices. These models achieve comparable accuracy to RNN but with significantly lower standard deviation values (0.012 for FPA-DT and 0.014 for FPA-RF). This lower variability implies that these models deliver more consistent and predictable results, which can be beneficial in real-world applications where reliability is crucial, such as medical diagnostics, financial forecasting, and automated decision-making systems. Furthermore, the Flower Pollination Algorithm (FPA) optimization consistently enhances traditional machine learning models. The improvements can be observed in models such as FPA-SVM (98.5% accuracy) surpassing SVM (98.2%) and FPA-LR (98.7%) outperforming LR (98.3%). This demonstrates that FPA not only refines predictive accuracy but also contributes to stabilizing model performance by reducing variability.

## 5. Conclusion

In this study, we propose RNN to address IDS with a huge sample of datasets in the training and testing phase. The RNN model demonstrates the highest accuracy (99.5%) and the best overall performance across all evaluated metrics, including precision, recall, and F1-score (each scoring 99.1%). This indicates that RNN is the most effective model in terms of predictive capability. However, despite its superior accuracy, it also exhibits the highest variability, as reflected in its standard deviation (STD = 0.14). A high standard deviation suggests that the model's performance fluctuates significantly across different test cases, making it less reliable in scenarios requiring consistent results.

In future work, to enhance the model's performance, advanced RNN variants such as Long Short-Term Memory (LSTM) or Gated Recurrent Units (GRUs) are often used. These variants address the vanishing gradient problem inherent in traditional RNNs, allowing the model to learn from longer sequences more effectively. The final layers of the model typically include fully connected layers and a softmax or sigmoid activation function for classification, enabling the model to predict device types or detect anomalies. By automating feature extraction and leveraging the temporal structure of the data, this approach reduces the reliance on manual feature engineering, improves generalization to new devices, and enhances the model's ability to adapt to evolving IoT environments. This methodology aligns with recent advancements in deep learning for IoT security, as demonstrated in studies like those by Nguyen et al. (2020) and Kolcun et al. (2020), which highlight the effectiveness of combining convolutional and recurrent layers for IoT device identification and anomaly detection.

## References


[1] I. Ullah and Q. H. Mahmoud, "Design and Development of a Deep Learning-Based Model for Anomaly Detection in IoT Networks," IEEE Access, vol. 9, pp. 103906-103926, 2021.

[2] A. J. Siddiqui and A. Boukerche, "Adaptive Ensembles of Autoencoders for Unsupervised IoT Network Intrusion Detection," Computing, vol. 103, no. 10, pp. 2221-2242, 2021.

[3] X. Kan, Y. Fan, Z. Fang, L. Cao, N. N. Xiong, D. Yang, and X. Li, "A novel IoT network intrusion detection approach based on Adaptive Particle Swarm Optimization Convolutional Neural Network," *Information Sciences*, vol. 570, pp. 1-20, 2021.

[4] A. Basati and M. M. Faghih, "APAE: an IoT intrusion detection system using asymmetric parallel auto-encoder," *Neural Computing and Applications*, vol. 33, pp. 1-15, 2021.

[5] A. K. Pani, "An efficient algorithmic technique for feature selection in IoT-based intrusion detection system," *Indian Journal of Science and Technology*, vol. 14, no. 25, pp. 1-10, 2021.

