IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 18 April 2024, accepted 30 April 2024, date of publication 3 May 2024, date of current version 10 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3396461

RESEARCH ARTICLE

Deep-IDS: A Real-Time Intrusion Detector for IoT Nodes Using Deep Learning

SANDEEPKUMAR RACHERLA[1], (Senior Member, IEEE), PRATHYUSHA SRIPATHI[1], (Member, IEEE), NURUZZAMAN FARUQUI[2], MD ALAMGIR KABIR[3], (Member, IEEE), MD WHAIDUZZAMAN[4], (Senior Member, IEEE), AND SYED AZIZ SHAH[5]

[1]Amazon, Seattle, WA 98109, USA
[2]Department of Software Engineering, Daffodil International University, Daffodil Smart City, Birulia, Dhaka 1216, Bangladesh
[3]Division of Computer Science and Software Engineering, Mälardalen University, 721 23 Västerås, Sweden
[4]School of Information Systems, Queensland University of Technology, Brisbane, QLD 4000, Australia
[5]Centre for Intelligent Healthcare, Coventry University, CV1 5FB Coventry, U.K.

Corresponding author: Md Alamgir Kabir (md.alamgir.kabir@mdu.se)

ABSTRACT The Internet of Things (IoT) represents a swiftly expanding sector that is pivotal in driving the innovation of today's smart services. However, the inherent resource-constrained nature of IoT nodes poses significant challenges in embedding advanced algorithms for cybersecurity, leading to an escalation in cyberattacks against these nodes. Contemporary research in Intrusion Detection Systems (IDS) predominantly focuses on enhancing IDS performance through sophisticated algorithms, often overlooking their practical applicability. This paper introduces Deep-IDS, an innovative and practically deployable Deep Learning (DL)-based IDS. It employs a Long-Short-Term-Memory (LSTM) network comprising 64 LSTM units and is trained on the CIC-IDS2017 dataset. Its streamlined architecture renders Deep-IDS an ideal candidate for edge-server deployment, acting as a guardian between IoT nodes and the Internet against Denial of Service, Distributed Denial of Service, Brute Force, Man-in-the-Middle, and Replay Attacks. A distinctive aspect of this research is the trade-off analysis between the intrusion Detection Rate (DR) and the False Alarm Rate (FAR), facilitating the real-time performance of the Deep-IDS. The system demonstrates an exemplary detection rate of 96.8% at the 70% threshold of DR-FAR trade-off and an overall classification accuracy of 97.67%. Furthermore, Deep-IDS achieves precision, recall, and F1-scores of 97.67%, 98.17%, and 97.91%, respectively. On average, Deep-IDS requires 1.49 seconds to identify and mitigate intrusion attempts, effectively blocking malicious traffic sources. The remarkable efficacy, swift response time, innovative design, and novel defense strategy of Deep-IDS not only secure IoT nodes but also their interconnected sub-networks, thereby positioning Deep-IDS as a leading IDS for IoT-enhanced computer networks.

INDEX TERMS Network security, deep learning, intrusion-detection system (IDS), Internet of Things (IoT), LSTM, response mechanism, intrusion detection rate.

I. INTRODUCTION

By 2025, the number of IoT devices is expected to exceed 41 billion [1]. IoT technology lies at the core of modern smart services, encompassing smart healthcare systems, smart homes, smart cities, smart offices, and smart manufacturing industries [2]. Its rapid integration across various sectors not only fuels its growth but also increasingly attracts the attention of cybercriminals [3], leading to a significant rise in the rate of intrusions on IoT devices [4]. Keeping pace

The associate editor coordinating the review of this manuscript and approving it for publication was Giovanni Pau.

with the swift advancements in the IoT sector, cybercriminals are constantly developing new strategies for intrusion [5]. Addressing the dynamic nature of these threats necessitates a real-time IDS designed for efficient detection. This paper introduces Deep-IDS, a real-time IDS that boasts remarkable performance and an impressive detection rate.

The Deep-IDS is an edge-server-based system that sits between the IoT network and the Internet. Under the hood, an optimally designed LSTM network trained with the intrusion features of five frequently occurring IoT intrusions [6] classifies the incoming network packets into one of the six classes. The system allows traffic flow from sources classified as benign and discards others. The incorporation of an effective threshold, discovered by empirical experiments, ensures a balance between the detection and false alarm rates and draws a distinct line separating the proposed Deep-IDS from similar approaches. It has been designed to detect intrusions from an array of IoT nodes consisting of sensing, processing, communication, and power subsystems. The overall design of Deep-IDS ensures insignificant detection delay, enabling it to detect intrusion in real-time.

