sensors

MDPI

Article

A Novel IDS with a Dynamic Access Control Algorithm to Detect and Defend Intrusion at IoT Nodes

Moutaz Alazab [1,2,*], Albara Awajan [1], Hadeel Alazzam[1], Mohammad Wedyan [3], Bandar Alshawi [4] and Ryan Alturki [5]

1 Department of Intelligent Systems, Faculty of Artificial Intelligence, Al-Balqa Applied University, Al-Salt 19385, Jordan; a.awajan@bau.edu.jo (A.A.); hadeel.alazzam@bau.edu.jo (H.A.)
2 Cybersecurity Department, School of Computing and Data Sciences, Oryx Universal College with Liverpool John Moores University, Doha 34110, Qatar
3 Department of Computer Sciences, Faculty of Information Technology and Computer Sciences, Yarmouk University (YU), Irbid 21163, Jordan; mwedyan@yu.edu.jo
4 Department of Computer and Network Engineering, College of Computing, Umm Al-Qura University, Makkah 24382, Saudi Arabia; bmhshawi@uqu.edu.sa
5 Department of Software Engineering, College of Computing, Umm Al-Qura University, Makkah 24382, Saudi Arabia; rmturki@uqu.edu.sa
* Correspondence: moutaz.a@oryx.edu.qa or m.alazab@bau.edu.jo

Abstract: The Internet of Things (IoT) is the underlying technology that has enabled connecting daily apparatus to the Internet and enjoying the facilities of smart services. IoT marketing is experiencing an impressive 16.7% growth rate and is a nearly USD 300.3 billion market. These eye-catching figures have made it an attractive playground for cybercriminals. IoT devices are built using resource-constrained architecture to offer compact sizes and competitive prices. As a result, integrating sophisticated cybersecurity features is beyond the scope of the computational capabilities of IoT. All of these have contributed to a surge in IoT intrusion. This paper presents an LSTM-based Intrusion Detection System (IDS) with a Dynamic Access Control (DAC) algorithm that not only detects but also defends against intrusion. This novel approach has achieved an impressive 97.16% validation accuracy. Unlike most of the IDSs, the model of the proposed IDS has been selected and optimized through mathematical analysis. Additionally, it boasts the ability to identify a wider range of threats (14 to be exact) compared to other IDS solutions, translating to enhanced security. Furthermore, it has been fine-tuned to strike a balance between accurately flagging threats and minimizing false alarms. Its impressive performance metrics (precision, recall, and F1 score all hovering around 97%) showcase the potential of this innovative IDS to elevate IoT security. The proposed IDS boasts an impressive detection rate, exceeding 98%. This high accuracy instills confidence in its reliability. Furthermore, its lightning-fast response time, averaging under 1.2 s, positions it among the fastest intrusion detection systems available.

Keywords: Intrusion Detection System; LSTM; deep learning; detection rate; optimization; Dynamic Access Control

check for updates

Citation: Alazab, M.; Awajan, A.; Alazzam, H.; Wedyan, M.; Alshawi, B.; Alturki, R. A Novel IDS with a Dynamic Access Control Algorithm to Detect and Defend Intrusion at IoT Nodes. *Sensors* **2024**, *24*, 2188. https://doi.org/10.3390/s24072188

Academic Editors: Rongxing Lu and Antonio J. Jara

Received: 29 December 2023
Revised: 24 January 2024
Accepted: 26 February 2024
Published: 29 March 2024

1. Introduction

The Internet of Things (IoT) is the technology between machines and the internet, enabling multiple facilities, including ubiquitous control, monitoring, and smart services [1]. The advent of the 5G network and the availability of high-speed internet access has fueled the growth of this sector [2]. At the current growth rate of 16.7% [3], the net IoT market worth is expected to cross USD 300 billion [4]. However, this massive market suffers from security vulnerabilities. IoT devices are compact, low-cost, and consume little power. With these features in an IoT product, a company will lose market competition. These characteristics limit the scope of incorporating advanced cybersecurity features into

IoT devices [5]. The accelerated growth, the massive worth of the market, the reward of accessing interconnected nodes, and existing vulnerabilities make IoT an attractive ground for cybercriminals. This paper presents a practical IoT intrusion detection system that has been developed, keeping utility and usability at the center of the design. This innovative and practical IDS with novel features is a potential solution to IoT intrusions.

The methodology proposed in this paper has been developed to address the research gap discovered in the literature review presented in this paper. The effectiveness of the deep learning (DL) approaches in different papers testifies to the feasibility of developing a practical IDS [6]. In this endeavor, the first challenge is selecting an effective DL model, which is done in this paper through mathematical characteristics analysis [7]. Unlike most of the deep learning-based IDS research, this paper does not jump to any conclusions with the development of the technology only. An application model has been developed and presented in this paper to apply the proposed IDS. It has been achieved by developing an algorithm that dynamically controls access to IoT devices based on the network traffic level of maliciousness. It senses the malicious signals and blocks the malicious traffic source for a period of time, which depends on the number of intrusion attempts performed by the source. This unique feature makes the proposed IDS a practical solution to strengthen IoT security.

Furthermore, this paper observes IDS development from a utility and usability perspective. It is common for machine learning approaches to get some false positive (FP) predictions. Responding to every prediction and blocking the traffic based on false positive predictions undermine the utility offered by any IDS [8]. Moreover, it severely impacts the usability of the IDS. These product-centric characteristics are absent in most IDS research because of the lack of a proper application model. The proposed IDS introduces an innovative analysis approach that balances detection and false positive rates. Although this optimization technique marginally reduces the detection rate, it improves the user experience by reducing the service interruption caused by false alarms. As a result, the proposed IDS is an excellent solution from a utility and usability perspective.

This paper selects the DL network through mathematical analysis, which ensures better and more reliable performance with justification. Instead of using an existing network, this paper has presented an innovative approach to optimizing network architecture for intrusion detection. Integrating the Dynamic Access Control (DAC) algorithm is one of this paper's novel contributions, extending the capability of the proposed IDS from detection to defense. It dynamically blocks the malicious traffic source for periods of time, depending on the intrusion volume. Another novelty of this paper is identifying the optimization issue of false positive and detection rates and solving it through an appropriate threshold selection method. The unique features and novel contributions of this paper are listed below:

- Development of the novel Dynamic Access Control (DAC) algorithm for defending against further intrusion from the same source by blocking it for periods associated with the number of intrusions.
- Applying the unique concept of detection rate optimization to enhance the usability of intrusion detection systems by minimizing the false positive rate, which lowers the probability of service interruption for false alarms.
- Achieving 97.16% validation accuracy, with precision, recall, and F1 score of 97.07%, 97.11%, and 97.05%, respectively.
- Gaining a detection rate of 98.73% by maintaining an average detection time of 1.14 s, which is considered to be real-time detection.

The rest of the paper has been organized into six sections. Section 2 presents a review of the recent literature. The methodology proposed in this paper is meticulously detailed in Section 3. Section 4 outlines the specific implementation and response mechanisms of the proposed Intrusion Detection System (IDS). The findings from the experiments and an analysis of their effectiveness are presented in Section 5. Section 6 highlights the limitations of the paper and discusses potential avenues for the future direction of this research. Finally, Section 7 concludes the paper by summarizing it.

2. Literature Review

The study by A. Awajan [9] on DL application achieves 93.74% accuracy for five frequently occurring IoT intrusions. The fully connected (FC) network architecture presented in this paper is incapable of interpreting network sequential data, which is a significant disadvantage of this paper. D. Musleh et al. applied machine learning (ML) to extract features and detect intrusion from image data. While their performance is impressive, the practical application of it is limited because it considers image features only [10]. A similar ML-based study conducted by S. Alkadi et al. [11] analyzes six ML models on four datasets related to IoT intrusion. While this paper provides deep insights into ML applications in IDS research, it does not contribute to developing any practical solution. An interesting study by M. Alazab et al. [12] applies the Moth–Flame Optimizer (MFO) algorithm for intrusion detection. While this approach deserves appreciation for impressive performance, it does not discuss the effective application of the performance. Unlike these approaches, the proposed IDS has been developed from an application perspective. That is why detection and defending have received equal attention in this paper, which is absent in this literature.

R. Chaganti et al. [13] have proposed an IDS for a software-defined networking (SDN)-enabled IoT environment. The rigorous analysis of this paper is worth drawing attention to. However, this approach applies to four types of intrusions only. Moreover, it does not emphasize detection rate optimization. The study by A. Henry et al. [14] follows a similar approach of combining different models. However, it suffers from the same limitation of false alarm rate and detection rate optimization. Similar weaknesses are noticeable in the papers of A. Fatani et al. [15], M. Bacevicius et al. [16], and H. Alshahrani et al. [17]. This paper presents a unique approach to discovering the balance between intrusion detection and false positive rates. It enhances the usability of the IDS and makes it more effective. Similar to other papers, none of these approaches can ensure protection against 14 different types of intrusions.

The study of S. Alosaimi et al. [18] demonstrates an appreciable multi-layer security system using DL for IoT networks. However, it is confined to different types of DDoS attacks only. Moreover, because of using multiple layers, the system has taken on a complicated shape. Compared to this approach, the proposed IDS is simpler yet more efficient. It is a faster and more practical solution to IoT intrusion. The ensemble method presented by Y. Alotaibi et al. [19] deserves appreciation for combining four types of models. However, this effective system does not extend to developing a practical defense mechanism as proposed in this paper. The explainable AI-based approach of X. Larriva-Novo et al. [20] excellently presents the underlying intrusion detection processes. However, the mathematical analysis to justify the selected model is missing in the research discussed in the proposed paper.

