APPLIED ARTIFICIAL INTELLIGENCE
2024, VOL. 38, NO. 1, e2381882 (42 pages)
https://doi.org/10.1080/08839514.2024.2381882

Taylor & Francis
Taylor & Francis Group

Analysis of Encrypted Network Traffic for Enhancing Cyber-security in Dynamic Environments

Faeiz Alserhani

Department of Computer Engineering and Networks, College of Computer and Information Sciences, Jouf University, Al Jouf, Saudi Arabia

ABSTRACT

At present, encrypted data is the cornerstone of Internet communication, providing the maximum degree of privacy and security protection for all transmitted data while shielding users against potential cyber threats and attacks. However, since the Deep Packet Inspection (DPI) system is the primary layer of defense against numerous cyberattacks, applying encrypted network data poses severe issues for detection and prevention systems. In dynamic contexts such as the Internet of Things (IoT), detecting intrusion inside encrypted network traffic is vital. Yet, it is equally important to predict and prevent any cyber-attacks that may compromise the integrity and security of the network infrastructure. As a result, there is a fundamental need for methodologies based on intelligent analysis of patterns and attributes of encrypted network traffic. To satisfy security requirements in such a context, we propose an application of deep learning models for enhanced intrusion detection systems (IDS). The Tree-based Spider-Net Multipath (TBSNM) methodology is utilized, while an Advanced Encryption Standard (AES) technique is used to authenticate users. User selection is accomplished through robust Deep Reinforcement Learning with the Tabu Search (DRL-TS) algorithm, while channel selection is optimized through rigorous training employing Proximal Policy Optimization (PPO). Path selection is then determined by analyzing traffic statistics extracted from the Routing Information Protocol (RIP). Finally, an optimized IDS is established based on a Lightweight Deep Neural Network with Hunger Games Search and Remora Optimization Algorithm (LDNN-HGS-ROA). Evaluation results have shown that the proposed system architecture is effective in detecting attacks, achieving an enhanced IDS architecture with higher performance rates.

ARTICLE HISTORY
Received 9 March 2024
Revised 22 June 2024
Accepted 11 July 2024

Introduction

Within the context of the current digital environment, the ubiquity of encrypted data travel across the Internet is paramount for safeguarding sensitive information. The transition toward encryption has presented a significant

CONTACT Faeiz Alserhani fmserhani@ju.edu.sa Department of Computer Engineering and Networks, College of Computer and Information Sciences, Jouf University, Al Jouf 72388, Saudi Arabia

obstacle to conventional intrusion detection systems in the current era (Mishra et al. 2022). In light of the fact that these systems have difficulty in successfully analyzing encrypted communication, cyber attackers take advantage of this weakness, which is why it is necessary to investigate potentially creative alternatives. Sophisticated cybersecurity attacks use a range of attack vectors, including various encrypted malware, ransomware, and botnets. These attacks have widespread implications affecting a multitude of systems and networks, such as enterprise systems, mobile devices, "vehicular ad hoc networks" (VANETs), "wireless sensor networks" (WSN), and the "Internet of Things" (IoT) (Gazzan and Sheldon 2023).

The capabilities of traditional intrusion detection systems are severely limited when dealing with encrypted data. It is challenging for these systems to detect malicious behavior as they cannot examine the contents of encrypted payloads. The present research underscores the need to examine enhanced approaches and methodologies, specifically focusing on the limitations of conventional intrusion detection techniques in view of the extensive encryption use. Intelligent, computerized data collection and correlation of unusual activity are critical to network security. This activity can encourage the implementation of suitable actions to strengthen the organization's security posture when paired with a more comprehensive framework that considers the latest cyber threats (Papanikolaou et al. 2023). Cybersecurity experts stress that modern sectors such as IoT, telemetry, medical, and other systems using big data analytics need a customized security strategy. Because there are so many possible attacks and threats, securing the flow of information inside a system may be very difficult (Djenna et al. 2023). It could be challenging to detect fake data injection attacks since many Cyber-Physical Systems (CPS) lack security measures like message authorization. It is challenging to protect against eavesdropping attacks in the absence of ubiquitous encryption, particularly on antiquated technological platforms (Ahmad et al. 2024). System states need to be referenced to recognize replay attacks. Furthermore, the possibilities for network traffic defense are often limited by the use of antiquated equipment in operation.

Since most IDSs are meant mainly to handle general-purpose communication protocols like TCP and UDP, they are unable to handle control protocols like DNP3 and Modbus (Khan et al. 2023). Arithmetic operations on encrypted data are made possible via homomorphic encryption (HE); besides, they are controlled by encryption. Command signals are sent across a network of devices using connected control equipment. This procedure describes how to include homomorphic encryption within a management framework (Miyamoto et al. 2023). Moreover, the overall security of Low Power Wide Area Networking (LPWAN) is a challenging technical issue that is always evolving. Every connection uses end-to-end encryption with a counter mode (CTR) operating AES 128-bit key that has been verified (Mohamed et al. 2022). Additionally, there are

now fewer obstacles to adopting virtual private network (VPN) services. Along with protecting the interfaces, procedures, and internet protocol addresses of their interactions, users may now hide the VPN server's IP address and/or pad every data packet to an identical size before encoding, which can prevent signature detection systems (Jorgensen et al. 2023). The compromised top tier is disabled upon detection of a cyber-attack, and only the secure and stabilizing bottom tier is used to restore system stability. This is accomplished by the integration of machine learning-based cyber-attack detection into the encrypted control architecture (Kadakia et al. 2024).

Additionally, it makes things more difficult for attackers since they have to go through many micro-segments to get to their target (Zanasi, Russo, and Colajanni 2024). Detecting new malware families and internal threat attacks is the main goal for intruders. It also acts as a prescriptive Security Operations Centre (SOC) for small and medium-sized businesses. Advanced features for malware detection, network security, exchange of threat data, and security awareness are all included in the solution. Enhancing Security with the existence of Encrypted Networks using Deep Learning for Intrusion Detection presents several security and privacy problems. Some research studies address some specific issues. However, the detection difficulties faced by the IoT environment with the existence of encrypted data have not been addressed intensively.

Existing research is hindered by privacy considerations, which prevent deep packet inspection and restrict threat detection capabilities. User selection in highly dynamic circumstances provides additional hurdles, as existing approaches need a more in-depth investigation to adjust to evolving behaviors and conditions. Furthermore, inappropriate training practices are a problem, as current techniques rely on offline instruction. This method is difficult for large contexts because of the vast amount of network data that requires constant retraining. Retraining necessitates gathering data from different nodes, which might cause privacy problems and processing time issues. Balancing the need for effective retraining with privacy protection remains a serious challenge. The use of encryption has become an essential component, in protecting the integrity and confidentiality of data, but at the same time, it concurrently makes the process of intrusion detection more difficult. Hence, the purpose of this research is to leverage deep learning capacities for intrusion detection specifically applied in IoT settings, in order to contribute to addressing the growing number of cybersecurity problems related to encrypted communication systems.

The motivation behind this research effort originates from the issue that encryption has built-in security protections, and methods for analyzing encrypted network traffic differ from those used for regular, unencrypted data. When decryption is used on the data under examination, it introduces serious privacy issues and technological complications that

fundamentally undermine the secure end-to-end communication concept. As a result, using behavior-based analysis becomes a good substitute. By concentrating on patterns and abnormalities in user behavior and data flow, this method avoids the requirement for decryption and protects privacy while successfully detecting and thwarting harmful activity within encrypted traffic streams. The following is a description of the goals being pursued.

- Develop an Efficient Authentication System for IoT nodes: Create a secure network authentication mechanism to enhance both security and efficiency in IoT environments.
- Maintain Channel Security: Design and implement a protocol for selecting a secure transmission channel, hence improving the overall security and efficiency of data transfer across network entities (IoT nodes).
- Improve Network Traffic Analysis: Use information collected from routing protocols to increase network traffic analysis and management, resulting in more intelligent and effective security solutions
- Enhance Intrusion Detection with Deep Learning: Examine and put into practice deep learning models that can precisely detect and classify intrusions that target IoT.
- Implement Robust Training Procedures: Develop effective IDS training procedures that strike a compromise between the necessity for regular retraining and privacy protection in expansive network settings. The Remora Optimization Algorithm (ROA) is used to select the most informative traffic features, and the proposed design is evaluated. Current similar studies are compared with the proposed design using performance metrics such as accuracy, authentication rate, and throughput.

In addition, the following are some of the main highlights of the research.

- The Tree-based Spider-Net Multipath combined with Symmetric Encryption methodology is utilized for authentication of network elements (IoT nodes). The system provides secure communication channels, protecting against illicit access and unauthenticated users.
- DPI is a detection model used in network traffic analysis enabling detailed examination and comprehension of data packets as they move across the network. Monitoring of network data involves providing information on traffic patterns, potential threats, and anomalies.
- Effective user and channel selection is made possible by Deep Reinforcement Learning with the Tabu Search (DRL-TS) approach, which optimizes network resource allocation and overall network performance. This approach ensures optimal resource utilization by enabling intelligent decision-making in constantly changing network settings.

- By using the RIP protocol's data to train the Deep Learning model for IDS, threat detection capabilities are enhanced. Through this relationship, security breaches may be proactively identified and mitigated by combining real-time routing information analysis with typical intrusion detection approaches.
- The proposed IDS design combines LDNN and HGS-ROA architecture to offer an efficient and intelligent detection model. This approach lowers the computational cost and increases the speed and scalability of the IDS, making it suitable for usage in resource-constrained environments such as IoT.

The remaining sections comprise the remaining sections of this manuscript: The literature evaluation of earlier studies that are more relevant to our research is covered in Section II. The main issue statements that have been addressed in previous publications can be listed in Section III. Section IV describes the current research methodology for the proposed work, which contains a protocol, a mathematical representation, and a pseudocode. Section V describes the experimental findings as well as a comparison of the proposed and present works. Section VI concludes the proposed study and makes plans for future research.

Literature review

In this research, we have conducted a comprehensive literature review to explore available IDS models in general, besides models that operate with existing encrypted data in network traffic. Ilca, Lucian, and Balan (2023) proposed a novel anomaly-based intrusion detection system for IoT networks that leverages a Deep Learning approach. Specifically, a DNN model that removes highly correlated features using filter-based feature selection has been presented. Additionally, the model is adjusted using a range of parameters and hyperparameters. The UNSW-NB15 (Moustafa et al. 2015) dataset, which includes several types of attacks, has been used for evaluation. The authors (Sharma et al. 2023) presented a better understanding of anomaly detection in the context of IoT network data analysis by examining the unrealized potential of merging graph theory with machine learning algorithms. To assess how effectively graph theory representation increases classification accuracy, they used a comprehensive experimental approach that includes feature analysis, data preparation, visualization, and machine learning model comparison. More specifically, they converted the network information about traffic into a graph-based architecture where devices are represented as nodes and communication instances are represented as edges. Next, they integrated these graph properties into the ML algorithms. Graph theory

analysis of network data reveals a small but noticeable increase in the performance of the assessed machine-learning models.

(Al-Bakhat and Almuhammadi 2022) presents a hybrid deep learning strategy for detecting abnormalities in encrypted network data. The authors use CNNs and LSTM networks to capture both spatial and temporal aspects of traffic data. Their strategy entails pre-processing encrypted traffic to extract key information and then training the hybrid model to detect abnormalities. However, the model's performance may be heavily reliant on a specific training dataset, limiting its application to other types of network conditions. The hybrid strategy, while effective, is also computationally demanding and may necessitate large resources. Wang, Wang, and Sun (2022) proposed a deep-forest-based mechanism for identifying malicious data in encrypted traffic. The deep forest model uses a cascade of decision trees to repeatedly revise its predictions. However, this paradigm may suffer with scalability, particularly when dealing with significant amounts of network traffic. Moreover, the designed system depends on the quality of extracted features, which may not be effective under different traffic settings. Alwasel et al. (2023) have proposed MUSE model which it is a deep hierarchical stacked neural network system designed to identify malicious activities in real-time and accurately that alters the payload or meta-information of dataflow between the edge, core, and IoT clouds. While smaller models in the periphery require much less time to train, the central cloud's massive models take a very lengthy period. Layers of learned edge cloud models are aggregated and combined with the MUSE system to build a partly pre-trained core cloud model. This improves the accuracy of recognizing massive core cloud models as well as the training time. However, complexity in terms of time is a challenge that often arises with advanced deep-learning models. In order to enable unsupervised learning of the model, the authors of (Gupta et al. 2022) present an IDS model that makes use of the deep learning technique conditional generative adversarial network (CGAN). Additionally, they incorporate an eXtreme gradient boosting (XGBoost) classifier to expedite result comparison and visualization. Since the proposed system creates fake data to fool attackers. The settings were chosen so that the model would perform at its best without requiring major adjustments or additional work. The multiple layer network allows the model to learn from dataset samples, resulting in a more effective training procedure. The author in (Sood et al. 2022) has proposed an innovative approach to detect intrusion against IoT devices, using DL models. To identify malicious traffic that may start an attack on linked IoT devices, the system employs a four-layer deep fully linked (FC) network architecture. The proposed system has been built to be protocol-neutral to make implementation easier. In the course of the experimental performance investigation, their design performs dependably for both simulated and actual invasions.