The proposed Deep-IDS performs better than most of the IDSs developed using similar technology [7]. A well-engineered network trained with relevant intrusion features and equipped with an array of simplified IoT nodes has contributed to this achievement. Moreover, the empirical approach of threshold selection has ensured the optimal trade-off between the False Alarm Rate (FAR) and the detection rate, making the Deep-IDS an effective solution to secure IoT networks. Furthermore, it strengthens security by blocking malicious traffic sources. The unique contributions of Deep-IDS have been listed below:

- **Detection Quality:** The Deep-IDS detects intrusions with 97.67% accuracy. Besides, the precision, recall, and F1-scores are 97.67%, 98.17%, and 97.91%, respectively.
- **Detection Rate:** The proposed IDS has a real-time detection rate of 96.8% with only 1.49 seconds of response delay.
- **FAR Minimization:** It uses an innovative empirical approach that minimizes the false alarm rate by discovering the optimal trade-off point between the detection rate and the false alarm rate.

The remainder of this paper is structured into six sections. The second section provides an overview of the relevant literature. The third section outlines the methodology. In the fourth section, we discuss the implementation process and the response mechanism of the proposed IDS. The fifth section is dedicated to presenting the performance evaluations and analysis. The sixth section explores the limitations and future directions of this research. Finally, the paper concludes in the seventh section.

II. LITERATURE REVIEW

Machine Learning (ML) and Deep Learning (DL) algorithms have become pivotal in enhancing network security across various domains, including Wireless Sensor Networks (WSNs) [8], Online Social Networks (OSNs) [9], Software Defined Networks (SDNs) [10], and IoT networks [11]. Particularly, the field of intrusion detection within IoT networks emerges as a dynamic area of research, reflecting the critical need for robust security measures [12]. Azumah et al. [13] proposed a sophisticated approach utilizing deep LSTM networks to identify intrusions in IoT devices for smart homes. Furthermore, Ahsan et al. [14] explored the efficacy of DenseNet, CNN, and a combined CNN-LSTM model in detecting DDoS attacks, showcasing the versatility of DL methodologies. Additionally, the research conducted by Yadav et al. [15] examines the detection of malicious traffic in IoT devices connected to 5G networks, employing a novel combination of Artificial Neural Networks (ANN) and ML classifiers. The investigation of Banaamah et al. [16] into various DL-based strategies for intrusion detection underscores the significant potential of these technologies to fortify IoT network security. The recent advancements in applying DL methods in intrusion detection have inspired the proposed Deep-IDS.

The DL-based IDS developed by Ashiku et al. [17], utilizing the UNSW-NB15 dataset, reports an impressive intrusion detection accuracy of 94.40%. Musleh et al. [18] innovatively applied feature extraction techniques using VGG-16 and DenseNet on intrusion datasets and, through the employment of ML models such as Random Forest, K-Nearest Neighbors, and Support Vector Machine (SVM), achieved an accuracy of 92.40%. Other notable IDSs, including those developed by Logeswari et al. [19], Mebawondu et al. [20], and Abdelkhalek et al. [21], have reported accuracies of 82.20%, 76.96%, and 83.50%, respectively. Moreover, the Secured Automatic Two-level IDS (SATIDS) introduced by Elsayed et al. [22] showcases a remarkable accuracy of 96.56%. Another innovative DL-based framework by Kumar et al. [23] advances the field further by achieving an intrusion detection accuracy of 97.45%. Outperforming these approaches, the proposed Deep-IDS demonstrates a superior accuracy of 97.67%.

The systematic review paper by Saied et al. [24] shows that most DL-based IDSs are detection accuracy-centric, leaving a significant research gap for improving the detection rate, reducing detection delay, and false positive rate minimization. None of the methods presented earlier conducted experiments to minimize the FAR, which has been done in the proposed LSTM-based Deep-IDS. Altunay et al. [25], Bakhsh [26], Elsayed [22] et al., Chaganti et al. [27], and many other studies used LSTM network for IoT intrusion detection similar to the proposed method. However, most state-of-the-art systems ignored the impact of LSTM network simplicity to perform real-time intrusion detection with insignificant IDS response delay [28]. It is a major research gap in this field. A simplified LSTM network architecture has been designed for the proposed Deep-IDS to minimize the IDS response delay and make it an effective real-time IDS.