The survey on intrusion detection using machine learning by S. V. N. Santhosh Kumar et al. [6], S. Fraihat et al. [21], and B. Kaur et al. [22] supports the hypothesis of this paper. The recent research in intrusion detection using IoT focuses on performance improvement while ignoring the development of application models. Detecting intrusion by training DL networks and applying the model to ensure security is not the same. The latter is more challenging, which has been addressed in the proposed paper. The false positive rate and interruption in IoT response are two major usability drawbacks of IDSs that have been studied and solved in the proposed paper. Moreover, the proposed paper introduces the DAC algorithm that extends the IDS's ability to defend against further intrusion from the same source. Because of these novelties and outstanding performance, the proposed LSTM-based IDS stands out from the rest.

Research Gap Analysis

The previous section's literature review points out the existing research's limitations. It further compares those limitations with the proposed paper. It demonstrates the existing gap in the current IDS-related research field and how the proposed methodology fills the gaps. These gaps have been listed in Table 1. The most common research gap in every paper reviewed is the direction of the practical application of their proposed methodology. The practical application refers to implementing the IDS in a real IoT network. Most papers developed the IDS and confined their study to intrusion prediction. Developing technology and practically applying it to solve an existing problem is not the same. Most of the papers presented the development of the IDS using various innovative approaches; however, they did not discuss the ways they can be implemented. It has been further observed that the binary classification is where malicious and benign network traffic is detected. This is another research gap. Apart from these two research gaps, some other significant research gaps are specific to different papers. These research gaps have been listed in Table 1 as Significant Weaknesses.

Table 1. The research gaps discovered in recent literature reviews.

Author(s)	Method Used	Significant Weaknesses	Practical Application	Detection Range
A. Awajan	Deep Learning	Incapable of interpreting network sequential data	No	5
D. Musleh	Machine Learning	Limited to image features only	No	2
S. Alkadi	Machine Learning	Does not contribute to developing any practical solution	No	4
M. Alazab	Moth–Flame Optimizer (MFO)	Does not discuss effective applications	No	2
R. Chaganti	IDS for SDN-enabled IoT	Limited to four types of intrusions; no focus on detection rate optimization	No	4
A. Henry	Combining Different Models	Lack of false alarm rate and detection rate optimization	No	2
A. Fatani	Not Specified	Similar weaknesses in false alarm rate and detection rate optimization	No	2
M. Bacevicius	Machine Learning	Similar weaknesses in false alarm rate and detection rate optimization	No	2
H. Alshahrani	Not Specified	Similar weaknesses in false alarm rate and detection rate optimization	No	2
S. Alosaimi	Deep Learning	Limited to DDoS attacks; complicated system	No	1
Y. Alotaibi	Ensemble Method	Does not extend to developing a practical defense mechanism	No	2
X. Larriva-Novo	Explainable AI	Lack of mathematical justification	No	2
S. V. N. Santhosh Kumar	Survey Using ML	Focus on performance improvement while ignoring practical applications	No	2

3. Methodology

3.1. Deep Learning Model Analysis & Selection

Selecting an appropriate network is one of the challenges in DL technology [23]. This statement applies to intrusion detection using DL, as it inherits the natural characteristics of the technology [24]. H. Liu et al. surveyed multiple DL models for intrusion detection [25]. According to this study, Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Gated Recurrent Units (GRUs) are more effective than other DL models.

3.1.1. Recurrent Neural Networks (RNNs)

The IoT devices receive short sequential data as network packets. RNNs perform excellently in classifying short sequential data. The working principle of RNNs is defined by Equation (1), where i_t is the input at time step t; s_t is the hidden state; o_t is the output;

V_{ss}, V_{is}, and V_{so} are the weight matrices; and c_s and c_o are the biases. The variable $\sigma(\cdot)$ is the activation function [26].

$$s_t = \sigma(V_{ss}s_{t-1} + V_{is}i_t + c_s) \tag{1}$$

$$o_t = V_{so}s_t + c_o \tag{2}$$

According to the mathematical principle of RNN defined in Equations (1) and (2), it is effective at classifying network packets that maintain timestamps. That is why it is a potential candidate for intrusion detection at IoT nodes. However, RNN has limitations, including vanishing gradient problems and performance issues for long sequences.

3.1.2. Long Short-Term Memory (LSTM) Networks

The LSTM network is a variation of RNNs that overcomes the limitations of RNNs [27]. It combines a memory cell to prevent vanishing gradient problems with a tri-gating mechanism to handle long sequences. These gates are defined by Equations (3)–(5), which are input, forget, and output gates, respectively [28].

$$p_t = \sigma(Q_p[s_{t-1}, i_t] + d_p) \tag{3}$$

$$q_t = \sigma(Q_q[s_{t-1}, i_t] + d_q) \tag{4}$$

$$r_t = \sigma(Q_r[s_{t-1}, i_t] + d_r) \tag{5}$$

The input signals to the network update the cell state and specify a candidate cell for the subsequent sequence. As a result, LSTM networks are better at handling network packets. The update and candidate cell states are defined by Equations (6) and (7), respectively.

$$\tilde{D}_t = \tanh(Q_D[s_{t-1}, i_t] + d_D) \tag{6}$$

$$D_t = p_t \odot D_{t-1} + q_t \odot \tilde{D}_t \tag{7}$$

The network's final output is modeled as Equation (8).

$$s_t = r_t \odot \tanh(D_t) \tag{8}$$

In Equations (3)–(5), p_t, q_t, and r_t represent the forget, input, and output gates, respectively. D_t and $\tilde{D}_t$ denote the updated and candidate cell states, respectively. The weight matrices Q_p, Q_q, Q_r, and Q_D and the bias vectors d_p, d_q, d_r, and d_D are the model parameters. The symbol $\odot$ denotes element-wise multiplication.

3.1.3. Gated Recurrent Units (GRUs) Network

This is a simplified variant of LSTM networks. Unlike LSTM networks, it uses two gates, updated and reset, expressed as Equations (9) and (10). In Equation (9), z_t is the update gate, W_z is the weight matrix for the update gate, h_{t-1} is the previous hidden state, x_t is the input at the current time step, and b_z is the bias for the update gate. And in Equation (10)

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{9}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{10}$$

The GRU network uses a candidate state, which is a filtered version of the input combined with the previous hidden state. The linear interpolation between the previous and candidate hidden states produces the current hidden state. In Equation (11), $\tilde{h}_t$ is the candidate hidden state, W_h is the weight matrix for the candidate hidden state, and b_h is the bias for the candidate hidden state. In Equation (12), h_t is the current hidden state.

$$\tilde{h}_t = \tanh(W_h \cdot [r_t \odot h_{t-1}, x_t] + b_h) \tag{11}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{12}$$

3.1.4. Selection of an Appropriate Deep Learning Model

All three networks mathematically analyzed in Section 3.1 show their ability to handle network sequential data. A sample of the data is presented in Figure 1. To properly detect the intrusion, it is essential to temporarily store the correct state and explore the anomalous nature of the subsequent states. Moreover, the network sequences are random, and the network processing them must be capable of handling both long and short sequences.

```
2023-04-21    09:15:00,    192.168.1.5,
203.0.113.75,  55555,  80,  TCP,  1600,
10   2023-04-21   09:15:01,   192.168.1.6,
198.51.100.22,  54321,  443,  TCP,  1300,
7   2023-04-21   09:15:02,    192.168.1.5,
203.0.113.75, 55555, 80, TCP, 1600, 3 2023-
04-21  09:15:03,  192.168.1.7,  192.0.2.66,
49270, 123, UDP, 50, 1
```

Figure 1. A sample of sequential data in the experimental network.

Regarding random sequences with varying lengths, the LSTM networks perform better than RNN and GRU networks [29]. Moreover, it is not affected by vanishing gradient problems like RNN. Furthermore, it has three gates capable of adjusting the data flow according to the volume of traffic on the network. Considering the advantages, the LSTM network has been selected as the DL model to develop the IDS.

3.2. Dataset & Preprocessing

The CIC-IDS2017 dataset used in this paper is chosen for its comprehensive features, thorough documentation, and widespread recognition. Developed by the Canadian Institute for Cybersecurity, this dataset is consistently updated by the same institution [30]. Comprising over 2.8 million instances, it presents a diverse range of intrusion types in IoT nodes. The dataset encompasses a total of 78 feature variables. Table 2 details a subset representing each class label.

Table 2. A subset of the CIC-IDS2017 dataset with all class labels.