The study by Pradeepthi and Maheswari (2023) introduces a new method for intrusion detection that safeguards data from attackers using encryption. The suggested technique's operating manner is split into three distinct phases. Step one: Use a state-of-the-art hybrid approach to machine learning and deep learning to detect network breaches. The second step is to identify the different kinds of attacks by using a sophisticated deep learning system. Users are notified automatically of potential network attacks based on the kind of attacks that are likely to occur. Step 3: The data is securely stored using state-of-the-art encryption technology. Network users will be able to access the storage's data after the four-barrier authentication procedure is complete. However, the lower-level properties of the proposed model have not been tuned with respect to the SVM's objective, which is a drawback. Moreover, the loss function gradients approach zero when more layers with distinct activation functions are added to neural networks, making the network challenging to train. The authors in (Salvakkam et al. 2023) proposed presented a novel approach for identifying cloud computing breaches. Comparative evaluations have been made using various datasets, including KDDcup 1999, UNSW-NB15, and NSL-KDD datasets. They have investigated the shortcomings of the current IDS in order to construct a more accurate and improved IDS. However, to perform difficult tasks on the DL and ML versions, planning is required, and network information must be updated rapidly, swiftly, and permanently to retrain models. To effectively categorize in-vehicle network data, authors in Lo et al. (2022) suggested a hybrid deep learning-based detection system for intrusions (HyDL-IDS)," and that is using a spatial-temporal description. The long short-term memory (LSTM) model and convolutional neural networks (CNNs) are sequentially used to autonomously extract spatial and temporal information from in-car network data. A benchmark dataset for automobile hacking has been used to evaluate the proposed HyDL-IDS. Even though the authors proposed a CNN-LSM model using supervised learning, adversarial, and unsupervised methods may be investigated to find zero-day attacks. Moreover, the investigation of semantic features may enhance the HyDL-IDS performance.

In (Alrayes et al. 2023), a deep neural decision forest (DNDF) method has been established to improve the ability of classification trees. This technique capitalizes on deeper systems' ability to absorb information descriptions. The CICIDS 2017 dataset was their initial choice for traffic in network analysis. The functioning of the DNDF algorithm was afterward evaluated using two more datasets: an additional collection of network usage data and CICIDS 2018. Their work showed that DNDF, a mix of DNNs and decision trees, outperformed reference approaches with a remarkable accuracy of 99.96% in latent models in deep layers using the CICIDS 2017 dataset. This success emphasizes the potential of DNDF in network security and IDS because of its improved feature representation, optimized modeling, and resilience to

imbalanced and noisy data from the input. In Ilyas and Alharbi (2022), five distinct ML classifications for various attack types have been developed. CSE-CIC-IDS2018 dataset has been used for evaluation. However, there are some limitations including, using a restricted set of valid characteristics, to be used in developed decision trees, random forests, Gaussian naive Bayes, support vector classifiers, and multi-layer perceptrons. To identify different types of network intrusions, authors in Lin et al. (2022) created PEAN, a new design of multimodal deep learning system for categorizing encrypted messages. PEAN uses a self-attention method, raw byte, and length sequence input to understand the complex connections between network packets in a significant flow. Moreover, PEAN's network packet characterization skills were enhanced by the usage of unsupervised pre-training. However, in comparison to other baseline methods, PEAN requires more GPU RAM and takes longer to train. When used as an inference service, PEAN consumes significantly less GPU RAM. It is possible to train a model consistently and then deploy it to a gateway or edge device by using a high-performance central device. Training time will not affect the online inference service experience because DL algorithms are learned offline.

The authors of (Chuang and Ye 2023) have presented the Reptile-TL model, which uses transfer learning and meta-learning methodologies to increase the target domains' performance by leveraging the source domain. They investigated the performance of six models in three crucial aspects and assessed the benefits of transfer learning in intrusion detection using the SDN data environment. The experiments that were conducted imitated three practical problems: zero-day attack, short sample sizes, and class imbalances. The model combines deep learning and transfer learning by employing a CNN pertaining to the source domain as the feature extractor for the target domain, followed by Reptile meta-learning to obtain suitable initial parameters. The system they constructed scored 0.71 for identifying anomalies in unidentified attacks; second, it scored 0.98 and 0.51 for anomaly detection and attack type identification for small samples; third, it scored 1.00 and 0.91 for Class imbalance detection for anomalies and attack mechanism identification. The authors in Rezaei and Liu (2019) give a complete review of the use of deep learning algorithms to categorize encrypted network data. The authors examine a variety of deep learning models, including CNNs, recurrent neural networks (RNNs), and autoencoders, and analyze their advantages and disadvantages in terms of encrypted traffic categorization. Although comprehensive, the overview may lack in-depth insights into specific models or implementation challenges. Also, 2. Rapid evolution requires ongoing changes to the overview. The study in (Fu, Li, and Xu 2023) describes a unique approach for real-time detection of unknown encrypted malicious traffic based on flow interaction graph analysis. The suggested method generates a flow interaction graph that depicts the linkages and interactions among

various network flows. The ultimate goal is to uncover unusual patterns of malicious activity in these graphs by analyzing their structure and properties using sophisticated graph neural networks (GNNs), even when the traffic is encrypted. However, real-time detection demands powerful resources and efficient algorithms to evaluate flow interaction graphs swiftly. Scalability and generalization issues may also affect the method's ability.

Authors in Ullah et al. (2023) present IDS based on transformers and transfer learning for networks with unbalanced traffic. Network feature representation and feature interactions in unbalanced data are both learned by IDS-INT using transformer-based transfer learning. Originally, descriptions of network interactions provide detailed information on all forms of attacks, including information about hosts, nodes in the network, attacker types, sources, and so on. To learn the precise feature representation, the transformer-based learning by transfer technique uses their semantic anchors. Then, the "Synthetic Minority Oversampling Technique (SMOTE)" is used to identify minority attacks and balance unusual traffic. The CNN model looks for deep features that can be extracted from the balanced network flow. In the end, several attack types are identified from the deep features by using the CNN-LSTM hybrid approach. In Meddeb et al. (2023), the authors have used a stacked autoencoder-based method called Stacked AE-IDS to reduce correlation and model significant properties of MANETs with high-level description. Using this technique, the input is replicated with less correlation, and the DNN-IDS uses the autoencoder's output as its input. The majority of possible attacks identified in mobile network routing systems can be detected by this IDS design, but the focus is Denial of Service (DoS). Through modeling high-level representations of pertinent information and reducing correlation, the Stacked AE-IDS approach improves the efficacy of IDSs in detecting attacks on MANETs. The proposed technique is especially useful for MANET security since it focuses on DoS attacks and how they impact mobile network routing services. This method has not been evaluated using more complex attacks. The study of (Xu et al. 2023) provides a proposed model for intrusion detection that integrates an attention mechanism with several deep-learning models. Several benefits of this hierarchical paradigm include the following: first, the characteristics of the traffic data are extracted, and noise is removed using the SCDAE model; second, spatial dimension detection of CNN model, has been used to extract the geographical characteristics of network traffic data; thirdly, network traffic data's temporal components may be mined, because Bi-directional long short-term Memory (BiLSTM) can precisely capture the relationship between the front and back properties; fourthly, each time step's weighted output has a Self-Attention mechanism added to it to collect and retain pertinent data. Therefore, a CNN-BiLSTM-Attention model has been created, and the Softmax classifier has been used to obtain the classification results. The authors of (Latif et al. 2024) presented a design of Convolutional

neural networks (CNNs), along with genetic algorithms (GAs). A bootstrap aggregation ensemble methods are integrated synergistically inside the framework as a tri-layer architectural approach. It is used in three crucial phases: Initially, they transform the cutting-edge cybersecurity dataset Edge_IIoTset into image data so that CNN-based analytics may be performed. Second, each base learning model's hyperparameters are adjusted using GA, which improves the model's performance and flexibility. Lastly, to further strengthen the IDS's resilience, the outputs of the best-performing models are combined using ensemble methods.

In Hsiao and Sung (2022), sensing data is sent over a wireless sensing network to the front-end incorporating a microcontroller of the farm so that the data can be fused. Following the completion of the fusion of the sensing data, the processed data is sent to the farm's data processing center, where the blockchain algorithm is packaged. After the system packages the data, it sends them to the cloud database for storage. Remote personnel may simultaneously manage and analyze the data in the cloud database. Blockchain-based data encapsulation is used to effectively stop hackers from stealing or destroying any data. In the experimental stage, a full blockchain encapsulation database system is put into place, allowing the operator to encrypt data that has been fused for blockchain technology at the distant end. The solution may especially improve the security of data processing throughout the blockchain's encryption process. Lastly, a private cloud database securely houses each encapsulated piece of data. The authors in (Yan et al. 2022) present a technique of an MG-distributed control system for improving cybersecurity that is based on the quantum key distribution (QKD). Measurement-device-independent QKD (MDI-QKD) is added to counter side-channel attacks and make the framework suitable for industrial applications. This work includes the following contributions. 1) To ensure data transmission security in MG distributed control, a new QKD-based quantum-secure control architecture is created. 2) A scalable QKD network for MG control is formed using MDI-QKD with an asymmetric protocol. 3) To make real-time parameter adjustments in QKD systems, a deep neural network (DNN)-based technique for quick parameter optimization is presented. The research study of Faragallah et al. (2022) proposes an efficient cybersecurity solution for protecting high-efficiency video coding (HEVC) frameworks. The suggested selected cybersecurity HEVC framework employs a robust hybrid technique based on watermarked and selective encoding to guarantee privacy and intellectual property rights for the transported HEVC data. The watermarking approach employs singular value decomposition, and the homomorphic transform of the discrete wavelet transforms to make watermarked HEVC streams more resilient to attacks. Furthermore, the selective encryption strategy, which encrypts the motion vector difference using a combination of the chaotic logistic map and discrete cosine transform sign bits, offers the feature of HEVC format compliance with minimal encrypting overhead

expenses. In Bakhsh et al. (2023), authors developed a method to defend IoT systems from online attacks; DL-based IDS is proposed, using Random Neural Networks (RandNN), Long Short-Term Memory (LSTM), and Feed Forward Neural Networks (FFNN). This study reports on the possible benefits of each DL model. For instance, LSTM excels at identifying long-term relationships in network traffic, but FFNN can handle complicated IoT network traffic patterns. The RandNN model adapts and learns from network data by using its random connections and flexible dynamic.

To sum up, the problem of examining encrypted communication, particularly in the dynamic and varied nature of IoT networks, is frequently not adequately addressed by existing methods even with the progress made in ML and DL. Developing a reliable method that can effectively analyze encrypted traffic while preserving privacy and functionality is required. This can be achieved by deep analysis of malicious activity in network traffic using state-of-the-art DL approaches while avoiding data decryption in intermediate network boxes. This can result in the construction of more flexible and robust intelligent IDS frameworks that provide resilient security.

Problem statement

We propose a detection method to identify and neutralize malicious devices exhibiting unusual incursion behavior. This method is expected to provide scheduled techniques among the necessary bandwidth of individual device measures. An algorithm such as the Deep Reinforcement Learning (DRL) algorithm can minimize communication between IoT devices. The issues of privacy and efficiency are based on incapacity due to complete packet inspection. More research efforts are required to optimize security measures in IoT, including investigating complex Q-learning algorithms that adapt to changing settings and dynamic behaviors. We use symmetric encryption, such as AES, for authentication in our Deep Packet Inspection (DPI) procedure to properly fulfill privacy standards. By combining Deep Reinforcement Learning with Tabu Search (DRL-TS), user selection will undoubtedly adapt to changing conditions and dynamic behaviors. We also used a non-zero-sum game with imperfect information in DRL to represent the interaction between the attacker and IDS. Both sides may dynamically alter their behavior in this game depending on their anticipated rewards and strategy. Additionally, the Tabu search algorithm is a straightforward method that does not need much memory space and is appropriate for implementation in dynamic contexts. It is also efficient in terms of the amount of memory required and the speed at which it converges. Hybrid technology will also enable efficient user selection. Employing the Lightweight Deep Neural Network (LDNN) for both online and offline training procedures will overcome the retrained process. Using LDNN, develop an inexpensive, light device that can fully extract characteristics of data at the

same time as reducing the computing burden by expanding and compressing feature maps to reach a pace of less than one gigabyte an hour. To achieve this, the HGS-ROA has been encompassed in the recommended approach for selecting the ideal attributes that are suitable for accurate detection of attacks and it has produced a useful result in a reduced processing period. The hybrid LDNN-HGS-ROA provides a great technique for finding incursions in terms of identifying intrusions. For the objective of choosing the most efficient features to recognize invasions in the Internet of Things systems, a hybrid LDNN-HGS-ROA strategy must be developed to reduce the number of harmful attacks and to pinpoint the predictions of unidentified attack types.