The Whale Integrated LSTM (WILS) framework for intrusion detection by othi et al. [29], hybrid metaheuristics-deep learning based IDS by Sanju et al. [30], Enhanced LSTM (ELSTM) and Recurrent Neural Network (RNN) combination based IDS developed by Donkol et al. [31] are few of the sophisticated IDS development approaches with promising performance. These complex methods are focused on detection performance and ignore practical applications. It is another research gap in the existing IDS research field. The proposed Deep-IDS not only detects intrusions but also controls the traffic flow from the malicious source to protect the IoT node. That is why it is a practical approach to IoT IDS development.

III. METHODOLOGY

This section presents the methodology used to develop the proposed Deep-IDS. It starts by analyzing the most effective network for developing the IDS. Then, the dataset is thoroughly studied to understand its features. Subsequently, feature extraction and processing methods are developed. After that, the IoT node architecture is designed, and its corresponding communication protocols are studied. Finally, the LSTM network architecture, training process, and optimization methods have been presented.

A. DEEP LEARNING MODEL ANALYSIS & SELECTION

DL approach utilizes artificial neural networks with multiple layers for learning complex patterns in data [32]. That is why it is a good fit for understanding and effectively detecting intricate attack patterns within network traffic. According to Alsoufi et al. [33], multiple DL models work well in IoT intrusion detection. However, CNNs [34], Recurrent Neural Networks (RNNs) [35], and Long Short-Term Memory (LSTM) networks [11] stand out from the rest. This section explores the conceptual background of these models to identify an appropriate model for IoT intrusion detection.

1) CONVOLUTIONAL NEURAL NETWORKS

Intrusions in a computer network through IoT come from many different data structures. If it is a grid-like data structure, CNN is effective in classifying them [36]. A CNN is defined by equation 1, consisting of an input layer, multiple hidden layers, and an output layer [37]. The hidden layers include convolutional, pooling, and fully connected layers. The primary advantage of CNNs is their ability to learn hierarchical feature representations from raw data automatically [38]. The computer network intrusion pattern frequently changes. And it is necessary to retrain the models. The CNN effectively retrieves features from new intrusion patterns.

$$Y_{ij} = (X * K)_{ij} = \sum_m \sum_n X_{(i+m)(j+n)} K_{mn} \tag{1}$$

In the equation 1, X is the convolution operation, K is the kernel of $n \times n$, Y_{ij} is the output feature map at position (i, j). The kernel slides over the input data, computing the element-wise multiplication followed by a summation to produce the output feature map.

2) RECURRENT NEURAL NETWORKS (RNNs)

When classifying from short sequential data, the RNN performs better than most of the DL models [39]. Unlike feedforward networks, RNNs maintain an internal state that captures information about previous time steps in the input sequence. This internal state allows RNNs to model temporal dependencies in data effectively [40]. These characteristics of RNN make it a potential DL model for intrusion detection. Mathematically, the RNN is defined as equation 2.

$$h_t = \sigma(W_{hh}h_{t-1} + W_{xh}x_t + b_h) \tag{2}$$

$$y_t = W_{hy}h_t + b_y \tag{3}$$

In equation 2, x_t is the input at time step t, h_t is the hidden state, y_t is the output, W_{hh}, W_{xh}, and W_{hy} are the weight matrices, and b_h and b_y are the biases. The activation function $\sigma(\cdot)$ is typically a nonlinear function such as the hyperbolic tangent or ReLU.

3) LONG SHORT-TERM MEMORY (LSTM) NETWORKS

The RNN suffers from performance issues for long sequences. At the same time, the vanishing gradient imposes additional challenges for this model [41]. The LSTMs are a type of RNN designed to overcome the vanishing gradient problem [42]. LSTMs introduce a memory cell and three gating mechanisms. The first gate is the input gate defined by equation 4. The second gate is the forget gate, which is expressed by equation 5. And the last gate is the output gate conceptualized by equation 6. These gates control the flow of information within the LSTM cell, allowing it to learn and retain long-term dependencies effectively [43]. As a result, it can be trained to detect intrusion from the lengthy sequential signals of a computer network.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{4}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{5}$$

$$o_t = sigma(W_o[h_{t-1}, x_t] + b_o) \tag{6}$$

The update and candidate cell states are defined by equations 7 and 8, respectively.

$$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C) \tag{7}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{8}$$