Duration	Src IP	Src Port	Dst IP	Dst Port	Protocol	Label
0	10.0.2.15	57158	216.58.208.46	80	6	Benign
5	192.168.10.5	80	192.168.10.50	57164	6	FTP-Patator (FTPP)
4	192.168.10.5	22	192.168.10.50	57165	6	SSH-Patator (SSHP)
3	192.168.10.5	80	192.168.10.50	57166	6	DoS slowloris (DoSS)
2	192.168.10.5	80	192.168.10.50	57167	6	DoS Slowhttptest (DoSL)
1	192.168.10.5	80	192.168.10.50	57168	6	DoS Hulk (DoSH)
0	192.168.10.5	80	192.168.10.50	57169	6	DoS GoldenEye (DoSG)
0	192.168.10.5	443	192.168.10.50	57170	6	Heartbleed (HRB)
3	192.168.10.5	80	192.168.10.50	57171	6	Web Attack—Brute Force (WAB)
2	192.168.10.5	80	192.168.10.50	57172	6	Web Attack—XSS (WAX)
1	192.168.10.5	80	192.168.10.50	57173	6	Web Attack—SQL Injection (WAQ)
0	192.168.10.5	80	192.168.10.50	57174	6	Infiltration (INF)
0	192.168.10.5	80	192.168.10.50	57175	6	Bot
1	192.168.10.5	80	192.168.10.50	57176	6	PortScan (PRS)
0	192.168.10.5	80	192.168.10.50	57177	6	DDoS

3.2.1. Dataset Cleaning & Splitting

There are multiple missing and duplicate values in the original dataset. They have been manually cleaned. After cleaning, the cleaned dataset has 55,910 instances. This has been split into training, testing, and validation sets with a ratio of 70:15:15. The data have been randomly distributed among these subsets [31]. There are 39,137 instances of training the network at this ratio. Both the testing and validation set have 8386 instances. In every phase of data splitting, the instances have been randomly shuffled. There are 78 exclusive features in the CIC-IDS2017 dataset. It is a labeled dataset with 15 classes. Of these classes, 14 are malicious, and one is benign.

3.2.2. Dataset Normalization

There are multiple numerical features in the dataset whose scale significantly varies. As a result, it is essential to normalize these features to prevent the LSTM network from incorrect prediction. This paper uses the Min–Max algorithm, which performs feature normalization. It maps the target features on a scale of 0 to 1 [32]. It is performed by Equation (13), where the feature value is x, and the highest and lowest values of the features are X_{min} and X_{max}, respectively.

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{13}$$

After feature normalization, each feature lies within the 0 to 1 range. As a result, the learning algorithm smoothly converges to the lower error region faster.

3.2.3. Categorical Feature Processing

There are non-numeric features in the CIC-IDS2017 dataset. At the same time, there are some numeric features, for example, IP addresses and port addresses, whose numerical values do not represent themselves. These features have been converted into categorical features using the one-hot encoding scheme. In this process, the categorical values are converted into a vector of zeros, except the index representing the class, which is set to 1. The categorical variable classes $C = \{c_1, c_2, ..., c_n\}$ are represented as c_i as defined by Equation (14), where the ith index is 1 and the rest are zero [33].

$$\text{one\_hot}(c_i) = \begin{bmatrix} 0 & 0 & \dots & 1 & \dots & 0 \end{bmatrix} \tag{14}$$

3.2.4. Feature Selection

There are both categorical and numerical features in the dataset. The chi-squared test has been used in this paper to determine the significant association between categorical variables and the target class. The chi-squared statistic is calculated using Equation (15), where χ^2 is the chi-squared statistic [34].

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{15}$$

The chi-squared statistic measures the difference between observed and expected frequencies. If the value is larger, it is evident that the two variables are more dependent. A smaller value of χ^2 represents the weak association between two variables [35]. Features that exhibit strong associations with the target variables have been used in this paper.

The numerical features are selected using the variance threshold (VT) method, defined by Equation (16), where x is the feature, n is the number of instances, and $\bar{x}$ is the mean value of the features. Based on the variance measured using Equation (16), the features are selected, which is expressed by Equation (17).

$$\text{Variance}(X) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2 \tag{16}$$

$$\text{Selected Features} = \{X \mid \text{Variance}(X) > T\} \tag{17}$$

The significant feature correlation coefficients determined through the chi-squared test conducted in this study are detailed in Table 3. A coefficient threshold of 0.41 was established for this experiment; features with coefficients below this threshold were excluded from consideration in this paper. Applying the 0.41 threshold, the top 20 most impactful features were retained to train the network.

Table 3. The correlation coefficient according to the chi-squared test.

Feature	Correlation	Feature	Correlation	Feature	Correlation
Flow duration	0.54	Device type	0.57	Payload size	0.27
Number of packets	0.8	Operating system	0.62	HTTP method	0.26
Payload entropy	0.63	Application software	0.52	HTTP request path	0.24
Protocol	0.8	Network topology	0.71	HTTP request headers	0.29
Packet size	0.67	User behavior	0.73	HTTP response code	0.24
TCP flags	0.76	Subnet ID	0.47	HTTP response headers	0.23
IP header flags	0.73	VLAN ID	0.5	IP addresses	0.13
HTTP headers	0.61	Login time	0.41	Device type	0.27
Payload	0.63	Logout time	0.47	Operating system	0.26
Entropy	0.79	Network access patterns	0.46	Application software	0.23

3.2.5. Sequence Generation

The network packets received by the IoT nodes form sequential data with timestamps. However, these sequences vary depending on multiple factors, including the volume of data, encryption algorithm, and inserted malicious code. The proposed LSTM network requires sequential data as the input, which is not readily available in the CIC-IDS2017 dataset. This paper uses the sliding window approach to produce sequential data, which resembles the network packet stream the IoT nodes handle [36]. It has been done using the mathematical principle defined in Equation (18).

$$(X_{t-w+1}, X_{t-w+2}, \ldots, X_t) \to Y_{t+1} \tag{18}$$

In Equation (18), the sliding window is represented as w, the input feature at timestamp t is X_t, and Y_{t+1} is the output feature at the same timestamp. The window slides over the table and generates input and equivalent output sequences according to the defined timestamp. Because of the data variations in the table, the sequences randomly vary. At the same time, the timestamps are random as well. As a result, it resembles the actual network traffic stream sequence.

3.3. IoT Node Architecture and Vulnerabilities

The proposed Intrusion Detector System (IDS) works at the IoT node. Understanding the IoT node architecture and communication protocol is essential for seamlessly integrating the proposed IDS. The IoT node architecture, associated communication protocol, and possible security vulnerabilities have been discussed in this section.

3.3.1. IoT Node Design Pattern

The IoT nodes connected to the IDS, an edge server, are illustrated in Figure 2. Each node consists of four units. These units are Sensor, Processor, Communication System, and Power Source. The sensors sense the intended input to the IoT device. The processor processes it. The communication system employs the communication protocols and governs the nodes' communication over the internet. The proposed IDS operates from the edge server at the internet access point (IAP). The edge server communicates with the internet gateway through the IAP [37].

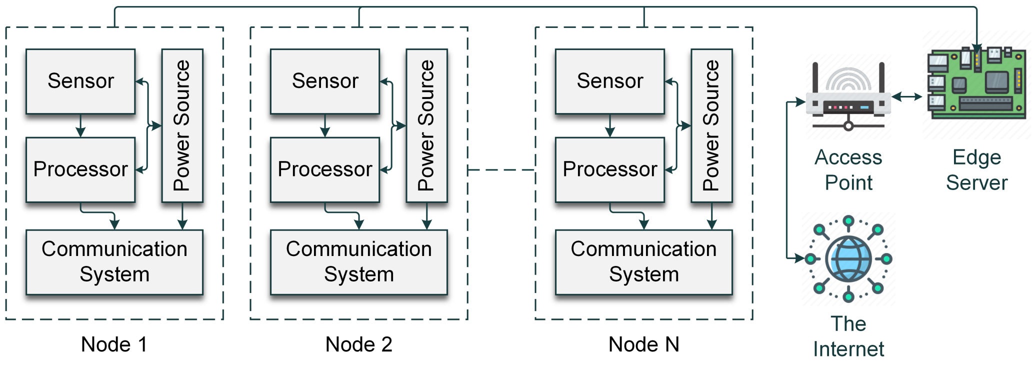

Figure 2. The design pattern of the IoT node.

3.3.2. Anomaly Indication Analysis

Malicious signals are injected into the IoT networks, causing IoT devices to deviate from their natural characteristics [38]. It also hampers the regular performance of the devices. The deviation from the natural characteristics and performance issues are prominent indicators of IoT intrusion. Usually, the energy consumption rate changes when an IoT node is compromised [39]. Anomalies in latency and throughput are also indicators of possible intrusion. The energy consumption and throughput are defined by Equations (19) and (20).

$$E = P_{active} \cdot T_{active} + P_{idle} \cdot T_{idle} \tag{19}$$

$$Th = \frac{D}{T} \tag{20}$$

3.3.3. Vulnerabilities & Probabilistic Model

The proposed IDS has been trained on the CIC-IDS2017 dataset. This dataset contains 15 target classes listed in Table 2. There are 14 intrusions and 1 benign class in the target variable. The proposed IDS is capable of detecting these 14 vulnerabilities. A probabilistic model has been developed in this paper, expressed as Equation (21), to define a particular detected intrusion based on its probability [40]. This probabilistic model measures the probability of successful intrusions provided in the dataset. It has been observed that each intrusion on the dataset has the potential to affect the IDS. That is why all 14 intrusions have been considered in this paper.

$$P(A|C,S,D) = \frac{P(A,C,S,D)}{P(C,S,D)} \tag{21}$$

In Equation (21), $P(A)$ is the probability of a successful attack, and $P(C)$ represents the capabilities of the attacker. The probability of breaching the existing protocol is $P(S)$, and $P(D)$ expresses the probability of the network's defense mechanisms.