Proposed method

In order to traverse the encrypted landscape, traditional approaches, which are dependent on patterns and signatures, are not well prepared. DL, in addition to, ML has emerged as a potentially fruitful option owing to its ability to learn complex data representations on its own. Thus, it is required to develop a more reliable framework for intrusion detection by using the capacity of learning models to recognize patterns of encrypted communication. Attackers are always refining their tactics in the current cyber threat environment, which makes things dynamic and intricate. Traditional security measures may be easily circumvented by polymorphic attacks, especially if they employ encryption as their cover identity. As a result, in order to properly handle the encrypted nature of network data and effectively counter emerging threats, a paradigm shift in the methodologies utilized for intrusion detection is required

To address the limits of security systems in protecting against malicious encrypted communication, we propose a universal intrusion detection system based on the capabilities of intelligent classification models, as seen in Figure 1. This can be explained in four stages, and the proposed flow can be explained as follows,

- Network Construction and Authentication
- Efficient User and Channel selection
- Network Traffic Analysis
- Deep Learning-based Intrusion detection

First: Network construction and authentication

Network model construction

Initially, the "Tree-based Spider-Net Multipath" architecture is implemented, which combines the resilience of spider-net setups with the dependability of tree frameworks. This design guarantees both efficient interaction channels and flexibility in response to changing network circumstances. By sending

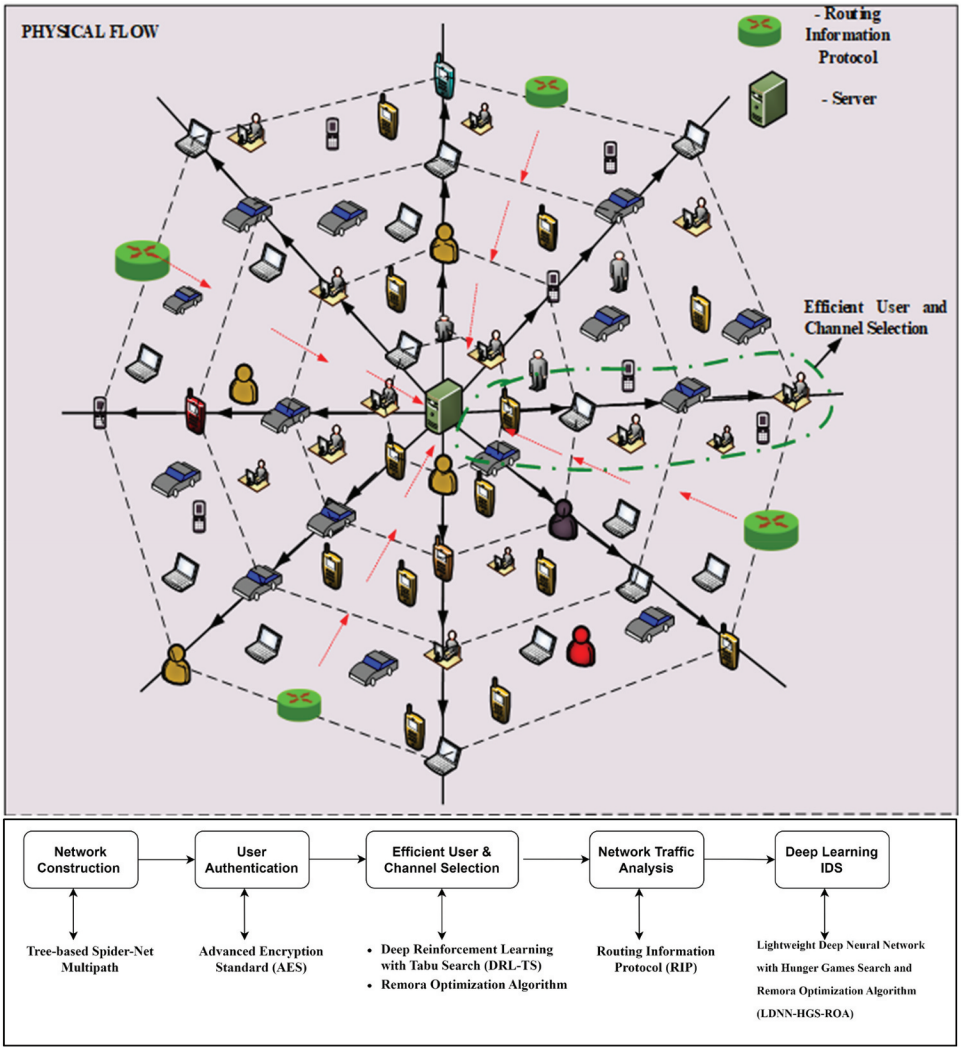

Figure 1. Overall architecture of the proposed method.

data across many pathways at once, routing over multiple paths improves load balancing and dependability. To maximize communication pathways and improve its overall effectiveness, the network constantly adjusts to changing circumstances.

Spider-net structure. A "Spider-Net" structure can be represented such as,

(1) The centroid node "O" is the detection of an event.

(2) The spider-net length is denoted as S_L and Equation (1) can be,

$$S_L = 2 * O_r * E_{nth} \tag{1}$$

(3) The number of links on radial can be "L" and the spider-net radius can be represented as "r" in Equation (2).

$$L = (2 * \pi * r)/S_L \tag{2}$$

(4) The number of links in spiral "R" can be formed on centroid node "O."

(5) The artificial spider-net diameter can be represented as "A" which refers to the spider-net coverage based on geographic area.

(6) The sector angle denotes two adjacent radial links that can be represented as $'\theta'$,

$$\sum_{i=1}^{n} \theta i = 2\pi \tag{3}$$

$$\theta = 2\pi/L \tag{4}$$

According to Equation (3), the total of all the angles between the radial links is $'2\pi'$. According to Equation (4), the sector angle $'\theta'$ may be computed as a ratio of $'2\pi'$ with the number of radial connections "L."

(7) The area of the triangle ABC be "T" in Equation (5) can be

$$T = \sum_{i=1}^{n} 1/2\ L_1^2 \sin\theta_1 \tag{5}$$

Equation (6) provides the maximum area coverage of the spiral or polygon spider nets.

$$T = \sum_{i=1}^{n} 1/2\ L_n^2 \sin\theta_1 \tag{6}$$

(8) Let $'\delta'$ represent the distance in the radial link $'L_i'$ between the centroid node "O" and the first hop node from it. The next hop distance is represented by $'\delta_i'$ where "i" is the spiral link number at which the node is located. The value "P," which is determined using the formula in Equation (7), represents the total distance between "O" and the edge node "e."

$$P = \delta + \sum_{i=1}^{n} \delta i \tag{7}$$

Multi-path routing. Clones of the route discovery agents are created by the centroid node and sent along various pathways to reach the sink node to carry out the path discovery utilizing mobile agents. The energy available to the nodes (ED), the distance between adjacent nodes (Dn), the hop count (hc), and the sink node are all gathered by these mobile agents. The energy factor (Efp), distance factor (Dfp), and cost (Cp) of the path are computed by the sink node to determine the length of the path as well as the lowest and maximum energy available routes. A path can be available on nodes in a minimum energy, as shown in Equation (8); besides, a path can be available on nodes in a maximum energy *EDmax*, as shown in Equation (9).

$$ED_{min} = Min\{ED(1), ED(2), \ldots, ED(n)\} \tag{8}$$

$$ED_{max} = Min\{ED(1), ED(2), \ldots, ED(n)\} \tag{9}$$

A path can be available on nodes in a maximum energy $ED_{max,}$ and Equation (9) can be explained. The available factor path of energy is denoted as "Efp" and is expressed as in Equation (10).

$$Efp = ED_{min}/ED_{max} \tag{10}$$

The path of a distance factor "Dfp" is calculated as a distance path of a ratio "Q_d" with the path of a total hop count "Q_{hc}" as explained in Equation (11).

$$Dfp = Q_d/Q_{hc} \tag{11}$$

"Cf" represents the cost function found in Equation (12). Paths with lower "Cf" are given priority.

$$Cf = Efp + Dfp \tag{12}$$

Spider-net-based data collection and routing may be implemented using multipath architectures, as shown in Figure 2.

Network traffic analysis

The two main components of DPI are typically the feature library and the scanning algorithm. The scanning algorithm's purpose is to match words in the IP packet load's feature library with content. Similarly, we utilize DPI techniques for network traffic analysis. It represents the network traffic identification process based on DPI, as shown in Figure 3. The complete application layer's content may be retrieved by carefully examining the IP packet load content and rearranging the data. To identify a particular application data, the data flow content is then scanned and recognized in accordance with the feature library that is already in place. To prevent an undue delay to the application, DPI needs the capability to be able to swiftly analyze, identify, and reorganize application data.

With a high degree of detection accuracy, this method can reliably identify network flows based on the feature library and the particular application to which the flows belong. However, DPI trails behind the introduction of new apps cannot detect encrypted network data streams and cannot identify application traffic that has not yet been captured in the feature library. Therefore, we achieve reliable network traffic analysis results with the multipath aware routing in the simulated network construction process, and further, this transmission will be processed with the authentication.

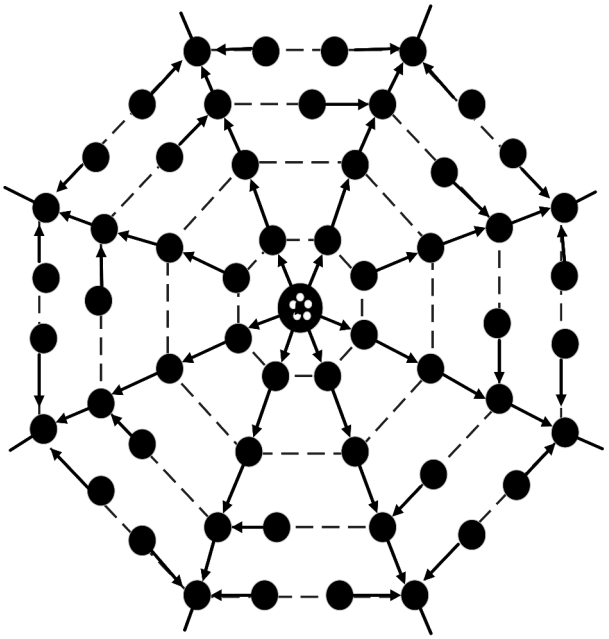

Figure 2. Tree-based Spider-Net multipath.

Authentication

We use symmetric encryption, such as the Advanced Encryption Standard (AES), for authentication in the DPI procedure to properly satisfy confidentiality and privacy standards. Users are registered using their credentials such as IP address, MAC address, ID, Password, and Biometrics, which are then used to generate a secret code. The AES encryption algorithm improves security and reduces time complexity due to its strong mathematical equation and faster key generation capability. During the authentication process, following AES decryption, the original data is obtained, and then it is compared to the user input. If similarity is established, authentication is accomplished, resulting in a secure and efficient process.

The process as shown in Figure 4 starts by generating a random string on the transmitter side, which can be used as a key or part of the payload. The sender's payload may be compacted for greater efficiency. This string is then encrypted using AES to secure the payload and then routed for transmission. Additional routing or security steps, called Relay Server including an Information Security System (RS/ISS) node, are to check for packet errors through integrity check facilities such as CRC and Hash function calculation. When the intermediate point is reached, the packet is received and decoded to examine its contents. The decrypted payload is analyzed against the malware list to ensure it does not contain malicious content, and once checked, the payload is re-encrypted using AES. The re-encrypted payload is then sent back to the recipient. The intended receiver

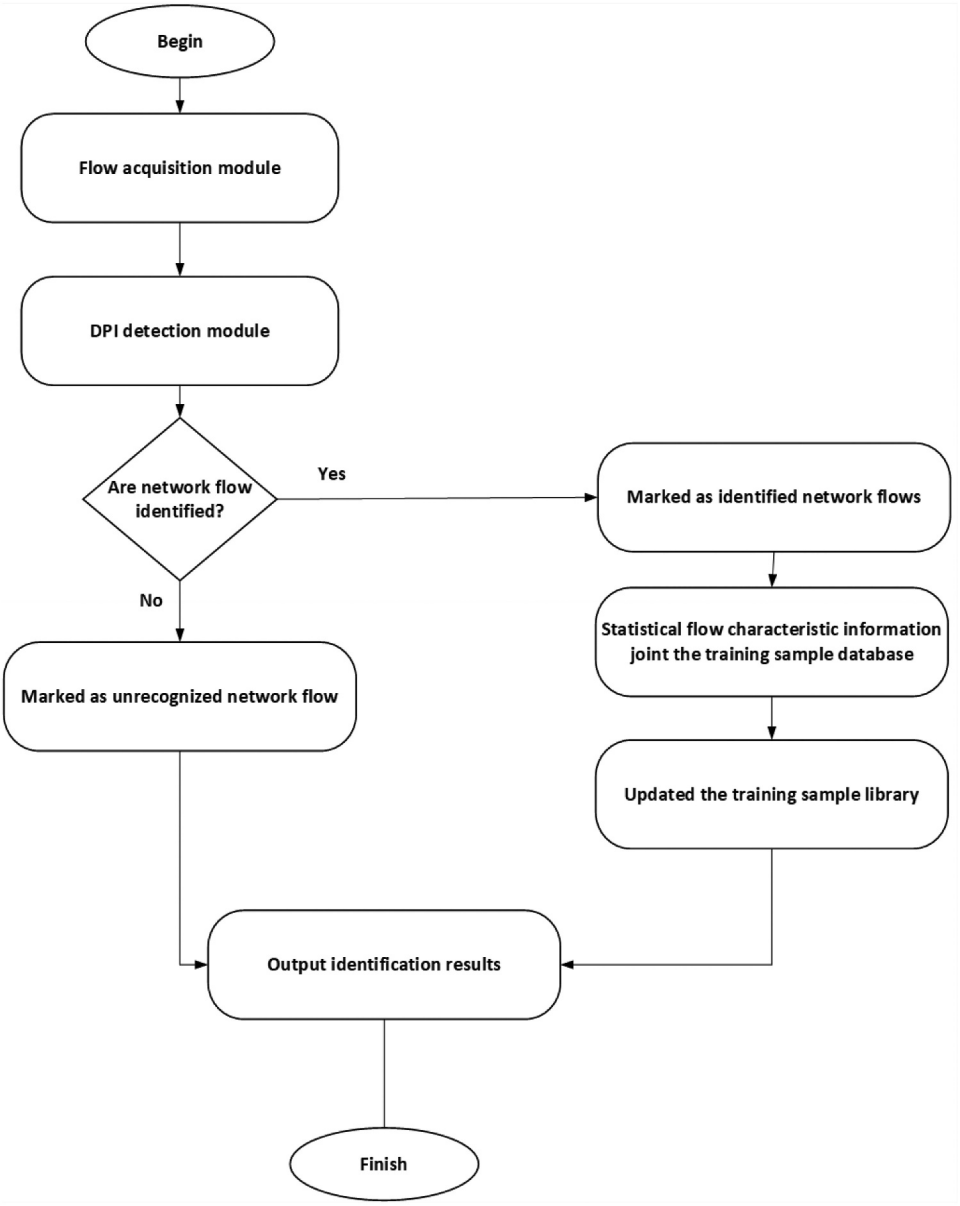

Figure 3. Network traffic identification process based on DPI.

receives the packet and decrypts the payload to access the original data. The payload undergoes another malware detection to confirm its security. If the packet is not malicious, it will be used by the recipient; otherwise, it is rejected. Decision points include generating the packet if it passes the initial malware testing and decryption phase and discarding the packet if malware is detected during intermediate on-boarding testing or later. The role of