The final hypothesis of the LSTM network is defined by equation 9

$$h_t = o_t \odot \tanh(C_t) \tag{9}$$

In equations, 4, 5, 6, the f_t, i_t, and o_t represent the forget, input, and output gates, respectively. C_t and $\tilde{C}_t$ denote the updated and candidate cell states, respectively. The weight matrices W_f, W_i, W_o, and W_C, and the bias vectors b_f, b_i,

```
2023-04-20 08:00:00, 192.168.1.2, 203.0.113.42, 52345,
80, TCP, 1500, 12 2023-04-20 08:00:01, 192.168.1.3,
198.51.100.16, 63212, 443, TCP, 1400, 8 2023-04-20
08:00:02, 192.168.1.2, 203.0.113.42, 52345, 80, TCP,
1500, 5 2023-04-20 08:00:03, 192.168.1.4, 192.0.2.55,
49160, 123, UDP, 48, 1
```

FIGURE 1. An example of a network sequential data.

b_o, and b_C are the model parameters. The symbol $\odot$ denotes element-wise multiplication.

These deep learning techniques can be adapted and fine-tuned for intrusion detection tasks, taking advantage of their unique capabilities to process and learn from complex network data.

4) SELECTION OF AN APPROPRIATE DEEP LEARNING MODEL

Selecting an appropriate DL model for intrusion detection in IoT networks depends on the data's nature and the task's specific requirements. Each model discussed (CNN, LSTM, RNN) has advantages and limitations. However, the selection criteria hinge upon the feature alignment of the dataset with the networks. The network data are long sequences as presented in figure 1.

The features ideal for training CNN, RNN, and LSTM are defined in equations 10, 11, and 12, respectively. Here, in equation 10, x, y represents 2D spatial data, R is a set of real numbers, and C_f is the set of CNN features. The S_D, V_L, and T_D in equations 11 and 12 are sets of sequential dependence data, variable length data, and data with temporal dynamics, respectively [44].

$$C_f = \{C_f \in \{x, y\} | x \in R, y \in R\} \tag{10}$$

$$R_f = \left\{R_f \in F \,\middle|\, \text{Property}(f) = S_D \vee V_L \vee T_D\right\} \tag{11}$$

$$L_f = \left\{L_f \in F \,\middle|\, \text{Property}(f) = S_D \vee V_L \vee T_D\right\} \tag{12}$$

The ideal features for RNN and LSTM networks are similar. However, LSTM networks are capable of handling more extensive sequences than RNN. This relation is defined by equations 13. On the other hand, CNN's features are exclusive to those of LSTM and RNN, which is expressed in equation 14. The mathematical relations among the ideal features to train CNN, RNN, and LSTM conclude that the LSTM network is an appropriate choice for developing the proposed Deep-IDS [45].

$$R_f \subset L_f \tag{13}$$

$$\{C_f \cap R_f\} \cup \{C_f \cap L_f\} = \phi\} \tag{14}$$

The CNN performs best for the grid data structure. The RNN suffers from vanishing gradient issues with long sequences. The LSTM networks perform well with sequential data. It is not affected by vanishing gradient problems, no matter how long the sequence is. The proposed methodology is a real-time network security system. That is why training and inference time are crucial factors to consider. RNNs have faster training times than LSTMs due to their simpler architecture. However, LSTMs can perform better, especially when dealing with long input sequences. The LSTM network achieved 97.67% classification accuracy, while RNN ended up with 91.44% in experimental analysis. CNNs can have longer training times due to many parameters, but they are highly parallelizable, making them suitable for GPU-accelerated training. The LSTM has been selected as the deep learning model for computer network intrusion detection by analyzing these limitations and advantages [46].

B. DATASET DESCRIPTION

The CIC-IDS2017 dataset, created by the Canadian Institute for Cybersecurity, has been used in this experiment. It was collected in 2017 and includes a variety of attack types relevant to the IoT and modern network environments [47]. It contains approximately 2.8 million instances, with a balance of benign and intrusion records. The dataset is divided into multiple files for each day of the week, allowing researchers to choose subsets of the data for their experiments. The raw dataset contains duplicate values and missing values. It also includes classes irrelevant to IoT nodes. After removing duplicate values, missing values, and irrelevant labels, the final dataset has 56,662 instances. This clean dataset has been split into training, testing, and validation sets with a ratio of 70:15:15, respectively. After splitting, there are 39,663 data for training, 8499 for testing, and 8499 for validating.