[6] I. Essop, J. C. Ribeiro, M. Papaioannou, J. Rodriguez, G. Zachos, and G. Mantas, "Generating datasets for anomaly-based intrusion detection systems in IoT and industrial IoT networks," *Sensors*, vol. 21, no. 6, pp. 1-15, 2021.
[7] Rajaprakash, S., et al. "RNN-Based Framework for IoT Healthcare Security for Improving Anomaly Detection and System Integrity." *Babylonian Journal of Internet of Things*, 2024.
[8] D. C. Attota, V. Mothukuri, R. M. Parizi, and S. Pouriyeh, "An Ensemble Multi-View Federated Learning Intrusion Detection for IoT," *IEEE Access*, vol. 9, pp. 1-14, 2021.
[9] M. Zhong, Y. Zhou, and G. Chen, "Sequential model-based intrusion detection system for IoT servers using deep learning methods," *Sensors (Switzerland)*, vol. 21, no. 7, pp. 1-18, 2021.
[10] E. S. P. Krishna and T. Arunkumar, "Hybrid Particle Swarm and Gray Wolf Optimization Algorithm for IoT Intrusion Detection System," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 4, pp. 1-10, 2021.
[11] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, "Deep recurrent neural network for IoT intrusion detection system," *Simulation Modelling Practice and Theory*, vol. 108, pp. 1-15, 2020.
[12] T. D. Nguyen, P. Rieger, M. Miettinen, and A.-R. Sadeghi, "Poisoning Attacks on Federated Learning-based IoT Intrusion Detection System," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1-15, 2021.
[13] A. Hamza, H. H. Gharakheili, and V. Sivaraman, "Combining MUD Policies with SDN for IoT Intrusion Detection," in *Proc. IoT Security and Privacy Workshop, SIGCOMM 2018*, pp. 1-10, 2018.
[14] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the Internet of Things: techniques, deployment strategy, validation strategy, attacks, public datasets, and challenges," *Cybersecurity*, vol. 4, no. 1, pp. 1-22, 2021.
[15] M. F. Elrawy, A. I. Awad, and H. F. A. Hamed, "Intrusion detection systems for IoT-based smart environments: a survey," *Journal of Cloud Computing*, vol. 7, no. 1, pp. 1-15, 2018.
[16] X. Larriva-Novo, V. A. Villagrá, M. Vega-Barbas, D. Rivera, and M. Sanz Rodrigo, "An IoT-Focused Intrusion Detection System Approach Based on Preprocessing Characterization for Cybersecurity Datasets," *Sensors (Switzerland)*, vol. 21, no. 6, pp. 1-12, 2021.
[17] J. Liu, D. Yang, M. Lian, and M. Li, "Research on Intrusion Detection Based on Particle Swarm Optimization in IoT," *IEEE Access*, vol. 9, pp. 1-10, 2021.
[18] I. Idrissi, M. Boukabous, M. Azizi, O. Moussaoui, and H. El Fadili, "Toward a deep learning-based intrusion detection system for IoT against botnet attacks," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 3, pp. 1-15, 2021.
[19] B. U. Islam Khan, F. Anwar, R. F. Olanrewaju, B. R. Pampori, and R. N. Mir, "A Novel Multi-Agent and Multilayered Game Formulation for Intrusion Detection in Internet of Things (IoT)," *IEEE Access*, vol. 8, pp. 1-12, 2020.
[20] S. Pundir, M. Wazid, D. P. Singh, A. K. Das, J. J. P. C. Rodrigues, and Y. Park, "Intrusion Detection Protocols in Wireless Sensor Networks Integrated to the Internet of Things Deployment: Survey and Future Challenges," *IEEE Access*, vol. 8, pp. 1-15, 2020.
[21] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A. R. Sadeghi, and S. Tarkoma, "IoT Sentinel: Automated Device-Type Identification for Security Enforcement in IoT," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 96–109, 2017. DOI: 10.1109/JIOT.2016.2623390.
[22] Y. Meidan, M. Bohadana, A. Shabtai, J. D. Guarnizo, M. Ochoa, and Y. Elovici, "ProfilIoT: A Machine Learning Approach for IoT Device Identification Based on Network Traffic Analysis," in *Proc. Symposium on Applied Computing*, pp. 506–509, 2017. DOI: 10.1145/3019612.3019878.
[23] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A. R. Sadeghi, "DIoT: A Federated Self-Learning Anomaly Detection System for IoT," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4512–4524, 2020. DOI: 10.1109/JIOT.2020.2972833.
[24] M. S. Kim et al., "Network Intrusion Detection System using 2D Anomaly Detection," in *Proc. 23rd Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2022, pp. 1-4.