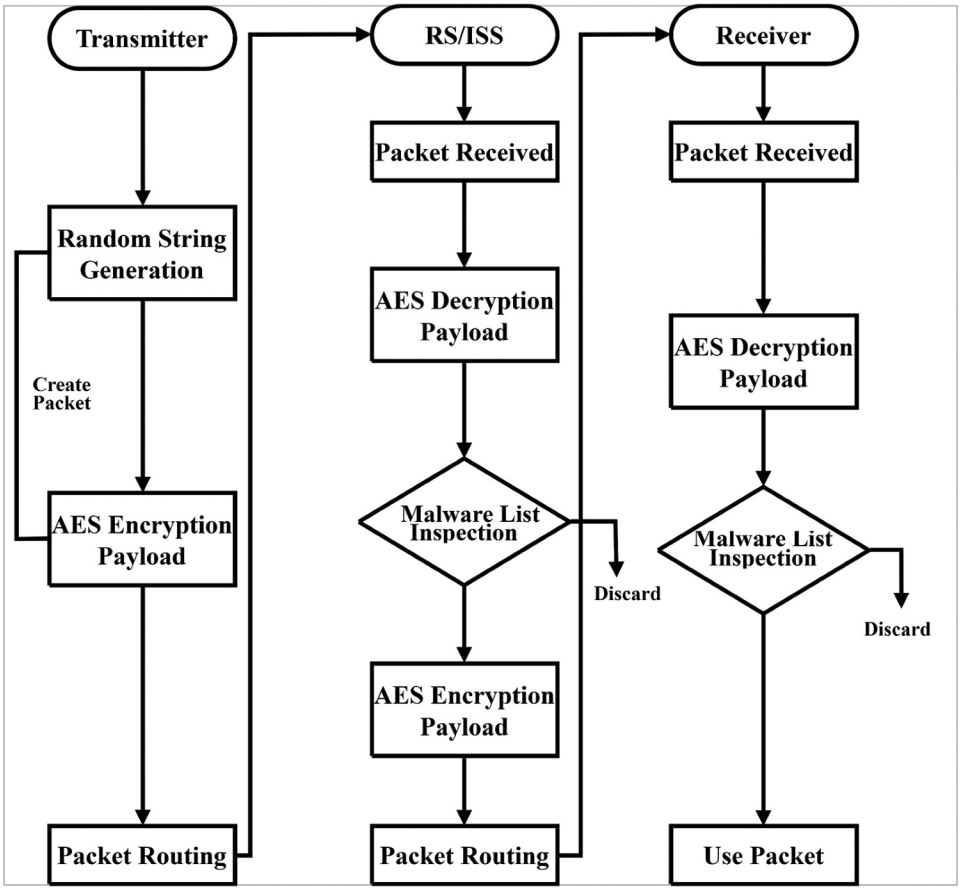

Figure 4. Deep packet inspection with AES.

this stage of the proposed system is to provide a secure data transfer protocol, emphasizing multiple layers of encryption, and malware checking to ensure data integrity and security throughout transmission.

Second: user and channel selection

The process of efficiently selecting users for intrusion detection in an IoT environment involves a sophisticated integration of Deep Reinforcement Learning with Tabu Search (DRL-TS). Initially, the problem is meticulously formulated, outlining criteria for effective client selection, such as trust, energy levels, available bandwidth, and network conditions like Received Signal Strength Indicator (RSSI) and Channel State Information (CSI).

When creating a robust client selection system, it is critical to incorporate various components of state representation, action space, and reward functions. This solution uses Deep Reinforcement Learning (DRL) to maximize client selection by making intelligent decisions based on previous data and

network conditions. The state representation is fully structured to capture relevant information, whereas the action space is defined to include potential decision outcomes. A critical component is the reward function, which evaluates the preference of specific activities and motivates the model to select users that contribute positively to IDS metrics. DRL is a model that incorporates historical data of client attitude, network performance metrics, and security requirements. Hence, the DRL model is built on previous experiences and relevant measurements. Performance may be optimized using a comprehensive coverage of state representations and well-defined action spaces. This model is trained iteratively, utilizing reinforcement learning techniques to improve decision-making capabilities based on received rewards. Notably, Tabu Search is seamlessly integrated into the process to enhance solution exploration, refining decisions made by the DRL model. This integration allows for a more efficient exploration-exploitation trade-off, essential for adapting to the dynamic nature of IoT environments. This holistic approach, DRL-TS, achieves low time consumption and reduced rounds and enhances overall intrusion detection performance in IoT environments. Proximal Policy Optimization is seamlessly integrated to handle the extended action space, enabling the model to adapt and learn channel selection strategies. The training loop is expanded to simultaneously optimize the channel selection and intrusion detection policies.

Deep reinforcement learning with tabu search

The construction of a Deep Reinforcement Learning model, which incorporates past information on client behavior, network circumstances, and intrusion detection results, forms the basis of the methodology. Using RL approaches, a flow may be trained repeatedly to enhance efficient decision-making on incoming reward packets. Remarkably, Tabu Search is included in the procedure to improve solution discovery and improve the DRL model's decision-making. To adjust to the dynamic nature of IoT settings, a more effective exploration-exploitation trade-off is made possible by this integration. In addition to reducing rounds and using less time, this all-encompassing method, Deep Reinforcement Learning with Tabu Search, or DRL-TS, also improves overall intrusion detection performance in IoT contexts.

Deep reinforcement learning. DRL has been utilized as a method of intrusion detection system to analyze incoming network traffic and sophisticated user behavior, which can involve new and zero-day attacks. The model operates in a real-time network context that considers accuracy along with processing time. The action of the RL agent is the same as the result of detecting an intrusion, and the result of the detection may be decided as a reward or penalized. Data in network traffic can be considered as RL environmental state variables. As seen in Figure 5a, the RL architecture operates in two

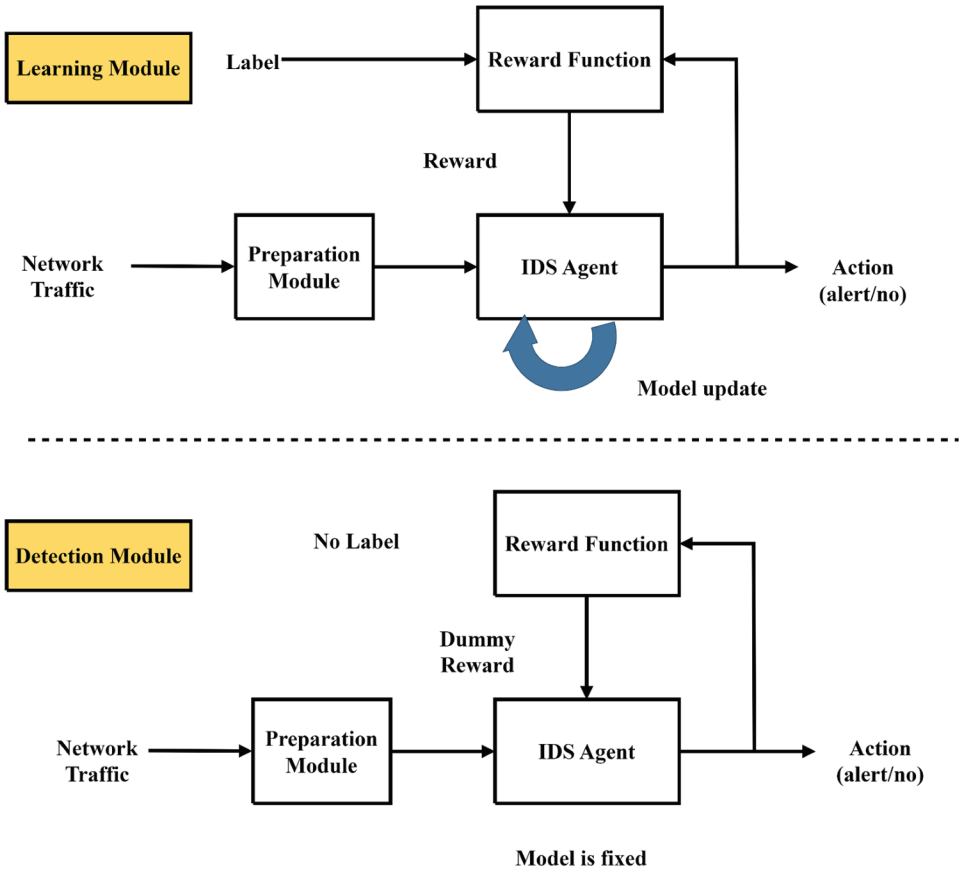

Figure 5a. Deep reinforcement learning.

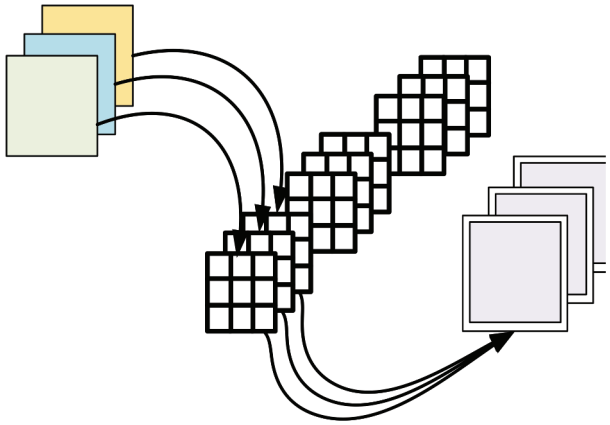

Figure 5b. Standard convolution.

modes: detection mode and learning mode to achieve self-updating progression. The detailed flow processing of these modalities is:

- **Learning Mode**:
  (1) The state variables can be processed using the RL agent, and it can be converted into raw traffic network data, and it returns to an action.
  (2) The label and action are based on calculating the reward using the reward function module, and it can be feedback sent to the RL agent.
  (3) The policy can be updated based on rewards and states of an RL agent.
  (4) Return to step 1.
- **Detection Module**
  (1) The state variables can be processed on the RL agent, and it can be converted into raw traffic network data, and it returns to an action.
  (2) The module of the reward function gives a dummy reward to the RL agent and continues the working process.
  (3) Continue to step 1.

The system computes detection performance to track a DRL agent's reward, while it is in the learning mode. In order to improve intrusion detection performance, the system replaces the existing data with a new detection model whenever the incentive decreases. In the detection mode, an RL agent and a fixed detection model process network traffic along with the use of a dummy reward. This mode serves just as a means of functioning. A switch flag can be used by the system to transition between modes, enabling flexible operation, evaluation, and changes to the detection model whenever needed.

Tabu search. Tabu Search (TS) is a sophisticated optimization technique that employs a memory structure to monitor previously visited solutions, preventing the program from returning to them. Iteratively examining the local neighborhood, iteratively investigating the existing solution, and iteratively expanding and intensifying the search. Every viable solution, $B \in \Omega$, has a set of local neighbors that correspond to it, $N(B) \subseteq \Omega$, where Ω is a collection of possible solutions. An action known as a "move" to B' may be performed from B to arrive at an outcome $B' \in N(B)$. Even if doing so results in a decline in the goal or fitness function, TS proceeds from a solution to its best acceptable neighbor. This non-greedy nature of TS helps to keep it from being trapped in the local optimum state, as shown in Algorithm 1. The techniques being researched are labeled as "tabu" or "forbidden." The recently visited alternatives are kept over iterations in a list called the tabu list. Everything is always checked against the tabu list before proceeding. If a move is discovered in the tabu list, the technique proceeds to the next iteration; if not, the move is discarded. The tabu status

Table 1. Tabu search algorithm parameters.

Parameters	Values
Tabu List Size	7
Number of Neighbor	6
Aspiration Level	0.05
Number of Iterations	100

of a particular solution may be overridden when a set of conditions (such as an ambition level or criterion) is satisfied. Sometimes, strategies of intensification and diversification are employed to improve the search. In the first case, the prospective areas of the feasible domain are the focus of the search. In the second situation, a significant amount of the search field is considered while considering possibilities. Table 1 lists the parameters for the TS algorithm.

Algorithm 1 TS algorithm utilizing a context of packet services in reward maximization.

1. **Initialization**: The initial solution can be mentioned as B_{init}.
2. $B \leftarrow B_{init}$
3. Let g(B) be the reward achieved using a result outcome.
   $g_{max} \leftarrow g(B_{init})$
4. **while** Number of Neighbourhood iterations $\leq$ maxiter **do**
5. **Formation of Neighbourhood**: Every possible neighbor of TS in initial solution outcome B is created, expected can be listed as tabu.
6. **Selection of Neighbor**: The chosen maximum reward solution achieves B' based on the neighbors set, and the condition B' is not listed as Tabu, $B \leftarrow B'$.
7. **Updation of Tabu list**: The reward $g(B')$ is added corresponding to the fixed solution B' is added to the TL.
8. **Updation of Maximum Reward**: The maximum obtained of a g_{max}reward can be updated as follows, **if** $g(B') > g_{max}$, **then** $g_{max} \leftarrow g(B')$ **end if**
9. **end while loop**

Proximal policy optimization

The model can adapt and learn channel selection techniques because Proximal Policy Optimisation (PPO) is smoothly incorporated to handle the larger action space. To optimize both the channel selection and intrusion detection policies at the same time, the training loop is extended. The curse of dimensionality makes several reinforcement learning techniques (RL) impractical for systems with many agents, such as Q-Learning. Functional approximate, such as Deep Neural Networks (DNN), has recently gained popularity in Multi-Agent Deep Reinforcement Learning (MDRL) systems because of their effectiveness at generalizing from observed to unseen states. For a policy π_θ , which is specified by θ, the objective of this optimization is to maximize the cumulative reward throughout the system. The policy links the current state with the distribution of probability across prospective actions. Among the several RL techniques used to optimize a policy, the Policy Gradient (PG) approaches

utilize rewards to generate an estimate of the policy gradient. This estimate will be used in a stochastic gradient ascent algorithm to improve the policy.

The actor-critic framework is used in Proximal Policy Optimisation (PPO), a cutting-edge policy gradient approach for reinforcement learning, in this research. The estimated value function (critic) is used in this structure to evaluate the actions taken by the actor, and the policy (actor) is utilized to choose those actions. PPO attempts to make the greatest improvement step possible to update the existing policy while remaining true to it, by combining the criticism with the experiences of the present. With typical PG approaches, there may be a significant loss of performance due to destructively major policy changes. This issue can be solved with Proximal Policy Optimisation (PPO), which alternates between optimizing a clipped policy surrogate and gathering data through contact with the environment.