3.4. Proposed IDS

There is an LSTM network at the heart of the proposed IDS. It runs from an edge server. The network packets from the internet access the server through the IAP. The bit stream is converted into sequences using the method explained in Section 3.2.5. These sequences are analyzed through the LSTM network. It classifies the sequences and allows them to pass to the IoT node if they are benign. Suppose the sequences carry a malicious signal intended to intrude on the IoT nodes. In that case, they are diverted from them, and necessary actions are taken using the Dynamic Access Control (DAC) algorithm.

3.4.1. LSTM Network Architecture

The LSTM network designed for the IDS proposed in this paper is illustrated in Figure 3. The architecture of this network has been finalized through multiple iterations by experimenting with different numbers of layers, which have been described in Section 3.4.2. It has been observed that the network performs optimally with 128 LSTM nodes. There is a dense layer after the LSTM layer with 15 nodes. After that, a softmax layer maps the signals from the nodes of the dense layer within the probability scale. Based on this probability, the output layer classifies the input into one of the 15 outputs.

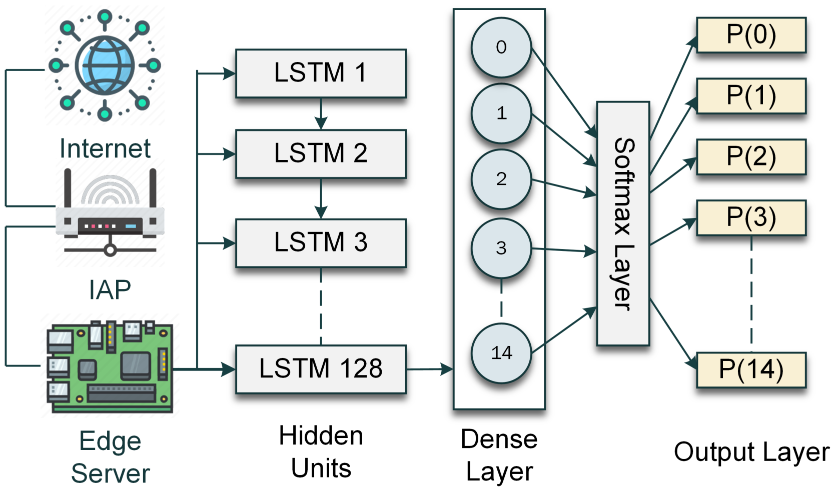

Figure 3. The LSTM network architecture.

3.4.2. Network Architecture Optimization

The network architecture has been optimized through 16 instances of the testing versus validation curve experiment. LSTM nodes increased from 32 to 152 in this experiment, with 8 new nodes simultaneously. The optimized network architecture has been decided from the characteristics and comparison of the testing and validation accuracy curve illustrated in Figure 4.

The difference between training and validation accuracy is marginal. This statement is valid for up to 128 hidden nodes. The training accuracy keeps increasing at a near-constant slope after 128 LSTM nodes. However, the validation accuracy reduces rapidly. This indicates that the network overfits more than 128 LSTM nodes. It also signifies that the network is optimized with 128 LSTM nodes.

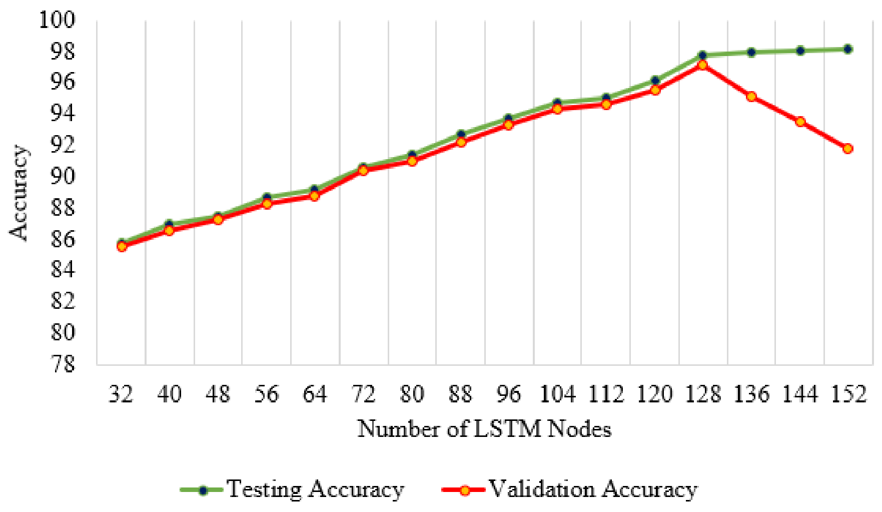

Figure 4. Network architecture optimization through testing and validation accuracy difference.

3.4.3. Training the Network

The optimized LSTM network has been trained with 39,137 instances. The overall training time is 18 min and 35 s. The learning curves are illustrated in Figure 5. It is evident from the figure that, after the first iteration, the training and validation accuracy maintains consistency. The same statement applies to training and validation loss. The training is completed after 10 epochs with 415 iterations every epoch. The proposed LSTM network learns to classify the intrusions with 97.16% validation and 97.71% training accuracy.

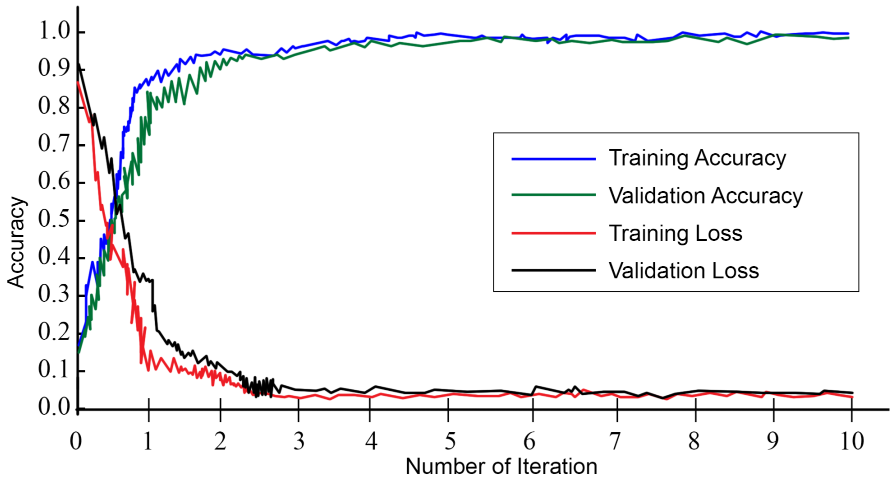

Figure 5. Learning progress with respect to iterations.

Weight Initialization

Random weight initialization is common in deep learning [41]. However, proper weight initialization is key to faster convergence and optimized results [9]. That is why the He weight initialization method has been used in this paper [42]. It is defined by Equation (22), where $\mathcal{N}(0, \sqrt{\frac{2}{n_{\text{input}}}})$ is the Gaussian distribution in which the mean is zero, and $\sqrt{\frac{2}{n_{\text{input}}}}$ is the standard deviation.

$$W \sim \mathcal{N}\left(0, \sqrt{\frac{2}{n_{\text{input}}}}\right) \tag{22}$$

The He weight initialization method generates the primary weights of the nodes from the 0 mean Gaussian distribution and variance. This variance is preserved throughout both back-propagation and forward-propagation. As a result, the probability of the vanishing gradient is lowered. At the same time, it prevents an exploding gradient as the weights maintain a steady variance.

Learning Algorithm

The Adaptive Moment Estimation (ADAM) optimization algorithm has been used to train the proposed LSTM network [43]. It combines the Adaptive Gradient (AdaGrad) Algorithm and the Root Mean Square Propagation (RMSProp) Algorithm, so it has both advantages. In the beginning, the first and the second moments are calculated using Equations (23) and (24), respectively. In these equations, the α_1 and α_2 represent the exponential decay rates for the moments.

$$p_t = \alpha_1 p_{t-1} + (1 - \alpha_1) a_t \tag{23}$$

$$q_t = \alpha_2 q_{t-1} + (1 - \alpha_2) a_t^2 \tag{24}$$

Once the moments are calculated, it is essential to correct the biases, which has been done using Equations (25) and (26), respectively.

$$\hat{p}_t = \frac{p_t}{1 - \alpha_1^t} \tag{25}$$

$$\hat{q}_t = \frac{q_t}{1 - \alpha_2^t} \tag{26}$$

Finally, the weights are updated using Equation (27), where λ is the learning rate, and δ is a small constant. The ω_t in the equation is the updated weight, and ω_{t-1} is the previous weight [44].

$$\omega_t = \omega_{t-1} - \lambda \frac{\hat{p}_t}{\sqrt{\hat{q}_t} + \delta} \tag{27}$$

3.5. Dynamic Access Control (DAC) Algorithm

One of the novel contributions of this paper is the development of the Dynamic Access Control (DAC) algorithm, which is presented as Algorithm 1. This algorithm has been developed to effectively apply the LSTM network developed in this paper to intrusion detection and defense. Furthermore, the algorithm includes additional functionalities to detect intrusion and defend against further intrusion from the same source.