1) FEATURES AND VARIABLES

The CIC-IDS2017 dataset comprises 78 features and a class label for each instance. These features include various network traffic attributes, such as flow duration, source and destination IP addresses, source and destination ports, protocol, flow bytes, and packet-related statistics. Each instance in the dataset is labeled as benign or one of the several types of attacks, including DoS, DDoS, BRF, Infiltration, and more.

2) SAMPLE DATASET TABLE

The CIC-IDS2017 dataset has 15 classes. One is for benign traffic, and the rest of the fourteen are different types of intrusions. A sample of the CIC-IDS2017 dataset is shown in Table 1.

C. FEATURE EXTRACTION AND PREPROCESSING

Feature extraction and preprocessing for LSTM are crucial steps in developing an effective IDS. It has been done using Algorithm 1. The quality and structure of the input data heavily influence the LSTM model's performance. Ensuring the dataset is preprocessed correctly can lead to more accurate and efficient training, ultimately resulting in a more robust and reliable model [48]. By normalizing the features, handling categorical variables, selecting relevant features,

TABLE 1. Sample of CIC-IDS2017 dataset with instances for each label.

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
4	192.168.10.5	22	192.168.10.50	57165	6		SSH-Patator
3	192.168.10.5	80	192.168.10.50	57166	6		DoS slowloris
2	192.168.10.5	80	192.168.10.50	57167	6		DoS Slowhttptest
1	192.168.10.5	80	192.168.10.50	57168	6		DoS Hulk
0	192.168.10.5	80	192.168.10.50	57169	6		DoS GoldenEye
0	192.168.10.5	443	192.168.10.50	57170	6		Heartbleed
3	192.168.10.5	80	192.168.10.50	57171	6		Web Attack – Brute Force
2	192.168.10.5	80	192.168.10.50	57172	6		Web Attack – XSS
1	192.168.10.5	80	192.168.10.50	57173	6		Web Attack – Sql Injection
0	192.168.10.5	80	192.168.10.50	57174	6		Infiltration
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
0	192.168.10.5	80	192.168.10.50	57177	6		DDoS

and generating input sequences, we ensure that the LSTM model can effectively capture the underlying patterns and relationships in the data [49]. This, in turn, allows the model to generalize well to unseen data, providing a high intrusion detection accuracy and minimizing false alarms.

Algorithm 1 Feature Extraction and Preprocessing for LSTM-Based IDS

```
Require: Dataset $D$, Window size $w$
Ensure: Preprocessed dataset $D_{seq}$ for LSTM
Normalization:
for $f_i \in D$ do                                   ▷ Each feature $f_i$
    $f_{i_{norm}} \leftarrow \frac{f_i - \min(f_i)}{\max(f_i) - \min(f_i)}$
end for
Categorical Features:
for $c_i \in D$ do                                   ▷ Each categorical $c_i$
    $c_{i_{encoded}} \leftarrow$ one-hot($c_i$)
end for
Feature Selection:
$MI \leftarrow$ Compute MI scores for $f_i$ against labels $Y$
Select $f_i$ with MI above threshold
Sequence Generation:
$D_{seq} \leftarrow \emptyset$
for $t = w$ to $|D|$ do
    $X_{seq} \leftarrow (X_{t-w+1}, \ldots, X_t)$
    $Y_{target} \leftarrow Y_{t+1}$
    $D_{seq} \leftarrow D_{seq} \cup \{(X_{seq}, Y_{target})\}$
end for
return $D_{seq}$
```

1) NORMALIZATION

Since LSTMs are sensitive to the scale of input features, it is essential to normalize the dataset. This experiment uses the min-max normalization method to scale appropriate features between 0 to 1 [50]. The process is defined in equation 15. In the equation 15, X represents a feature value, and X_{min} and X_{max} are the minimum and maximum values of the feature, respectively.

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{15}$$

2) HANDLING CATEGORICAL FEATURES

Some of the features, for example, IP address, type of protocol, etc., are non-scalable. These are categorical data. In this experiment, we encoded them using one-hot encoding governed by equation 16 [51]. In this equation, $P_{encoded}$ is the one-hot encoded vector for protocol types, and p_i is a binary value representing the presence (1) or absence (0) of the i-th protocol type.

$$P_{encoded} = [p_1, p_2, \ldots, p_n] \tag{16}$$

3) FEATURE SELECTION

Not every feature available in a dataset is relevant to the classification process [52]. It is essential to select the relevant feature to reduce the complexity of the model and improve computational efficiency. In this experiment, the Mutual Information (MI) [53] approach has been used to rank features based on their relevance to the target variable. The process is expressed using equation 17.

$$MI(X, Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \tag{17}$$

In equation 17, X and Y are the feature and label variables, and $p(x, y)$, $p(x)$, and $p(y)$ are their joint and marginal probabilities, respectively. Features with higher MI scores are considered more relevant for the task.