The neural network architecture may share policy and value functions. A random technique is used to train PPO, which then investigates actions using the most recent version. The starting conditions and training procedure affect the unpredictability of action selection. We utilize the policy network to observe and calculate the mean and standard deviation vectors, as shown in Algorithm 2.

Algorithm 2 Proximal Policy Optimization (PPO)

1. Initialize an actor as μ: $D \rightarrow F^{m+1}$ and $\sigma : D \rightarrow diag(\sigma_1, \sigma_2, \ldots, \sigma_{m+1})$
2. **for** i=1 to M **do**
   Run on a policy $\pi_\theta \sim N(\mu(D), \sigma(D))$ for time steps T and collect a (D_t, ht, r_t)
   Advantages estimate a $\widehat{A}_t = \sum_{t'>t} \gamma^{t'-t} r_{t'} - v(D_t)$
   Old policy update $\pi_{old} \leftarrow \pi_0$
3. **for** j=1 to N **do**
   Actor policy updation use of policy gradient:
   $\sum_i \quad {}_\theta L_i^{clip}(\theta)$
   Critic updation is:
   $L(\emptyset) = -\sum_{t=1}^{T} \quad \widehat{A}_t^2$
4. **end for**
5. **end for**

Third: deep learning-based intrusion detection

The main functionality of the proposed system is to detect various malicious activities in encrypted network traffic, utilizing the advantages of both Lightweight Deep Neural Networks (LDNN) supported by two optimization techniques: the Hunger Games Search (HGS) algorithm and the Remora Optimisation Algorithm (ROA). The LDNN-HGS-ROA algorithm, as a reliable and effective detection system, is the final anticipated outcome of this integration. Fundamentally, LDNN offers a robust defense against sophisticated cybersecurity attacks using DL algorithms to detect abnormal network patterns. It provides an advanced method for examining network data and identifying intrusion attempts. The incorporation of the HGS algorithm

increases intrusion detection efficacy by ensuring a thorough analysis of the network environment in search of optimal solutions. The HGS algorithm simulates the competitive dynamics found in the "Hunger Games," in which people attempt to outperform their competitors in order to survive. It draws inspiration from the idea of survival-of-the-fittest. In addition, The Remora Optimisation Algorithm (ROA) enhances the HGS algorithm by providing an additional degree of flexibility to the intrusion detection procedure. To provide continuous defense against emerging threats, the ROA algorithm continuously adapts the detection capability as threat landscapes change and new attack vectors appear.

Lightweight deep neural network

A crucial component of Lightweight Deep Neural Network, the lightweight unit extracts features using depth-wise convolutions. Separable and depth-wise convolutions have been employed in several effective, lightweight models to extract features. Convolution is divided into two separate phases by separable convolution: depth-wise convolution and point-wise convolution, as shown in Figures 5b, 5c. The feature map that results from convolution across depths is non-expandable and has the same total number of elements as the input layer. The convolution kernel in point-wise convolution has a size of $1 \times 1 \times M$, where M is the depth of the subsequent layer. Consequently, the preceding step's map will be weighted in the depth direction by this convolution process, generating a new feature map.

The standard-based convolution takes a $c_i \times u_i \times v_i$ as an input, and an output can be $c_j \times u_j \times v_j$. To calculate the standard convolutional layer, as shown in Figure 7, the computational cost can be,

$$c_i \cdot u_i \cdot v_i \cdot v_j \cdot k \cdot k \tag{13}$$

The separable convolution on a computational cost can be,

$$c_i \cdot u_i \cdot v_i\left(k^2 + v_j\right) \tag{14}$$

In Figure 5c, the depth-wise convolution is used with (k = 3) in the lightweight unit for obtaining features, and at just a small drop in accuracy, its computational cost is about nine times less than that of regular convolution.

The primary functionalities of Lightweight Unit A, a lightweight system with no remaining form, are sampling reduction and tensor output shape modification. The most popular technique for downsampling is to employ the max-pooling layer, which takes the highest-achieving window signal and reduces the signal dimensionality by performing translations and non-deformation processes. Lightweight Unit A can perform the downsampling function if stride = 2 is configured. The step size parameters vary in stride, define the filter's stride length for each convolution cycle, and control whether the filter's window

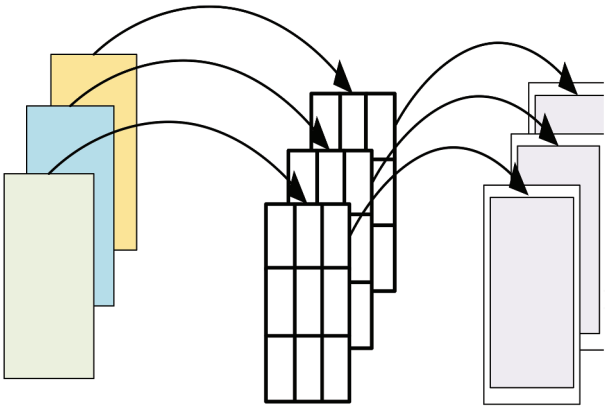

Figure 5c. Depth-wise convolution.

needs to overlap. It is possible to efficiently lower computing costs while extracting features by substituting this structure for max pooling.

An inverted residue framework, primarily used to execute the feature extract operation, is included in Lightweight Unit B, as shown in Figure 6. The main advantage of using the inverse residual structure is to avoid the problem of model excessive fitting and the deletion of the gradient; therefore, at the beginning of each unit, we first split the input into two equivalent portions. The convolution can only extract a very small number of features if the regular residual structure is employed, which means that the features are extracted using convolution after the map of characteristics has been reduced. As a result, following channel separation, the input tensor will increase through the expansion layer rather than being compressed. The expanding layer converts the low-dimensional environment to high-dimensional space using a 1×1 network structure. Modify the "Expansion Multiple" factor to achieve a balance between the number of features extracted and the model parameters. The feature map has been reduced using the 1×1 network design in the compression layer after features are extracted using depth-wise convolution.

During the layer's compression phase, the activation function ReLU is replaced with an activation function with a linear form since the former's output for negative inputs is zero, potentially destroying information during the conversion from high to low dimensions. Ultimately, the two branches are linked together, and channel shuffling is referred to in Figure 7 is used to allow data exchange between the two branches.

Hunger games search and remora optimization algorithm (HGS-ROA)

Enhancing the detection of intruders in IoT wireless networks is the primary objective of implementing the combination of the HGS-ROA method. A Remora Optimisation Algorithm (ROA) is used to select model features as

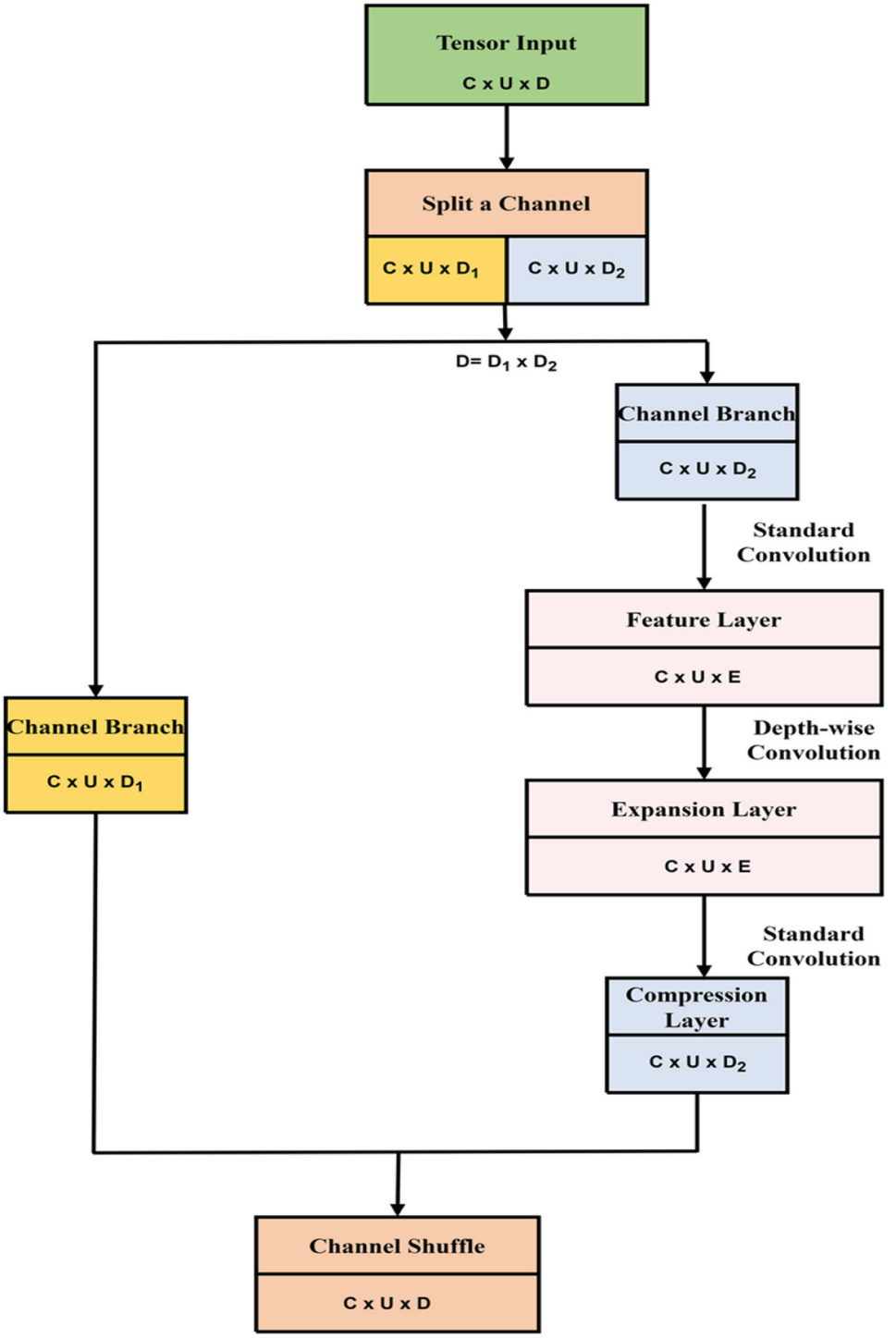

Figure 6. Lightweight unit B architecture.

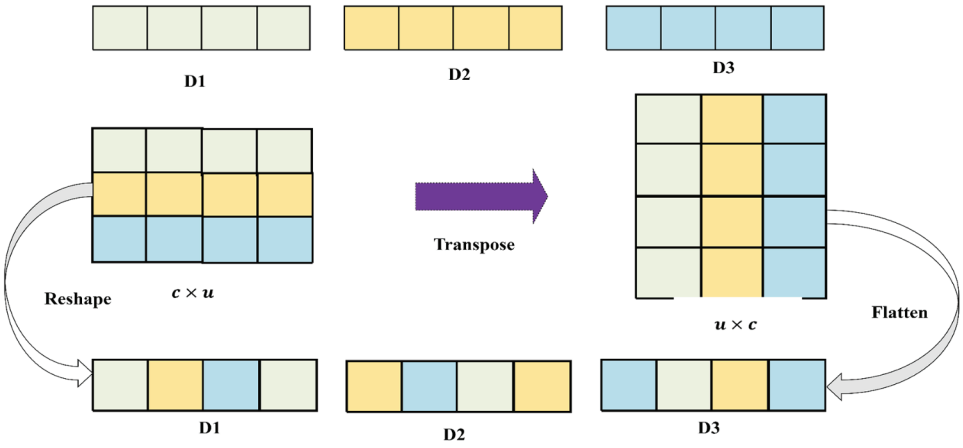

Figure 7. Channel shuffle.

efficiently as feasible. ROA uses two distinct mechanisms to update data inside its framework. Remora first leverages its prior knowledge to update a particular host via the global update mechanism in an attempt to get important resources. Remora navigates the host using the Local Update technique without altering any particular hosts. As a result, certain components do not change when optimization occurs. These two techniques increase ROA's adaptability and efficiency in challenging optimization scenarios. ROA needs to take three steps to select informative traffic features. First, initialization considers the position of the Remora candidate solution in the search space. For every position vector, remora's location P might be different. The following equation represents the current location.

$$X_r = (X_{r1}, X_{r2}, \cdots, X_{rd}) \tag{15}$$

In this case, X stands for location, d for search dimension, and r for number of remora. Every potential solution in ROA has to use the following formula to determine the fitness function.

$$F(X_r) = F(X_{r1}, X_{r2}, \cdots, X_{rd}) \tag{16}$$

Where F is utilized to confirm the associated candidate's fitness rating before choosing training traffic from an IoT wireless network. To speed up search and facilitate the selection of traffic features. ROA creates a main search area for every potential solution. To efficiently select features in the search space, each candidate solution has a subset of permitted features. By taking into account one and zero bits, ROA verifies that each candidate population can provide a traffic system. A fitness value is used to assess the feature subset, as shown in Figure 8. It shows that the characteristics that are chosen are represented by a single value bit, whereas the features that are not selected are

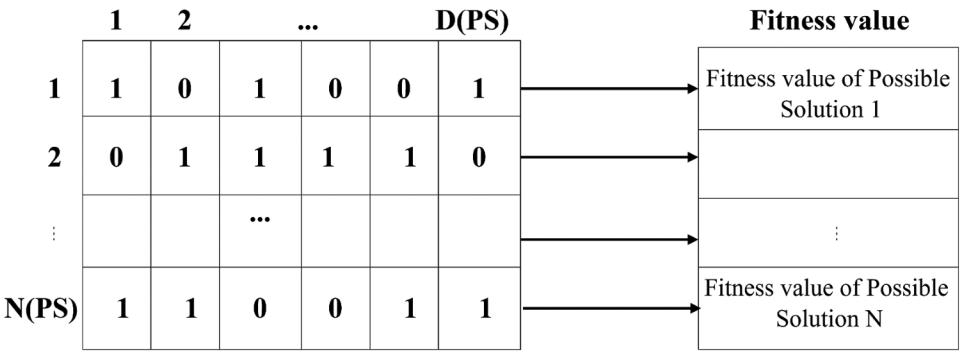

Figure 8. Fitness value of HGS-ROA.

represented by a zero value bit. The bit value serves as the basis for calculating the fitness value of each candidate solution M.