The Dynamic Access Control (DAC) algorithm is presented as Algorithm 1. It senses the presence and absence of network packets. When the bit stream is available, the algorithm enters a while loop and iterates as long as the IDS receives network packets. Initially, the DAC algorithm converts the packet stream into sequences segmented at timestamp t. The sequences are then passed to the LSTM network, and the network prediction is stored in variable d. If the sequence is not malicious, the algorithm passes the sequence to the IoT node. Otherwise, it dynamically generates a random delay r measured in seconds. After that, the IP address of the source is blocked for r seconds. At the same time, the packets are discarded, protecting the IoT node. This is how the proposed IDS detects the intrusion and defends against further intrusion from the same source.

Algorithm 1 Dynamic Access Control Algorithm

```
Require: $p$ : Network Packet
  $r \leftarrow$ RandBetween(1, 2)
  while $p \neq 0$ do
      $s_t \leftarrow$ Sequence(p, t)
      $d \leftarrow LSTM(s_t)$
      if $d$ is benign then
          IoTNode(d)
      else if $d$ is malicious then
          $r \leftarrow$ RandBetween(r, r+r)
          $ip \leftarrow$ IPScan(Source(p))
          RouterAPI.AccessControl(ip, false, r)
          Discard(d)
      end if
  end while
```

4. Implementation & Detection Rate Optimization

The proposed methodology has been implemented and experimented with in a test environment. However, the testbed replicates the real-world intrusion pattern, ensuring the experiment's integrity.

4.1. Experimental Setup

The proposed IDS and the DAC algorithm have been hosted in a Raspberry Pi 4 Model B. It has 4 GB primary memory. Odroid N2, NVIDIA Jetson Nano, and Orange Pi 5 are alternate options for the Raspberry Pi 4. The Raspberry Pi 4, with its balance of affordability and performance, is an excellent choice among single-board computers. A vast and active community supports it. While alternatives like the Odroid N2 and NVIDIA Jetson Nano provide higher performance in specific areas, they come at a higher cost and with less community support. The Orange Pi 5, though a strong competitor, still lags in terms of the software ecosystem and user community. The Raspberry Pi 4's strong support network, software compatibility, and array of compatible accessories make it a highly adaptable and cost-effective option, solidifying its position as the choice to implement the proposed IDS. The Python programming language with the TensorFlow library has been used to implement and train the LSTM network. The DAC algorithm has been coded in Python as well. The Cooja Network Simulator (CNS) [45] has been used to generate IoT network traffic and inject malicious packets [46]. No specific pattern has been maintained while creating the malicious traffic.

4.2. Detection Rate Optimization

During the experimental observation after implementation, it was observed that the IDS suffers from a high false positive rate (FPR). As a result, the false alarm rate (FAR) is high. It has been further observed that if the detection rate (DR) increases, the FPR increases. Conversely, the FPR declines when the DR declines. The relationship among the DR, FPR, and the threshold is illustrated in Figure 6.

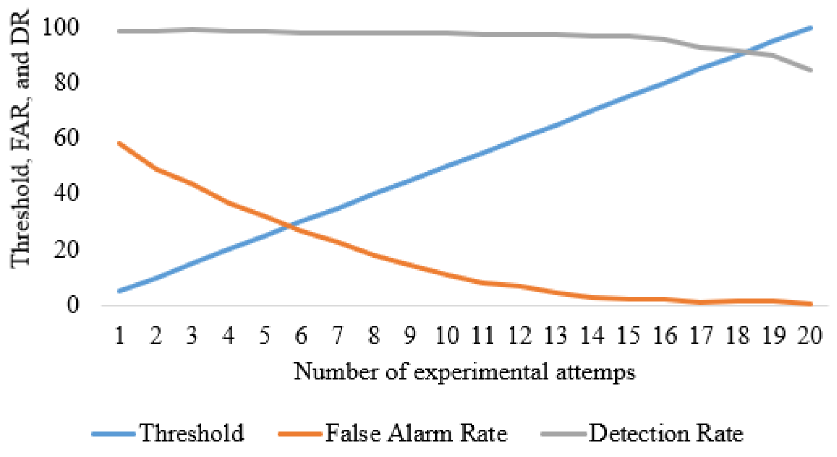

Figure 6. Threshold selection by FAR and DR analysis.

It is observable in Figure 6 that, as the threshold increases, the FPR declines. However, the DR declines as well. Reducing the DR raises a concern about the overall performance of the IDS. On the other hand, high FPR downgrades the effectiveness and usability of the proposed IDS. After in-depth analysis and the trade-off between FPR and DR, it has been observed that the IDS demonstrates optimal performance at 86.88. At this threshold, the FPR is only 1.14%, and the DR is 98.17%.

5. Performance Evaluation

The proposed IDS's performance is analyzed and presented in this section. First, the evaluation metrics are specified in this section, along with their underlying mathematical definitions. After that, a confusion matrix analysis is performed. Finally, the detection rate and response time are analyzed.

5.1. Evaluation Metrics

The evaluation metrics observed in the state-of-the-art literature have been used in this paper [47]. These evaluation metrics are derived from the number of instances of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) [48]. These values are obtained from the confusion matrix illustrated in Figure 7. The accuracy, precision, recall (sensitivity), and F1 score have been calculated using TP, TN, FP, and FN. These are defined by Equations (28), (29), (30), and (31), respectively [49].

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{28}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{29}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{30}$$

$$\text{F1 Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \tag{31}$$

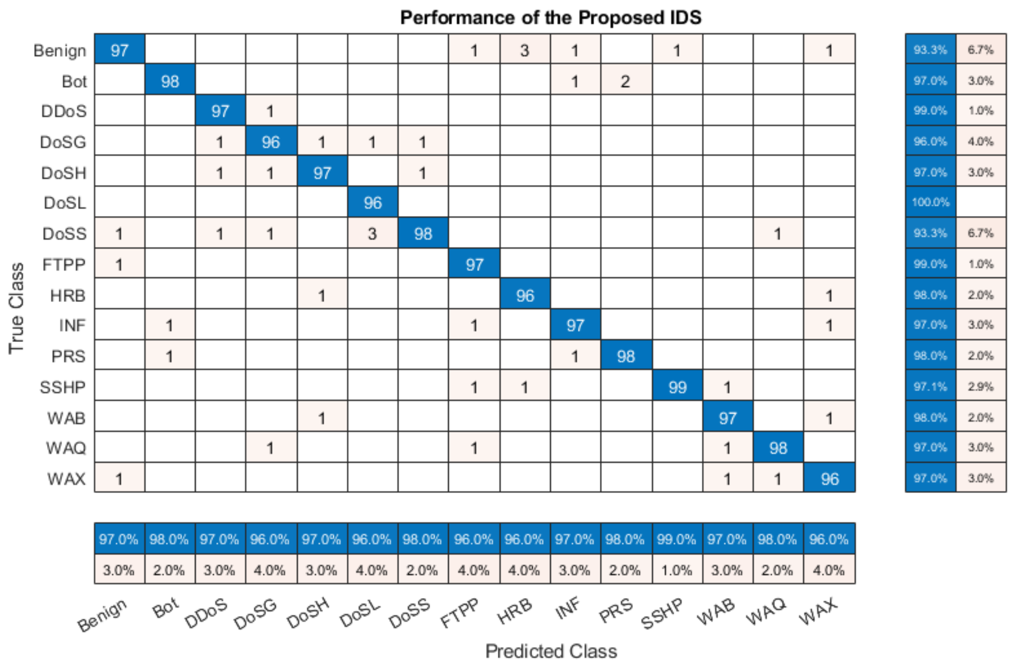

Figure 7. Confusion matrix analysis.

5.2. Confusion Matrix Analysis

The confusion matrix presented in Figure 7 shows the proposed IDS's performance. Observing the true positive values, the SSHP achieves the highest score of 99, whereas intrusions DoSG, DoSL, HRB, and WAX share the lowest score of 96. Regarding false positives, PRS has the lowest occurrence, with only 1, and DoSH, DoSS, and WAQ have the most with 5. DoSH has the lowest count of 0 for false negatives, while Benign records the highest with 8. The recall percentage is consistently high across all intrusions, ranging from 96% to 99%. The precision metric shows the greatest variation, with DoSL achieving the highest score at 100%, and with Benign and DoSS registering the lowest at 93.3%. The F1 score, which balances precision and recall, falls between 95.1% for Benign and 98% for DDoS, PRS, and SSHP, revealing the overall performance and effectiveness of the model across different intrusions.

Table 4 summarizes the performance of the proposed IDS illustrated in the confusion matrix of Figure 7.

Table 4. The performance of the proposed IDS according to the confusion matrix.

Intrusion	TP	FP	FN	Recall	Precision	F1 Score
Benign	97	3	8	97	93.3	95.1
Bot	98	2	3	98	97	97.5
DDoS	97	3	1	97	99	98
DoSG	96	3	4	96	96	96
DoSH	97	5	0	97	97	97
DoSL	96	2	6	96	100	97.9
DoSS	98	4	1	98	93.3	95.6
FTPP	97	4	1	96	99	97.5
HRB	96	3	2	96	98	97
INF	97	2	3	97	97	97
PRS	98	1	3	98	98	98
SSHP	99	3	6	99	97.1	98
WAB	97	2	5	97	98	97.5
WAQ	98	4	3	98	97	97.5
WAX	96	3	3	96	97	96.5

5.3. Detection Rate and Response Time Analysis

The proposed IDS's intrusion detection rate and response time performance are listed in Table 5. The detection rate is measured using Equation (32). The average intrusion attempted is 110, and the average detected intrusion is 108.6. This leads to an average response rate of 1.14 s and an average detection rate of 98.73%.

$$\text{Detection Rate, R} = \frac{\text{Detected Intrusion}}{\text{Total Intrusion}} \times 100 \tag{32}$$

Table 5. The detection and the response time of the proposed IDS.