4) SEQUENCE GENERATION

The LSTM networks are good at learning from and predicting labels from sequential data. This is the fundamental reason for choosing LSTM for this research. The training dataset is well-structured. However, they are not structured as network data sequences. The sliding window approach is defined by equation 18, which has been used to convert the tabular data into sequences. Here w is the window size, X_t and Y_{t+1} are

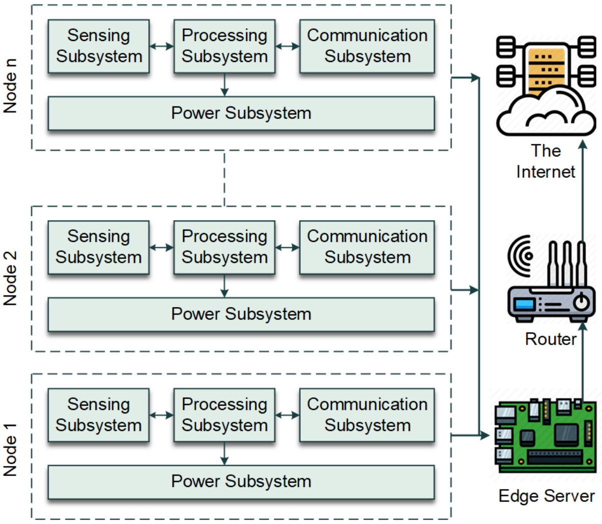


FIGURE 2. IoT node architecture.

the input and output features at time step t, respectively [54].

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

D. IOT NODE ARCHITECTURE AND COMMUNICATION PROTOCOLS

The proposed IDS works at the IoT node. Understanding the IoT node architecture and communication protocol is essential for the seamless integration of the proposed IDS. The IoT node architecture, associated communication protocols, and possible security vulnerabilities have been discussed in this section.

1) IoT NODE ARCHITECTURE

The IoT node architecture, illustrated in figure 2, consisted of sensing, processing, and communication subsystems. Each node has an additional power subsystem to power up the system. The sensors sense the environment and collect data. The processing subsystem consists of the microcontroller. It is responsible for processing and controlling the node's operations. Finally, the communication subsystem enables data transmission between the IoT node and other devices or networks. Usually, the IoT nodes are connected to edge servers. The edge server communicates with the WiFi router, which is a gateway [55]. The router is connected to the Internet.

The intrusion at the IoT node affects the performance of the devices. It is one of the significant indicators for detecting probable intrusion. In this paper, the performance of an IoT node has been characterized by its energy consumption, latency, and throughput. The energy consumption E of an IoT node as a function of its active time T_{active}, idle time T_{idle}, and power consumption during active and idle states, P_{active} and P_{idle}, respectively. The relationship among these entities is expressed by equation 19.

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

Latency and throughput have been used as two prominent indicators of IoT performance. Deviation in latency and throughput than usual is an indicator of intrusion. Latency L has been defined as the time required to complete a specific task, such as data acquisition, processing, or transmission. Throughput Th is the rate at which an IoT node can process or transmit data, measured in bits per second (bps). The relation between latency and throughput is defined by equation 20 where D is the amount of data processed or transmitted, and T is the time duration.

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

2) COMMUNICATION PROTOCOLS

Protocols govern communication over the Internet. IoT communication is not an exception. The IoT nodes use various communication protocols to exchange data with other devices or networks. Many intrusion uses protocol-level vulnerabilities. That is why secured communication protocols are essential for IoT devices [56]. In this paper, the secured communication protocols have been studied to improve the detection rate of the proposed IDS. The protocols analyzed have been listed in list [57]. These protocols have also been used in a wide range of applications as well.