Where D(PS) denotes the possible solution dimension, and N(PS) denotes the number of possible outcomes. ROA is a continuous function that produces the first candidate solution for high-quality candidate solutions. In addition, a hunger game search technique is integrated to create a new population of potential attributes, from which the most informative traffic features are selected. The Hybrid Hunger Games Search (HGS) is used to evaluate the level of selected variables and update the position using a normal distribution. The fitness value is calculated using HGS-ROA to take advantage of the quality of the chosen subgroup feature. The locations of the ROA are recalculated and updated by first using the ROA model to initialize the positions transferred to the HGS algorithm's optimization phase. The HGS, then, is used to characterize the behaviors of traffic activities. In the IoT wireless network, HGS-ROA determines the optimal fitness score for choosing informative traffic features. The speed at which features are selected to update the positions of each solution through the subsequent issue is accelerated by the inclusion of HGS.

$$M_r^{T+1} = M_{best}^T - \left(Rand(0,1) * \left(\frac{M_{best}^T + M_{Rand}^T}{2} \right) - M_{Rand}^T \right) \tag{17}$$

Assuming that T is the number of this present iteration, M_{Rand} represents the random location. The ideal position is represented by M_{best}. Utilizing the HGS-ROA technique to get the characteristics, the features used to identify an intrusion into an IoT wireless network rely on the traffic, including both benign and malevolent activities.

Fourth: Network traffic analysis

For effective network routing in dynamic environments, routing table updates, route metrics, and network topology modifications are essential data. Route

metrics identify the fastest and most efficient routes based on distance and speed, while routing table updates assess the most recent pathways. Network topology adjustments maximize the connections and structure of the network. Examining these components offers insightful information that may be used to support the accuracy and performance of IDS models. The Routing Information Protocol (RIP) is used to examine network traffic and determine the most cost-effective paths for network packets. RIP enhances path selection and stabilizes the network by serving as the hub for routers to exchange routing information. The continuous flow of network traffic analysis is necessary to keep the system adaptable and to foster a strong, responsive network design.

A great deal of networks employ RIP, a distance vector type of routing protocol that is extensively used since it uses an easy-to-implement method. A hop count is the total number of devices that provide a data path from a source to a destination. The optimal path from the starting point to the target is determined by RIP using a statistic called number hop count (distance).

Some sophisticated attacks involve unusual network activity and traffic patterns that can be identified during detection. Network topology changes are tracked, route metrics are monitored, and routing table modifications are gathered and recorded by the IDS models. It finds traffic abnormalities, establishes a baseline from past RIP data, and links these anomalies with additional network information. The impacted routes, suspected botnet nodes, and the anomaly's timing are all included in the warning that the IDS subsequently issues.

Experimental results

We have implemented the proposed methodology by conducting an experimentation analysis to evaluate the system performance metrics. Our system design is based on a distinct simulation environment and data derived from the routing protocol, distinguishing it from previous systems that rely on full datasets. Although the methodologies I selected for comparison have different algorithms and system architectures, the final valuation results of performance metrics may still be compared. While doing so, we can successfully illustrate the benefits and effectiveness of our simulation-based architecture in properly identifying abnormalities in the network.

The experimental work involves several steps:

Simulation setup

To simulate the proposed research method, Python 3.9.6 runs on Ubuntu 16.006 operating system with a machine of 16 RAM and 1 TB storage. The IoT network has been simulated using NS 3.26. Initially, the testing context

involves 50 IoT Devices, 4 Gateway, 1 Router, and 1 Cloud. Dependable communication pathways use a "Tree-based Spider-Net Multipath" routing protocol, which has been used to provide effective energy consumption, fault tolerance, and load balancing.

Comparative analysis

This section compares the proposed approach to many existing ones, such as Hybrid Decision Tree Method (HIDT) (Mishra et al. 2022), Federated Machine Learning (FML) (Gebermariam et al. 2023), and Exponential-Henry Gas Solubility Optimization (EHGSO) (Ninu 2023) and evaluates its efficacy in performance measurements such as Authentication rate, Accuracy, Throughput, Packet Delivery Ratio, Attack detection rate, Delay, precision, recall, and F1-Score.

Authentication rate

The authentication rate can be determined by taking into account a number of factors, such as the sum of authentication based on frequency attempts, the number of authentication-based successive rates, and a possible number of active users.

$$Authentication\ Rate = \frac{\mathcal{F}_u + \mathcal{R}_u + \mathcal{G}_u}{\mathcal{U}_u} \tag{18}$$

where $\mathcal{F}_u$ denotes an authentication based on frequency attempts, $\mathcal{R}_u$ denotes the number of users, $\mathcal{G}_u$ denotes an authentication based on successive rate and $\mathcal{U}_u$ denotes a possible number of active users. A comparison of the authenticity rates between the proposed approach and related research in the literature is shown in Table 2. It is obvious that our method archives higher rates in all experiments of different numbers of users, as shown in Figure 9.

The performance determined by FML and HIDT algorithms is compared to the proposed method on authentication rate. The existing FML has achieved 230 bytes and 250 bytes for HID. It is relatively easy to compute and handle the enhanced authentication rate that may be achieved by registering every user. It is evident that our method outperforms the two studies using different numbers of users.

Accuracy

The calculation of accuracy involves dividing the total number of samples by the sum of the true positive and true negative. The accuracy can be represented as,

Table 2. Numerical outcomes of authentication rate.

Number of users (x-axis)	Authentication rate (bytes)-(y-axis)		
	FML	HIDT	Proposed
10	130	155	180
20	160	180	210
30	190	220	240
40	220	235	270
50	230	250	300

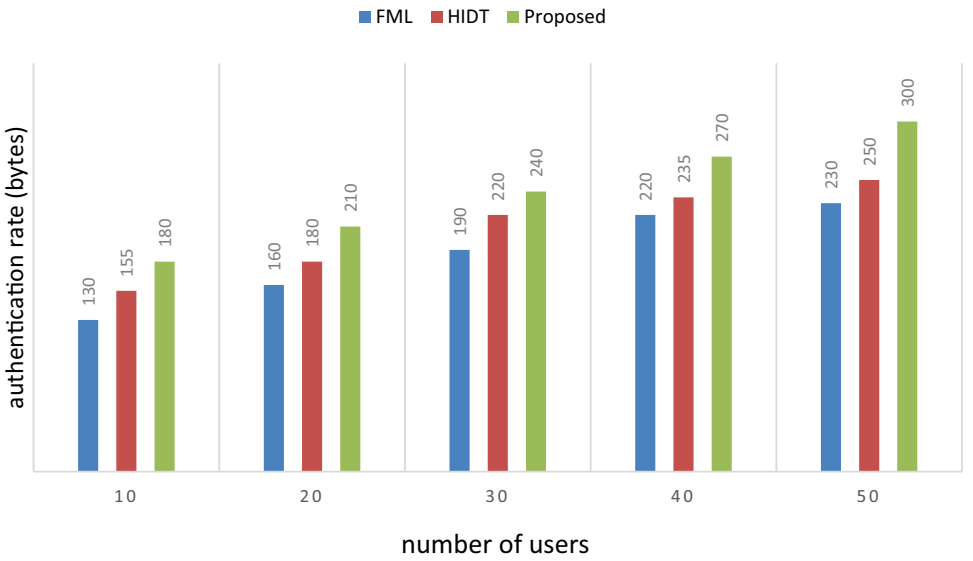

Figure 9. Number of users vs authentication rate (bytes).

$$Accuracy = \frac{\mathcal{T}_p + \mathcal{T}_n}{\mathcal{T}_p + \mathcal{T}_n + \mathcal{F}_p + \mathcal{F}_n} \tag{19}$$

where $\mathcal{T}_p$ denotes a true positive, $\mathcal{T}_n$ denotes a true negative, $\mathcal{F}_p$ denotes a false positive, and $\mathcal{F}_n$ denotes a false negative.

As illustrated in Table 3 and Figure 10, the current methods achieved good accuracy rates, with FML reaching 90% and HIDT attaining 95%. However, our proposed approach outperforms them all, achieving a remarkable accuracy rate of 99%.

Precision

A precision can be determined as the proportion of true positives to the total of false positives and true positives.

$$Precision = \frac{\mathcal{T}_p}{\mathcal{T}_p + \mathcal{F}_p} \tag{20}$$

Table 3. Numerical outcomes of accuracy.

Number of IOT devices (x-axis)	Accuracy(%) - (y-axis)		
	FML	HIDT	Proposed
10	60	67	76
20	70	75	83
30	78	85	90
40	82	89	94
50	90	95	99

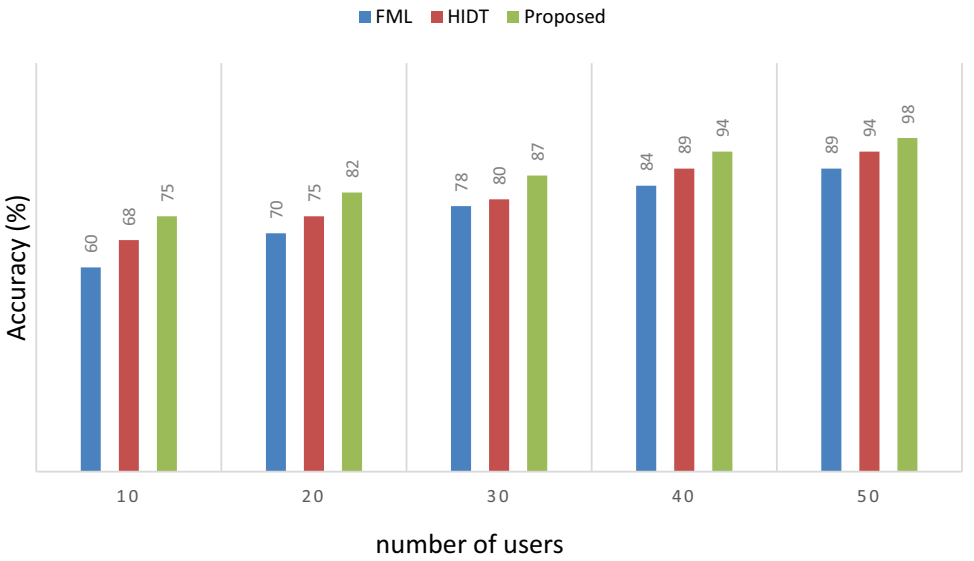

Figure 10. Number of IOT devices vs accuracy (%).

Based on our experiments, we have improved the system performance in terms of precision compared with current mechanisms, which can reach a precision of 89% for FML and 94% for HIDT. Table 4 and Figure 11 show how our method may improve the system outcomes by employing precession rates.

Throughput

The channel-based transfer of multiple data packets divided by the total number of packets during the given period of time yields the throughput. It represents an efficient processing of the system.

$$Throughput = \sum_{i=1}^{n} (\mathcal{R}_{\mathcal{B}} * \mathcal{B}_{\mathcal{G}}) / \mathcal{G}_{\mathcal{T}} \tag{21}$$

Where $\mathcal{R}_{\mathcal{B}}$ denotes the receiving data packet, $\mathcal{B}_{\mathcal{G}}$ denotes the size of the data packet, and $\mathcal{G}_{\mathcal{T}}$ denotes the total time that can be taken to simulate a process.

Table 5 and Figure 12 demonstrates the comparison of the throughput rates of EHGSO and FML models compared to our work. High rates of throughput rates can support high scalability to utilize resources for

Table 4. Numerical outcomes of precision.

Number of IOT devices (x-axis)	Precision(%) - (y-axis)		
	FML	HIDT	Proposed
10	60	68	75
20	70	75	82
30	78	80	87
40	84	89	94
50	89	94	98

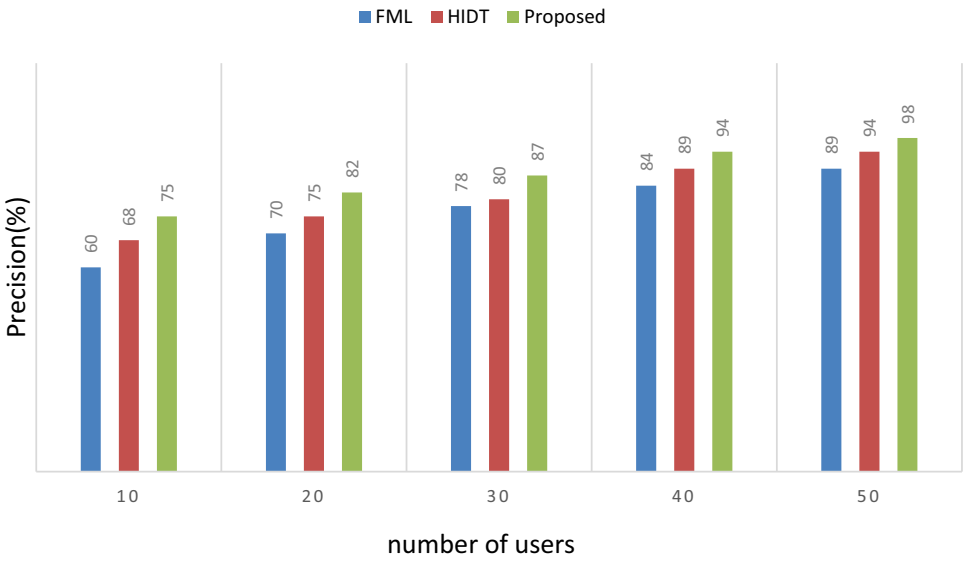

Figure 11. Number of IOT devices vs precision (%).

effective processing. It is obvious that the proposed method can achieve a higher throughput of 7 kbps, while it reaches 4 for EHGSO and 5.2 for HIDT.

Packet delivery ratio (PDR)

A packet delivery ratio can be determined as the ratio of correctly detected packets and delivered packets to the total number of data packets sent.

Table 5. Numerical outcomes of throughput.