Intrusion	Intrusion	Detected	Response Time (s)	Detection Rate (%)
Benign	110	108	0.23	98.18
Bot	110	107	1.8	97.27
DDoS	110	107	1.98	97.27
DoSG	110	108	1.05	98.18
DoSH	110	108	0.92	98.18
DoSL	110	109	1.1	99.09
DoSS	110	109	0.96	99.09
FTPP	110	110	1.25	100.00
HRB	110	109	1.46	99.09
INF	110	108	0.97	98.18
PRS	110	108	1.84	98.18
SSHP	110	110	0.36	100.00
WAB	110	109	1.85	99.09
WAQ	110	109	0.65	99.09
WAX	110	110	0.68	100.00

Noticeably, FTPP, SSHP, and WAX show a perfect detection rate of 100%, with the actual and detected intrusions both at 110. Conversely, Benign, Bot, DDoS, DoSG, DoSH, INF, and PRS recorded slightly lower detection rates at 98.18%, while the remaining intrusions reached 99.09%. Response times exhibit a wider range of values, with SSHP having the fastest response time of 0.36 s and DDoS the longest at 1.98 s. The categories of DoSL and DoSS boast nearly perfect detection, with a moderate response time of 1.1 s and 0.96 s, respectively, balancing efficiency and effectiveness. On the other hand, FTPP, though having a perfect detection rate, takes 1.25 s to respond. The performance of the proposed IDS is illustrated in Figure 8 in terms of detected intrusion, detection rate, and response time.

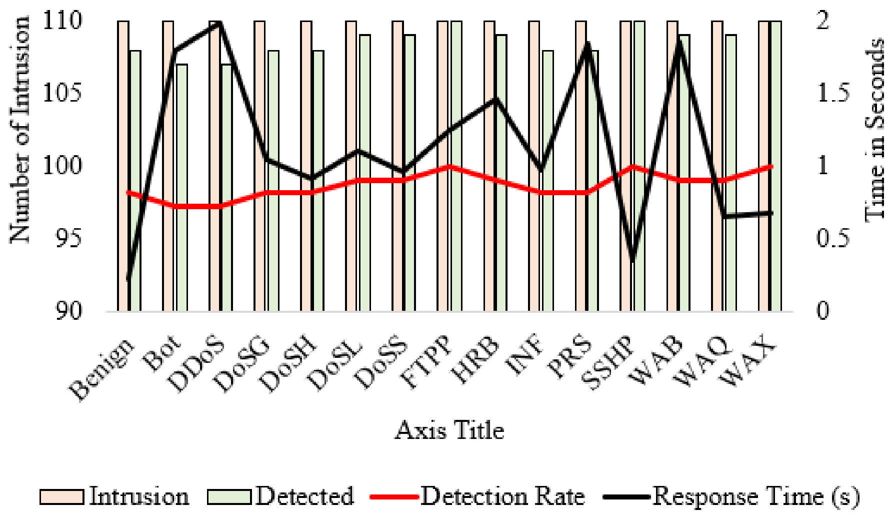

Figure 8. The detection rate and response time.

5.4. Performance Comparison

Table 6 showcases a comparative analysis of the proposed methodology's performance metrics against various other methods from different papers. Each paper here uses a similar data processing method as used in this paper. As a result, this evaluation is not biased towards the processed data of this paper. The metrics assessed include the number of intrusions detected, precision, recall, F1 score, detection rate, and response time. M. A. Almaiah et al. [50] and J. Cui et al. [51] both detect two intrusions, with Almaiah et al. showing a higher precision of 93.94%, as opposed to Cui et al.'s 88.55%. M. Barhoush et al. [52] distinguish themselves by detecting a significantly higher number of intrusions, with a commendable F1 score of 93.9%. However, the proposed methodology outperforms all the listed papers with the same capability of detecting 14 intrusions, boasting an F1 score of 97.05%, and an impressive detection rate of 98.73%. Moreover, it is the only method in the table to report a response time, which is a swift 1.14 s.

Table 6. Performance comparison of the proposed methodology with other papers.

Paper	Number of Intrusions	Precision	Recall	F1 Score	Detection Rate
M. A. Almaiah et al.	2	93.94	93.23	94.44	NA
J. Cui et al.	2	88.55	86.59	86.88	NA
M. Barhoush et al.	14	91.77	96.85	93.9	NA
M. Jeyaselvi et al.	2	95.65	92.74	91.48	98.16
N Sarkar et al.	2	88.3	87.78	87.68	NA
Proposed	14	97.07	97.11	97.05	98.73

5.5. Performance on BoT-IoT Dataset

The CIC-IDS2017 dataset is robust and contains a wide range of intrusions. The proposed IDS has been trained with this dataset to extend its detection range. However, another dataset, the BoT-IoT dataset, focuses more on IoT intrusions. An experiment has been conducted to evaluate the performance of the proposed IDS on the BoT-IoT dataset. Even though both CIC-IDS2017 and BoT-IoT datasets contain intrusion-related data, there are some mutually exclusive classes in these two datasets [53]. Similar classes in these two datasets have been identified using Equation (33), where D_{cic} and D_{bot} represent the CIC-IDS2017 and BoT-IoT datasets, respectively.

$$D_{cic} \overline{\bigoplus} D_{bot} = \{DoS, DDoS\} \tag{33}$$

A comparative analysis of proposed IDS performance using the CIC-IDS2017 and BoT-IoT datasets is presented in Table 7. The data reveal a strikingly similar level of effectiveness in detecting both DoS and DDoS attacks across these datasets. For DoS attack detection, the IDS shows an accuracy of 94.87% on the CIC-IDS2017 dataset and a closely comparable 94.11% on the BoT-IoT dataset. This similarity is further mirrored in the precision, recall, and F1 scores, with only minimal differences (less than 1% in most cases). Similarly, for DDoS attacks, the accuracy on CIC-IDS2017 is 93.26%, versus 91.64% on BoT-IoT, again showing a consistent performance level. The precision and recall numbers for DDoS attacks are also quite similar yielding F1 scores close to one another. Such simultaneous performance on two unrelated datasets highlights the effectiveness of IDS, showing its stable efficiency in detection of DoS and DDoS attacks in different network environments. The performance comparison is shown in Figure 9.

Table 7. Performance evaluation comparison against DoS and DDoS attacks using different datasets.

Intrusion Type	**CIC-IDS2017 Dataset**				**BoT-IoT Dataset**			
	Accuracy	**Precision**	**Recall**	**F1 Score**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
DoS	94.87	94.01	93.22	92.18	94.11	93.45	92.60	91.77
DDoS	93.26	93.14	93.45	92.74	91.64	92.21	92.44	90.15

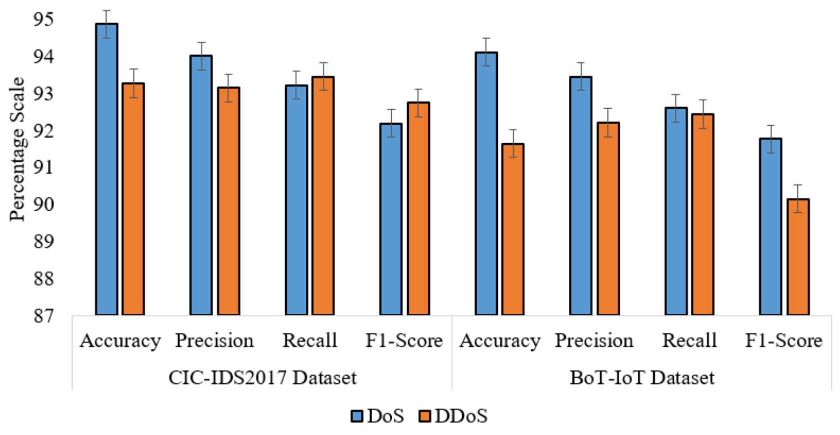

Figure 9. Performance comparison between CIC-IDS2017 and BoT-IoT datasets.

5.6. Feature Selection Effectiveness

The proposed IDS has been trained with selected features identified through the chi-squared test discussed in Section 3.2.4. As a result, features with a weak correlation with the target variable were excluded from the training process. An experiment has been conducted to analyze the effectiveness of the feature selection method applied in this paper by comparing the training performance of the proposed IDS with and without the feature selection method. Table 8 demonstrates the effectiveness of the feature selection method.

According to the data in Table 8, the convergence time is significantly reduced by 24.69%, from 24.67 min with all features to 18.58 min with selected features, indicating enhanced efficiency. This improvement is paralleled in accuracy metrics; training accuracy sees an increase of 7.29% from 90.42% to 97.71%, and validation accuracy improves by a total of 7.41%, from 89.75% to 97.16%. These figures underscore the substantial benefits of feature selection, not only in reducing computational time but also in significantly enhancing the model's performance during both training and validation, highlighting its critical role in developing efficient and precise IDS.

Table 8. Feature selection method's effectiveness analysis.