- IEEE 802.15.4 (e.g., Zigbee, Thread): low-power, short-range communication
- Bluetooth Low Energy (BLE): energy-efficient, short-range communication
- LoRaWAN: long-range, low-power wide area network communication
- Cellular IoT (e.g., NB-IoT, LTE-M): long-range communication using cellular networks

The performance of these communication protocols has been characterized by transmission range, data rate, and energy consumption. The energy consumption expressed as E_c of a communication module during an active communication is a function of the transmission power P_{tx}, the data rate R, and the amount of data D. The relation among these variables is expressed in equation 21.

$$E_c = \frac{P_{tx} \cdot D}{R} \tag{21}$$

The transmission ranges R_{tx} have been estimated using the Friis transmission equation defined by equation 22 [58].

$$P_{rx} = \frac{P_{tx} \cdot G_{tx} \cdot G_{rx} \cdot \lambda^2}{(4\pi)^2 \cdot d^2} \tag{22}$$

In equation 22, P_{rx} is the received power, G_{tx} and G_{rx} are the transmitter and receiver gains, λ is the wavelength of the signal, and d is the distance between the transmitter and receiver. The maximum transmission range is estimated by solving for d when P_{rx} is at the minimum detectable level in equation 22.

3) SECURITY VULNERABILITIES IN IoT PROTOCOLS

IoT protocols are designed to enable efficient communication between IoT devices and networks. Many studies show that cybercriminals study the protocols and discover the vulnerabilities to perform successful attacks on IoT devices [3]. The experiment presented in this paper has been conducted within the scope of the CIC-IDS2017 dataset. The potential vulnerabilities have been studied within the boundaries of this dataset and IoT protocols to specify the intrusions to detect.

a: COMMON SECURITY VULNERABILITIES

The CIC-IDS2017 dataset has 14 different intrusion patterns. However, it is a comprehensive dataset, not exclusive to IoT. That is why every intrusion is not applicable within the context of this research. The core focus of this experiment is to detect five frequently attempted intrusions at the IoT end. These intrusions are DoS, DDoS, BRF, MITM, and RP Attacks. A probabilistic model governs each intrusion at the IoT node. The proposed IDS needs to align with the probabilistic model of successful attacks explained in section III-D3b [57]. While quantum vulnerabilities are beyond the scope of this paper, it's important to note that the advent of quantum computing presents new challenges in cybersecurity. With their potential to break traditional encryption methods, Quantum computers could make current security measures obsolete, introducing a new layer of complexity in protecting against intrusions, including those targeted in this research [59].

b: PROBABILISTIC MODEL FOR SUCCESSFUL ATTACKS

The probability model for a successful attack is formulated with multiple factors. The most relevant and common factors are the attacker's capabilities, the protocol's security features, and the network's defense mechanisms [60]. The probabilistic model is defined in equation 23. Here $P(A)$ represents the probability of a successful attack, $P(C)$ represents the probability of the attacker's capabilities, $P(S)$ represents the probability of the protocol's security features, and $P(D)$ represent the probability of the network's defense mechanisms.

$$P(A|C,S,D) = \frac{P(A,C,S,D)}{P(C,S,D)} \tag{23}$$

In equation 23, $P(A,C,S,D)$ is the joint probability of a successful attack, the attacker's capabilities, the protocol's security features, and the network's defense mechanisms. The $P(C,S,D)$ is the joint probability of the attacker's capabilities, the protocol's security features, and the network's defense mechanisms. By estimating these probabilities, the vulnerability of IoT protocols to various attacks is accessed.

E. PROPOSED DEEP-IDS

The proposed Deep-IDS is developed using an LSTM network operated from an edge server. This section presents the LSTM network architecture, training progress, weight initialization method, and weight optimization algorithm.

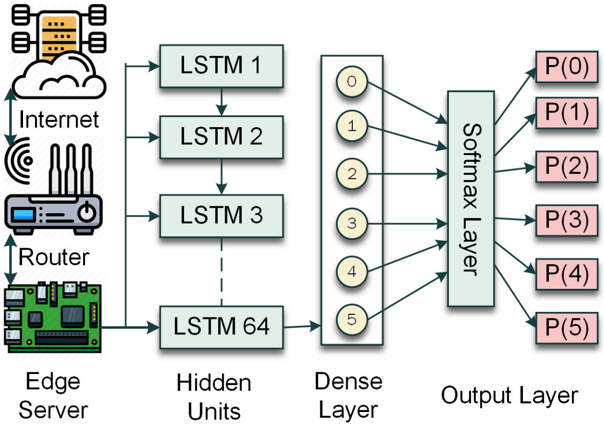


FIGURE 3. The LSTM network architecture.

1) LSTM NETWORK ARCHITECTURE

An LSTM network, illustrated in figure 3, with 64 hidden units followed by a Dense output layer with six nodes, is at the heart of the proposed Deep-IDS. The input layer of the experimenting LSTM network receives data with timesteps and feature pairs. The timesteps are the number of steps in the input data. The features are the number of features for each step. The input data are processed through the 64 hidden units. These units have internal memory cells. The temporal dependencies of the sequential data are stored in these memory cells. After processing, the data entered the dense layer. There are six nodes in the dense layer. The output from the dense layer is processed through the softmax function to convert the output into a probability value for each class.