Number of IOT devices (x-axis)	Throughput(kbps) - (y-axis)		
	EHGSO	HIDT	Proposed
10	1.5	2.9	3.5
20	2.9	3.8	4.5
30	3.5	4	5.8
40	3.8	4.5	6
50	4	5.2	7

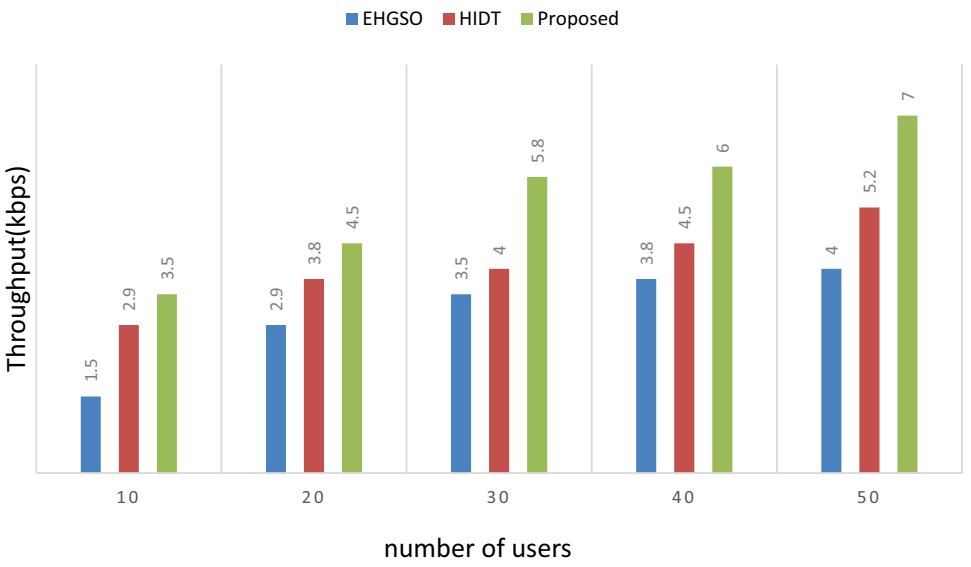

Figure 12. Number of IOT devices vs throughput (kbps).

$$Packet\ Delivery\ Ratio = \frac{\mathcal{CDB} + \mathcal{DB}}{\mathcal{TBDG}} \tag{22}$$

Where $\mathcal{CDB}$ denotes the correctly detected packet, $\mathcal{DB}$ denotes the delivered packet and $\mathcal{TBDG}$ denotes the successfully delivered data packet.

A packet delivery ratio can be measured as a performance metric. We have accomplished a bit higher rates for PDR to reach 92%. However, the PDR rate is 85% for EHGSO and 88% for HIDT as illustrated in Table 6 and Figure 13.

Attack detection rate (ADR)

The attack detection rate can be determined as a rising proportion of identified attacks to total users. It can be represented as,

$$\mathcal{D}_{rate} = \frac{\mathcal{A}_{detection}}{\mathcal{I}_{users}} \tag{23}$$

Where $\mathfrak{D}_{\mathfrak{rate}}$ denotes the rate of attack detection, $\mathcal{A}_{detection}$ denotes a detected attack, and $\mathcal{I}_{users}$ denotes an increased number of users.

The attack detection rate is the most important metrics to evaluate the performance of any IDS model. This rate can be measured using the IoT simulated network to evaluate the proposed method of detection. There has been a demonstrated higher rate of 98% as opposed to 91% for FML and 95% for HIDT as shown in Table 7 and Figure 14.

Table 6. Numerical outcomes of packet delivery ratio.

Number of IOT devices (x-axis)	PDR(%) - (y-axis)		
	EHGSO	HIDT	Proposed
10	57	63	70
20	63	69	76
30	70	77	82
40	79	84	87
50	85	88	92

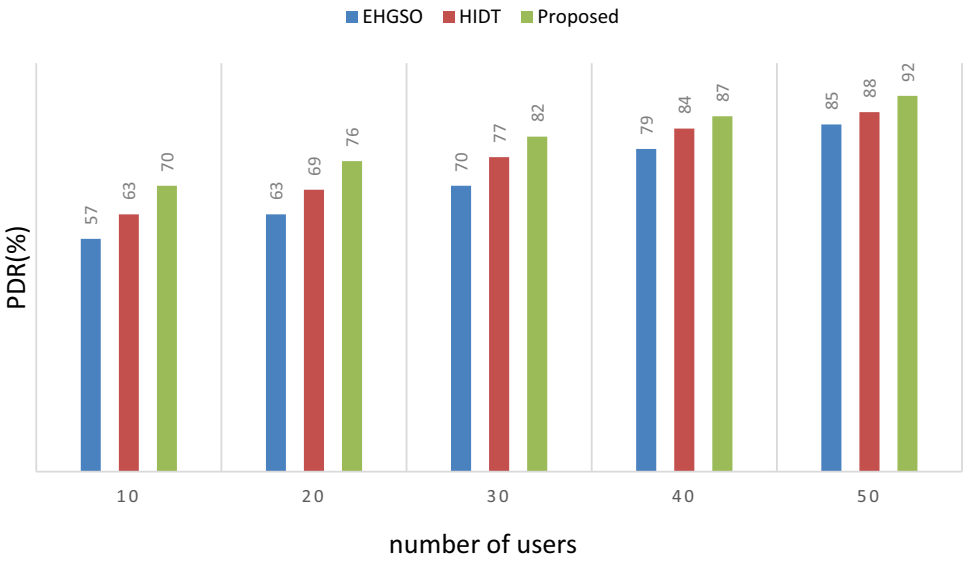

Figure 13. Number of IOT devices vs packet delivery ratio (%).

Delay

A delay can be determined as the time consumption between the source and destination of a packet across the network. A delay can be considerably decreased and slightly increased number of nodes.

$$Delay = \frac{\sum_{i=1}^{n} (\mathcal{RBT} - \mathcal{SBT}) * 1000\ ms}{\mathcal{IBDG}} \tag{24}$$

Where $\mathcal{RBT}$ denotes a timer of the data packet, $\mathcal{SBT}$ denotes a data packet sent timer and $\mathcal{IBDG}$ denotes a successfully delivering a complete data packet.

The delay has been measured based on the simulated network, and it is illustrated in Table 8 and Figure 15. A delay of 0.3 s has been experienced by the proposed method, and on the other hand, the delay of FML was 0.3 s and 0.9 s in HIDT system.

Recall

A recall can be defined as the ratio of the true positive to the summation of the false negative and true positive.

Table 7. Numerical outcomes of attack detection rate.

Number of IOT devices (x-axis)	Attack Detection Rate(%) - (y-axis)		
	FML	HIDT	Proposed
10	60	65	74
20	72	76	81
30	78	83	89
40	83	87	93
50	91	95	98

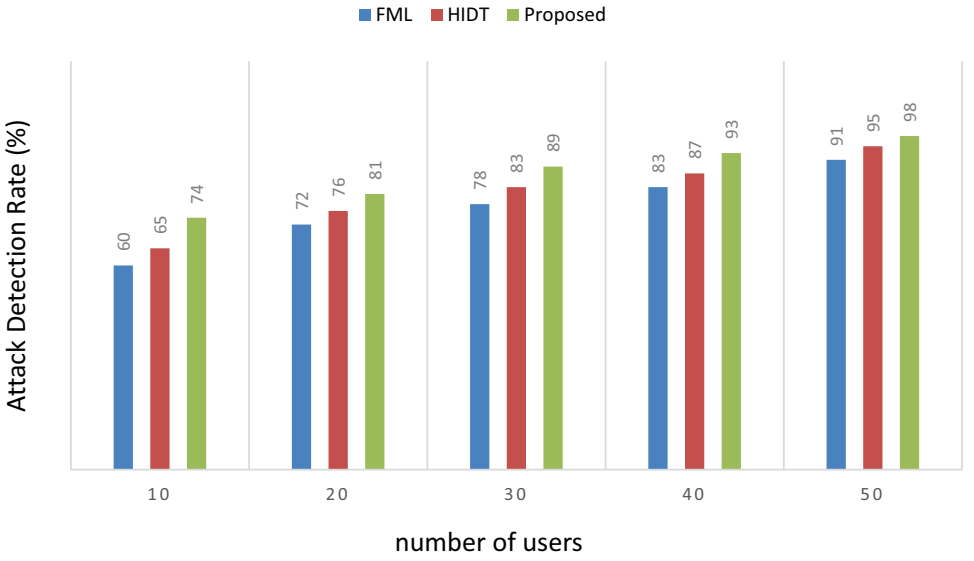

Figure 14. Number of IOT devices vs attack detection rate (%).

$$Recall = \frac{\mathcal{T}_p}{\mathcal{T}_p + \mathcal{F}_n} \tag{25}$$

The recall rate of 97% has been measured using our method, while FML has achieved 88% and 92% for HIDT model. A comparison of the observed models is shown in Table 9 and Figure 16.

Table 8. Numerical outcomes of delay.

Number of IOT devices (x-axis)	Delay (sec) - (y-axis)		
	FML	HIDT	Proposed
10	4.5	3.9	3
20	3.4	2.7	2.2
30	2.4	2	1.6
40	1.6	1.2	0.8
50	0.9	0.6	0.3

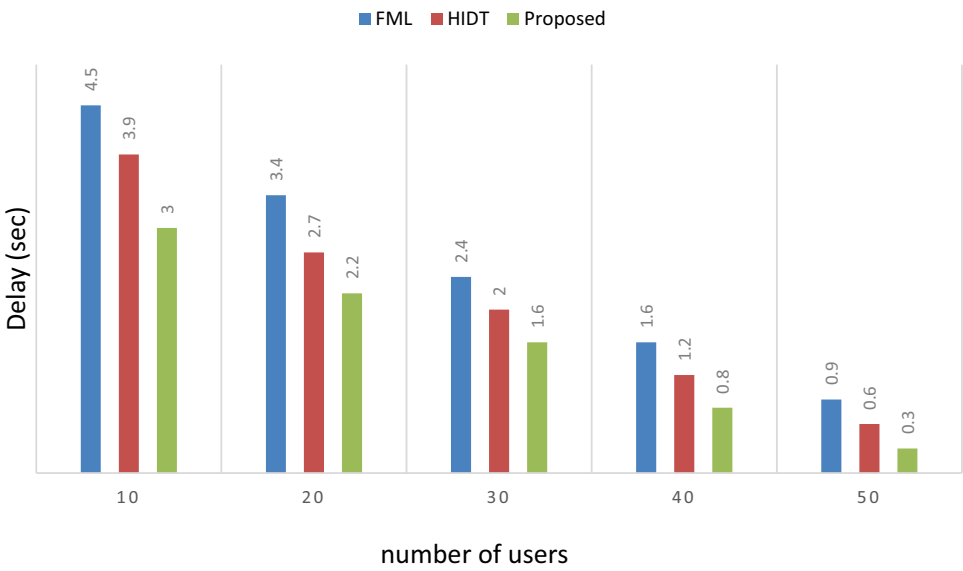

Figure 15. Number of IOT devices vs delay (sec).

F1-score

An F1-Score can be determined as the ratio of recalled products to the summation of precision and recall.

$$F1 - Score = 2 \times \frac{precision + recall}{precision \times recall} \tag{26}$$

Performance is also evaluated using the F1-Score, as illustrated in Table 10 and Figure 17. The suggested system outperforms the previous two models, reaching a 96% rate.

Research experiments summary

In summary, in this research, we have implemented a simulated experimental setup of 50 nodes. "Tree-based Spider-Net Multipath" has been selected to offer power efficiency and adaptability in response to changing network circumstances. AES is used for credential verification and user registration. Robust encryption guarantees security and minimizes computational complexity. Integration of

Table 9. Numerical outcomes of recall.

Number of IOT devices (x-axis)	Recall (%) - (y-axis)		
	FML	HIDT	Proposed
10	58	68	75
20	68	74	80
30	76	80	86
40	82	87	93
50	88	92	97

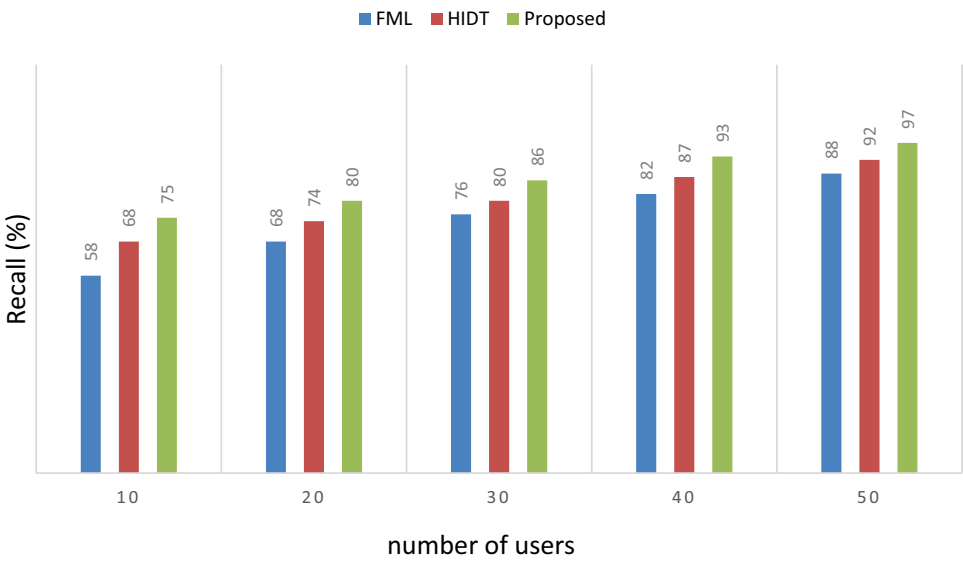

Figure 16. Number of IOT devices vs recall (%).

Table 10. Numerical outcomes of F1-score.