Evaluation Criteria	Selected Features	All Features	Effectiveness
Convergence Time	18.58 min	24.67 min	24.69% less time
Training Accuracy	97.71%	90.42%	7.29% improvement
Validation Accuracy	97.16%	89.75%	7.41% improvement

5.7. Resource Consumption

The hourly resource consumption data of the proposed IDS are listed in Table 9. During the early hours (00:00–03:00), there is a noticeable spike in CPU utilization, particularly at 02:00–03:00, reaching 76%, indicating heightened processing activity. Interestingly, memory usage does not follow a similar trend, remaining relatively stable around 2.5 GB initially and gradually increasing to 3.2 GB by 06:00. Disk I/O operations fluctuate moderately, peaking at 47 IOPS during 05:00–06:00. Network throughput remains consistently below 1.2 Mbps, implying a steady but not overwhelming network load. Power consumption, correlating with CPU usage, escalates from 3 Watts to 4 Watts, suggesting a direct relationship between processing intensity and energy usage.

Table 9. Hourly resource consumption of LSTM-based IoT IDS on Raspberry Pi.

Time Interval (Hour)	CPU Utilization (%)	Memory Usage (GB)	Disk I/O (IOPS)	Network Throughput (Mbps)	Power Consumption (Watts)
00:00–01:00	38	2.5	43	0.8	3
01:00–02:00	47	2.3	37	0.7	2.8
02:00–03:00	76	2.1	37	0.9	2.6
03:00–04:00	38	2.8	41	0.9	3.2
04:00–05:00	40	3.0	40	1.0	3.5
05:00–06:00	38	3.2	47	1.1	3.7
06:00–07:00	39	3.5	38	1.2	4.0
07:00–08:00	42	3.8	39	0.9	4.3
08:00–09:00	44	4.0	42	1.2	3.8
09:00–10:00	42	4.2	40	0.9	3.9

6. Limitations & Future Scope

The performance evaluation of the proposed IDS demonstrates remarkable results. Notwithstanding the outstanding performance, the proposed IDS has multiple limitations, which are discussed in this section.

6.1. Implementation Cost

The proposed IDS has been implemented using a Raspberry Pi B headless computer. It is a multipurpose computer and is expensive. As a result, the overall implementation cost of the proposed IDS is high. However, designing an embedded system only for the proposed IDS effectively reduces the cost. Moreover, a faster market adoption would reduce the cost even further. It opens the door to further research on implementing it as an embedded system. It also paves the way for conducting research from a business perspective.

6.2. Risk of Adversarial Machine Learning Attack (AML)

The proposed IDS has not been tested with an AML attack. The attempt to analyze the impact of an AML attack failed because of the lack of an appropriate dataset. However, the scope of conducting more experiments remains open. Subsequent research will explore the response to AML attacks and prevention mechanisms for the proposed IDS [33].

6.3. Experimental Testbed

The proposed IDS has been studied in an experimental testbed. Every possible measurement has been taken to resemble a real-world intrusion-like behavior of the testbed. However, exploring its performance in a real-world IoT environment would further support the system's overall integrity. An effort to do so is ongoing. The IDS's performance in a real-world environment will be presented in future work upon getting the opportunity.

The limitation of the LSTM-based IDS with the DAC algorithm is an opportunity to conduct further research to improve its quality. From this context, these limitations create the opportunity for the future scope of this research.

7. Conclusions

The LSTM-based intrusion detection system presented in this paper has an innovative Dynamic Access Control algorithm that detects the intrusion and prevents further intrusions. It blocks the source of intrusion for a dynamic period, which increases according to the number of intrusions. This unique approach makes this paper different than DL-based intrusion detectors. The network architecture has been designed through performance analysis and learning curve deviation. As a result, the LSTM network used in this paper performs optimally. Moreover, the training process used in this paper utilizes the He weight initialization method accompanied by an ADAM optimizer. It contributed to achieving an impressive validation accuracy of 97.16%. The precision, recall, and F1 score of this IDS are 97.07%, 97.11%, and 97.05%, respectively. These numerical evaluations signify the superiority of this IDS. Apart from performance evaluated through machine learning performance evaluation standards, the proposed IDS stands out from the rest because of the analysis from the user perspective. It is a practical solution to IoT intrusion, which detects and defends intrusion within 1.14 s. The detection rate of this IDS has been optimized, and it ensures a minimum number of false alarms. As a result, it does not interrupt the intended service often. The faster response and 1.27% false alarm rate ensure an uninterrupted user experience. Considering the technical and utility perspective, the proposed LSTM-based IDS is one of the most effective solutions to IoT intrusion.

Author Contributions: All authors contributed to the writing of the article, designing of the work, collecting the data, analyzing the data, and approving the final version of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: There is no external funding for the research.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* **2013**, *29*, 1645–1660.
2. Faruqui, N.; Yousuf, M.A.; Whaiduzzaman, M.; Azad, A.; Alyami, S.A.; Liò, P.; Kabir, M.A.; Moni, M.A. SafetyMed: A Novel IoMT Intrusion Detection System Using CNN-LSTM Hybridization. *Electronics* **2023**, *12*, 3541. [CrossRef]
3. Bataev, A.V.; Zhuzhoma, I.; Bulatova, N.N. Digital Transformation of the World Economy: Evaluation of the Global and Russian Internet of Things Markets. In Proceedings of the 2020 9th International Conference on Industrial Technology and Management (ICITM), Oxford, UK, 11–13 February 2020; pp. 274–278.
4. Ayittey, F.K.; Ayittey, M.K.; Chiwero, N.B.; Kamasah, J.S.; Dzuvor, C. Economic impacts of Wuhan 2019-nCoV on China and the world. *J. Med. Virol.* **2020**, *92*, 473. [CrossRef]

5. Almazrouei, O.S.M.B.H.; Magalingam, P.; Hasan, M.K.; Shanmugam, M. A Review on Attack Graph Analysis for IoT Vulnerability Assessment: Challenges, Open Issues, and Future Directions. *IEEE Access* **2023**, *11*, 44350–44376. [CrossRef]
6. Santhosh Kumar, S.; Selvi, M.; Kannan, A. A comprehensive survey on machine learning-based intrusion detection systems for secure communication in internet of things. *Comput. Intell. Neurosci.* **2023**, *2023*, 8981988. [CrossRef]
7. Xu, H.; Sun, Z.; Cao, Y.; Bilal, H. A data-driven approach for intrusion and anomaly detection using automated machine learning for the Internet of Things. *Soft Comput.* **2023**, *27*, 14469–14481. [CrossRef]
8. Yi, L.; Yin, M.; Darbandi, M. A deep and systematic review of the intrusion detection systems in the fog environment. *Trans. Emerg. Telecommun. Technol.* **2023**, *34*, e4632.
9. Awajan, A. A novel deep learning-based intrusion detection system for IOT networks. *Computers* **2023**, *12*, 34. [CrossRef]
10. Musleh, D.; Alotaibi, M.; Alhaidari, F.; Rahman, A.; Mohammad, R.M. Intrusion Detection System Using Feature Extraction with Machine Learning Algorithms in IoT. *J. Sens. Actuator Netw.* **2023**, *12*, 29. [CrossRef]
11. Alkadi, S.; Al-Ahmadi, S.; Ben Ismail, M.M. Toward Improved Machine Learning-Based Intrusion Detection for Internet of Things Traffic. *Computers* **2023**, *12*, 148. [CrossRef]
12. Alazab, M.; Khurma, R.A.; Awajan, A.; Camacho, D. A new intrusion detection system based on Moth–Flame Optimizer algorithm. *Expert Syst. Appl.* **2022**, *210*, 118439. [CrossRef]
13. Chaganti, R.; Suliman, W.; Ravi, V.; Dua, A. Deep learning approach for SDN-enabled intrusion detection system in IoT networks. *Information* **2023**, *14*, 41. [CrossRef]
14. Henry, A.; Gautam, S.; Khanna, S.; Rabie, K.; Shongwe, T.; Bhattacharya, P.; Sharma, B.; Chowdhury, S. Composition of hybrid deep learning model and feature optimization for intrusion detection system. *Sensors* **2023**, *23*, 890. [CrossRef] [PubMed]
15. Fatani, A.; Dahou, A.; Abd Elaziz, M.; Al-Qaness, M.A.; Lu, S.; Alfadhli, S.A.; Alresheedi, S.S. Enhancing Intrusion Detection Systems for IoT and Cloud Environments Using a Growth Optimizer Algorithm and Conventional Neural Networks. *Sensors* **2023**, *23*, 4430. [CrossRef]
16. Bacevicius, M.; Paulauskaite-Taraseviciene, A. Machine Learning Algorithms for Raw and Unbalanced Intrusion Detection Data in a Multi-Class Classification Problem. *Appl. Sci.* **2023**, *13*, 7328. [CrossRef]
17. Alshahrani, H.; Khan, A.; Rizwan, M.; Reshan, M.S.A.; Sulaiman, A.; Shaikh, A. Intrusion Detection Framework for Industrial Internet of Things Using Software Defined Network. *Sustainability* **2023**, *15*, 9001. [CrossRef]
18. Alosaimi, S.; Almutairi, S.M. An Intrusion Detection System Using BoT-IoT. *Appl. Sci.* **2023**, *13*, 5427. [CrossRef]
19. Alotaibi, Y.; Ilyas, M. Ensemble-Learning Framework for Intrusion Detection to Enhance Internet of Things' Devices Security. *Sensors* **2023**, *23*, 5568. [CrossRef]
20. Larriva-Novo, X.; Sánchez-Zas, C.; Villagrá, V.A.; Marín-Lopez, A.; Berrocal, J. Leveraging Explainable Artificial Intelligence in Real-Time Cyberattack Identification: Intrusion Detection System Approach. *Appl. Sci.* **2023**, *13*, 8587. [CrossRef]
21. Fraihat, S.; Makhadmeh, S.; Awad, M.; Al-Betar, M.A.; Al-Redhaei, A. Intrusion detection system for large-scale IoT NetFlow networks using machine learning with modified Arithmetic Optimization Algorithm. *Internet Things* **2023**, *22*, 100819. [CrossRef]
22. Kaur, B.; Dadkhah, S.; Shoeleh, F.; Neto, E.C.P.; Xiong, P.; Iqbal, S.; Lamontagne, P.; Ray, S.; Ghorbani, A.A. Internet of things (IoT) security dataset evolution: Challenges and future directions. *Internet Things* **2023**, *22*, 100780. [CrossRef]
23. Sarker, I.H. Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Comput. Sci.* **2021**, *2*, 420. [CrossRef]
24. Awajan, A.; Alazab, M.; Alhyari, S.; Qiqieh, I.; Wedyan, M. Machine learning techniques for automated policy violation reporting. *Int. J. Internet Technol. Secur. Trans.* **2022**, *12*, 387–405. [CrossRef]
25. Liu, H.; Lang, B. Machine learning and deep learning methods for intrusion detection systems: A survey. *Appl. Sci.* **2019**, *9*, 4396. [CrossRef]
26. Khan, A.H.; Li, S.; Chen, D.; Liao, L. Tracking control of redundant mobile manipulator: An RNN based metaheuristic approach. *Neurocomputing* **2020**, *400*, 272–284. [CrossRef]
27. Apaydin, H.; Feizi, H.; Sattari, M.T.; Colak, M.S.; Shamshirband, S.; Chau, K.W. Comparative analysis of recurrent neural network architectures for reservoir inflow forecasting. *Water* **2020**, *12*, 1500. [CrossRef]
28. Achar, S.; Faruqui, N.; Whaiduzzaman, M.; Awajan, A.; Alazab, M. Cyber-Physical System Security Based on Human Activity Recognition through IoT Cloud Computing. *Electronics* **2023**, *12*, 1892. [CrossRef]
29. Rajagukguk, R.A.; Ramadhan, R.A.; Lee, H.J. A review on deep learning models for forecasting time series data of solar irradiance and photovoltaic power. *Energies* **2020**, *13*, 6623. [CrossRef]
30. Cao, B.; Li, C.; Song, Y.; Qin, Y.; Chen, C. Network intrusion detection model based on CNN and GRU. *Appl. Sci.* **2022**, *12*, 4184. [CrossRef]
31. Trivedi, S.; Patel, N.; Faruqui, N.; Tahir, S.B.u.d. Human Interaction and Classification Via K-ary Tree Hashing Over Body Pose Attributes Using Sports Data. In Proceedings of the International Conference on Hybrid Intelligent Systems, Online, 13–15 December 2022; pp. 366–378.
32. Faruqui, N.; Yousuf, M.A.; Chakraborty, P.; Hossain, M.S. Innovative automation algorithm in micro-multinational data-entry industry. In Proceedings of the Cyber Security and Computer Science: Second EAI International Conference, ICONCS 2020, Dhaka, Bangladesh, 15–16 February 2020; pp. 680–692.