Number of IOT devices (x-axis)	Recall (sec) - (y-axis)		
	FML	HIDT	Proposed
10	60	69	74
20	69	75	80
30	71	78	85
40	79	85	91
50	85	90	96

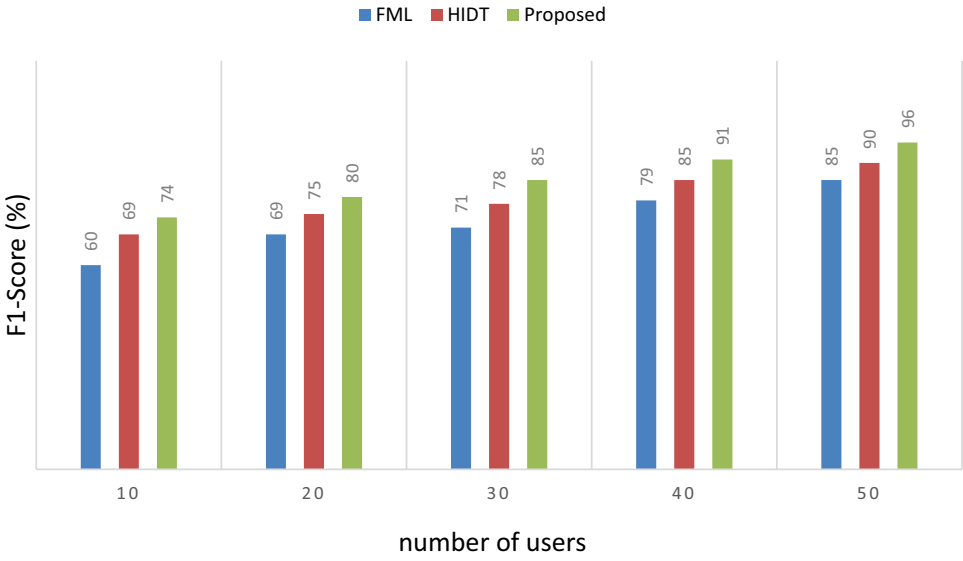

Figure 17. Number of IOT devices vs F1-score (%).

DRL-TS to select users according to network conditions, energy levels, and trust. Proximal Policy Optimization for managing a broader action space for channel selection. Routing Information Protocol (RIP) protocol has been utilized as a local routing algorithm. Network traffic data has been collected using the tracing method to capture the Packet data and then encrypt it by AES. Then, we have done the decryption for the traffic data and, hence, we have extracted relevant features for Deep Learning analysis. An LDNN is trained, and HGS-ROA is used for optimization and dynamic adaptation to new attacks. Therefore, RIP has been integrated into the Deep Learning model to obtain informative features. The evaluation metrics involves Authentication rate, Accuracy, Throughput, Packet Delivery Ratio, Attack detection rate, Delay, precision, recall, and F1-Score.

Conclusion

DL method is utilized for intrusion detection to improve cybersecurity in networks with encrypted traffic. The network construction uses tree-based spidernet multipath and authentication using a symmetric encryption-based AES algorithm, and network traffic analysis is done using the DPI technique. Efficient user and channel selection is done using Deep Reinforcement Learning with Tabu Search and the proximal policy optimization model. A lightweight Deep Neural Network with the Hunger Games Search and Remora Optimization Algorithm (HGS-ROA) and we have integrated RIP data into the deep learning model to enhance the effectiveness of the intrusion detection model. The suggested technique is evaluated by comparing the authentication rate of 300 bytes, accuracy of 99%, throughput of 7kbps, delay of 0.3 sec, recall of 97%, packet delivery ratio of 92%, attack detection rate of 98%, F1-score of 96% and precision in 98% of the proposed methods with those of the current methods. Our solution performs better than all other methods currently in use, according to the numerical analysis, in every metric. For future work, the detection facility of malicious encrypted traffic can be enhanced by expanding the datasets to include various attack activities and integrating real-time data streams. Feature engineering may also be refined to incorporate more useful qualities, resulting in a more consistent model. Furthermore, other ML and DL models with different capabilities can be involved to improve the accuracy and responsiveness.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was funded by the Deanship of Scientific Research at Jouf University through the Fast-track Research Funding Program.

ORCID

Faeiz Alserhani http://orcid.org/0000-0002-0161-7147

References

Ahmad, J., M. U. Zia, I. H. Naqvi, J. N. Chattha, F. A. Butt, T. Huang, and W. Xiang. 2024. Machine learning and blockchain technologies for cybersecurity in connected vehicles. *WIREs Data Mining and Knowledge Discovery* 14 (1):e1515. doi:10.1002/widm.1515.

Al-Bakhat, L., and S. Almuhammadi. 2022. Intrusion detection on quic traffic: A machine learning approach. *2022 7th International Conference on Data Science and Machine Learning Applications (CDMA)*, 194–99. IEEE. doi:10.1109/CDMA54072.2022.9752695.

Alrayes, F. S., M. Zakariah, M. Driss, and W. Boulila. 2023. Deep neural decision forest (DNDF): A novel approach for enhancing intrusion detection systems in network traffic analysis. *Sensors* 23 (20):8362. doi:10.3390/s23208362.

Alwasel, B., A. Aldribi, M. Alreshoodi, I. S. Alsukayti, and M. Alsuhaibani. 2023. Leveraging graph-based representations to enhance machine learning performance in IIoT network security and attack detection. *Applied Sciences* 13 (13):7774. doi:10.3390/app13137774.

Bakhsh, S. A., M. A. Khan, F. Ahmed, M. S. Alshehri, H. Ali, and J. Ahmad. 2023. Enhancing IoT network security through deep learning-powered intrusion detection system. *Internet of Things* 24:100936. doi:10.1016/j.iot.2023.100936.

Chuang, H. M., and L. J. Ye. 2023. Applying transfer learning approaches for intrusion detection in software-defined networking. *Sustainability* 15 (12):9395. doi:10.3390/su15129395.

Djenna, A., E. Barka, A. Benchikh, and K. Khadir. 2023. Unmasking cybercrime with artificial-intelligence-driven cybersecurity analytics. *Sensors* 23 (14):6302. doi:10.3390/s23146302.

Faragallah, O. S., W. El-Shafai, A. I. Sallam, I. Elashry, E. S. M. EL-Rabaie, A. Afifi, and H. S. El-Sayed. 2022. Cybersecurity framework of hybrid watermarking and selective encryption for secure HEVC. *Multimedia Tools & Applications* 81 (8):11577–606. doi:10.1007/s11042-022-12389-4.

Fu, C., Q. Li, and K. Xu. 2023. Detecting unknown encrypted malicious traffic in real time via flow interaction graph analysis. *arXiv preprint arXiv*:2301.13686. doi:10.48550/arXiv.2301.13686.

Gazzan, M., and F. T. Sheldon. 2023. An enhanced minimax loss function technique in generative adversarial network for ransomware behavior prediction. *Future Internet* 15 (10):318. doi:10.3390/fi15100318.

Gebremariam, G. G., J. Panda and S. Indu. 2023. Blockchain-based secure localization against malicious nodes in iot-based wireless sensor networks using federated learning. *Wireless Communications and Mobile Computing* 2023 (1):8068038.

Gupta, L., T. Salman, A. Ghubaish, D. Unal, A. K. Al-Ali, and R. Jain. 2022. Cybersecurity of multi-cloud healthcare systems: A hierarchical deep learning approach. *Applied Soft Computing* 118:108439. doi:10.1016/j.asoc.2022.108439.

Hsiao, S. J., and W. T. Sung. 2022. Enhancing cybersecurity using blockchain technology based on IoT data fusion. *IEEE Internet of Things Journal* 10 (1):486–98. doi:10.1109/JIOT.2022.3199735.

Ilca, L. F., O. P. Lucian, and T. C. Balan. 2023. Enhancing cyber-resilience for small and medium-sized organizations with prescriptive malware analysis, detection and response. *Sensors* 23 (15):6757. doi:10.3390/s23156757.

Ilyas, M. U., and S. A. Alharbi. 2022. Machine learning approaches to network intrusion detection for contemporary internet traffic. *Computing* 104 (5):1061–76. doi:10.1007/s00607-021-01050-5.

Jorgensen, S., J. Holodnak, J. Dempsey, K. de Souza, A. Raghunath, V. Rivet, N. DeMoes, A. Alejos, and A. Wollaber. 2023. Extensible machine learning for encrypted network traffic application labeling via uncertainty quantification. *IEEE Transactions on Artificial Intelligence* 5 (1):420–33. doi:10.1109/TAI.2023.3244168.

Kadakia, Y. A., A. Suryavanshi, A. Alnajdi, F. Abdullah, and P. D. Christofides. 2024. Integrating machine learning detection and encrypted control for enhanced cybersecurity of nonlinear processes. *Computers & Chemical Engineering* 180:108498. doi:10.1016/j.compchemeng.2023.108498.

Khan, L., H. T. Ullah, T. Ali, A. Ali, M. A. A. Mamun, and N. Kumar. 2023. A secure and efficient federated learning framework for IIoT using blockchain and reinforcement learning. *IEEE Transactions on Industrial Informatics* 19 (5):4797–805. doi:10.1109/TII.2022.3198618.

Latif, S., W. Boulila, A. Koubaa, Z. Zou, and J. Ahmad. 2024. DTL-IDS: An optimized intrusion detection framework using deep transfer learning and genetic algorithm. *Journal of Network and Computer Applications* 221:103784. doi:10.1016/j.jnca.2023.103784.

Lin, P., K. Ye, Y. Hu, Y. Lin, and C. Z. Xu. 2022. A novel multimodal deep learning framework for encrypted traffic classification. *IEEE/ACM Transactions on Networking* 31 (3):1369–84. doi:10.1109/TNET.2022.3215507.

Lo, W., H. Alqahtani, K. Thakur, A. Almadhor, S. Chander, and G. Kumar. 2022. A hybrid deep learning based intrusion detection system using spatial-temporal representation of In-vehicle network traffic. *Vehicular Communications* 35:100471. doi:10.1016/j.vehcom.2022.100471.

Meddeb, R., F. Jemili, B. Triki, and O. Korbaa. 2023. A deep learning-based intrusion detection approach for mobile ad-hoc network. *Soft Computing* 1–15. doi:10.1007/s00500-023-07974-0 .

Mishra, A., Y. I. Alzoubi, M. J. Anwar, and A. Q. Gill. 2022. Attributes impacting cybersecurity policy development: An evidence from seven nations. *Computers & Security* 120:102820. doi:10.1016/j.cose.2022.102820.

Miyamoto, M., K. Teranishi, K. Emura, and K. Kogiso. 2023. Cybersecurity-enhanced encrypted control system using keyed-homomorphic public key encryption. *Institute of Electrical and Electronics Engineers Access* 11:45749–60. doi:10.1109/ACCESS.2023.3274691.

Mohamed, A., F. Wang, I. Butun, J. Qadir, R. Lagerström, P. Gastaldo, and D. D. Caviglia. 2022. Enhancing cyber security of LoRaWAN gateways under adversarial attacks. *Sensors* 22 (9):3498. doi:10.3390/s22093498.

Moustafa, N., and J. Slay. 2015. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). *2015 Military Communications and Information Systems Conference (MilCIS)*, 1–6. IEEE. doi:10.1109/MilCIS.2015.7348942.

Ninu, S. B. 2023. An intrusion detection system using exponential henry gas solubility optimization based deep neuro fuzzy network in MANET. *Engineering Applications of Artificial Intelligence* 123:105969.

Papanikolaou, A., A. Alevizopoulos, C. Ilioudis, K. Demertzis, and K. Rantos. 2023. An AutoML network traffic analyzer for cyber threat detection. *International Journal of Information Security* 22 (5):1511–30. doi:10.1007/s10207-023-00703-0.

Pradeepthi, C., and B. U. Maheswari. 2023. Network intrusion detection and prevention strategy with data encryption using hybrid detection classifier. *Multimedia Tools & Applications* 83 (13):40147–78. doi:10.1007/s11042-023-16853-1.

Rezaei, S., and X. Liu. 2019. Deep learning for encrypted traffic classification: An overview. *IEEE Communications Magazine* 57 (5):76–81. doi:10.1109/MCOM.2019.1800819.

Salvakkam, D. B., V. Saravanan, P. K. Jain, and R. Pamula. 2023. Enhanced quantum-secure ensemble intrusion detection techniques for cloud based on deep learning. *Cognitive Computation* 1–20. doi:10.1007/s12559-023-10139-2.

Sharma, B., L. Sharma, C. Lal, and S. Roy. 2023. Anomaly based network intrusion detection for IoT attacks using deep learning technique. *Computers & Electrical Engineering* 107:108626. doi:10.1016/j.compeleceng.2023.108626.

Sood, T., S. Prakash, S. Sharma, A. Singh, and H. Choubey. 2022. Intrusion detection system in wireless sensor network using conditional generative adversarial network. *Wireless Personal Communications* 126 (1):911–31. doi:10.1007/s11277-022-09776-x.

Ullah, F., S. Ullah, G. Srivastava, and J. C. W. Lin. 2023. IDS-INT: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic. *Digital Communications and Networks* 10 (1):190–204. doi:10.1016/j.dcan.2023.03.008.

Wang, Z., P. Wang, and Z. Sun. 2022. SDN traffic anomaly detection method based on convolutional autoencoder and federated learning. *GLOBECOM 2022-2022 IEEE Global Communications Conference*, 4154–60, IEEE. doi:10.1109/GLOBECOM48099.2022.10002485.

Xu, H., L. Sun, G. Fan, W. Li, and G. Kuang. 2023. A hierarchical intrusion detection model combining multiple deep learning models with attention mechanism. *Institute of Electrical and Electronics Engineers Access* 11:66212–26. doi:10.1109/ACCESS.2023.3290613.

Yan, R., Y. Wang, J. Dai, Y. Xu, and A. Q. Liu. 2022. Quantum-key-distribution-based microgrid control for cybersecurity enhancement. *IEEE Transactions on Industry Applications* 58 (3):3076–86. doi:10.1109/TIA.2022.3159314.

Zanasi, C., S. Russo, and M. Colajanni. 2024. Flexible zero trust architecture for the cybersecurity of industrial IoT infrastructures. *Ad Hoc Networks* 156:103414. doi:10.1016/j.adhoc.2024.103414.