33. Trivedi, S.; Tran, T.A.; Faruqui, N.; Hassan, M.M. An Exploratory Analysis of Effect of Adversarial Machine Learning Attack on IoT-enabled Industrial Control Systems. In Proceedings of the 2023 International Conference on Smart Computing and Application, Nashville, TN, USA, 26–30 June 2023; pp. 1–8.
34. Paula, L.P.O.; Faruqui, N.; Mahmud, I.; Whaiduzzaman, M.; Hawkinson, E.C.; Trivedi, S. A Novel Front Door Security (FDS) Algorithm using GoogleNet-BiLSTM Hybridization. *IEEE Access* **2023**, *11*, 19122–19134. [CrossRef]
35. Faruqui, N.; Yousuf, M.A.; Kateb, F.A.; Hamid, M.A.; Monowar, M.M. Healthcare As a Service (HAAS): CNN-based cloud computing model for ubiquitous access to lung cancer diagnosis. *Heliyon* **2023**, *9*, e21520. [CrossRef]
36. Zhang, G.; Zheng, L.; Su, Z.; Zeng, Y.; Wang, G. M-sequences and sliding window based audio watermarking robust against large-scale cropping attacks. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 1182–1195. [CrossRef]
37. Faruqui, N.; Kabir, M.A.; Yousuf, M.A.; Whaiduzzaman, M.; Barros, A.; Mahmud, I. Trackez: An IoT-based 3D-Object Tracking from 2D Pixel Matrix using Mez and FSL Algorithm. *IEEE Access* **2023**, *11*, 61453–61467. [CrossRef]
38. Khan, A.Y.; Latif, R.; Latif, S.; Tahir, S.; Batool, G.; Saba, T. Malicious insider attack detection in IoTs using data analytics. *IEEE Access* **2019**, *8*, 11743–11753. [CrossRef]
39. Hodo, E.; Bellekens, X.; Hamilton, A.; Dubouilh, P.L.; Iorkyase, E.; Tachtatzis, C.; Atkinson, R. Threat analysis of IoT networks using artificial neural network intrusion detection system. In Proceedings of the 2016 International Symposium on Networks, Computers and Communications (ISNCC), Hammamet, Tunisia, 11–13 May 2016; pp. 1–6.
40. Zhou, Q.; Xu, Q.; Zeng, P.; Zhao, K.; Yuan, S. Scenario-based quantitative human vulnerability assessment of site-specific landslides using a probabilistic model. *Landslides* **2022**, *19*, 993–1008. [CrossRef]
41. Patel, N.; Trivedi, S.; Faruqui, N. An Innovative Deep Neural Network for Stress Classification in Workplace. In Proceedings of the 2023 International Conference on Smart Computing and Application (ICSCA), Hail, Saudi Arabia, 5–6 February 2023; pp. 1–5.
42. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1026–1034.
43. Trivedi, S.; Patel, N.; Faruqui, N. Bacterial Strain Classification using Convolutional Neural Network for Automatic Bacterial Disease Diagnosis. In Proceedings of the 2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Online, 27–28 January 2023; pp. 325–332.
44. Trivedi, S.; Patel, N.; Faruqui, N. A Novel Lightweight Lung Cancer Classifier Through Hybridization of DNN and Comparative Feature Optimizer. In Proceedings of the International Conference on Hybrid Intelligent Systems, Online, 13–15 December 2022; pp. 188–197.
45. Mansfield, S.; Veenstra, K.; Obraczka, K. TerrainLOS: An outdoor propagation model for realistic sensor network simulation. In Proceedings of the 2016 IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), London, UK, 19–21 September 2016; pp. 463–468.
46. Chatterjee, M.; Namin, A.S.; Datta, P. Evidence fusion for malicious bot detection in IoT. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 4545–4548.
47. Alzubi, O.A.; Alzubi, J.A.; Alazab, M.; Alrabea, A.; Awajan, A.; Qiqieh, I. Optimized machine learning-based intrusion detection system for fog and edge computing environment. *Electronics* **2022**, *11*, 3007. [CrossRef]
48. Alazab, M.; Abu Khurma, R.; Awajan, A.; Wedyan, M. Digital forensics classification based on a hybrid neural network and the salp swarm algorithm. *Electronics* **2022**, *11*, 1903. [CrossRef]
49. Alazab, M.; Awajan, A.; Mesleh, A.; Abraham, A.; Jatana, V.; Alhyari, S. COVID-19 prediction and detection using deep learning. *Int. J. Comput. Inf. Syst. Ind. Manag. Appl.* **2020**, *12*, 168–181.
50. Almaiah, M.A.; Almomani, O.; Alsaaidah, A.; Al-Otaibi, S.; Bani-Hani, N.; Hwaitat, A.K.A.; Al-Zahrani, A.; Lutfi, A.; Awad, A.B.; Aldhyani, T.H. Performance investigation of principal component analysis for intrusion detection system using different support vector machine kernels. *Electronics* **2022**, *11*, 3571. [CrossRef]
51. Cui, J.; Zong, L.; Xie, J.; Tang, M. A novel multi-module integrated intrusion detection system for high-dimensional imbalanced data. *Appl. Intell.* **2023**, *53*, 272–288. [CrossRef]
52. Barhoush, M.; Abed-alguni, B.H.; Al-qudah, N.E.A. Improved discrete salp swarm algorithm using exploration and exploitation techniques for feature selection in intrusion detection systems. *J. Supercomput.* **2023**, *79*, 21265–21309. [CrossRef]
53. Dwibedi, S.; Pujari, M.; Sun, W. A comparative study on contemporary intrusion detection datasets for machine learning research. In Proceedings of the 2020 IEEE International Conference on Intelligence and Security Informatics (ISI), Arlington, VA, USA, 9–10 November 2020; pp. 1–6.