2) TRAINING THE NETWORK

The proposed LSTM network has been trained with 39,663 instances. The training progress of the proposed network is illustrated in figure 4. The training is completed after ten epochs. In each epoch, there are 429 iterations. It takes 3 minutes and 21 seconds to complete the training. The training and validation accuracy sharply rises till the second epoch. After that, the learning curves maintain smooth near-linear characteristics. The training and validation loss demonstrate similar but inverse behavior. They rapidly reduce up to the second iteration. After that, the curves maintain near-linear characteristics.

a: WEIGHT INITIALIZATION

The initialization of weights is critical in successfully training deep neural networks [61]. Appropriate initialization of weight parameters accelerates the convergence rate and mitigates gradient issues, including vanishing or exploding gradients [62]. In this experiment, the He initialization has been used. He Initialization, also known as He-Normal initialization, was proposed by Kaiming He et al. [63] in 2015. The design of this particular model is tailored to

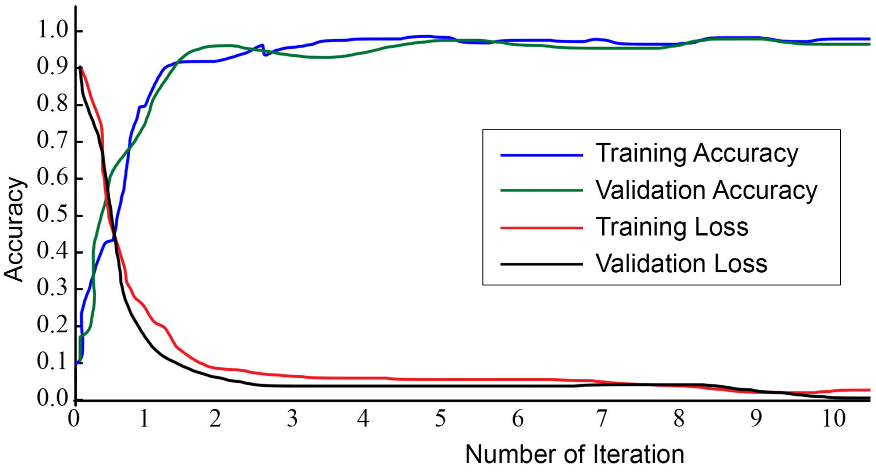


FIGURE 4. **Learning progress.**

accommodate deep neural networks that utilize Rectified Linear Unit (ReLU) [64] activation functions, as well as their variations, including Leaky ReLU and Parametric ReLU. While the hyperbolic tangent (tanh) activation function is commonly utilized in LSTM networks, it has been observed that He Initialization remains efficacious even when implemented in conjunction with ReLU or its derivatives. The primary idea behind He Initialization is to draw the initial weights from a Gaussian distribution with a mean of 0 and a variance of $\frac{2}{n_{\text{input}}}$, where n_{input} is the number of input units in the weight tensor. The layer's weights W are initialized as expression 24.

$$W \sim \mathcal{N}\left(0, \sqrt{\frac{2}{n_{\text{input}}}}\right) \tag{24}$$

In equation 24, the $\mathcal{N}(0, \sqrt{\frac{2}{n_{\text{input}}}})$ represents a Gaussian distribution [65] with a mean of 0 and a standard deviation of $\sqrt{\frac{2}{n_{\text{input}}}}$. The justification for employing this particular initialization technique is its ability to preserve the weights' variance throughout the forward and backward propagation phases. This, in turn, mitigates the risk of encountering vanishing or exploding gradient issues that may arise from excessively small or large gradients.

b: OPTIMIZATION ALGORITHM

The ADAM optimization algorithm is widely utilized for optimizing deep learning models during training [66]. The method under consideration amalgamates the benefits of the Adaptive Gradient Algorithm (AdaGrad) [67] and Root Mean Square Propagation (RMSProp) [68] by preserving distinct adaptive learning rates for individual weights and revising them by the first and second moments of the gradients. In the equation 25, g_t represents the gradient at time step t. Once the gradient is obtained, the first and second momentum is calculated using its value by equations 26 and 27, respectively. The β_1 and β_2 are the exponential decay rates for the moments in these equations.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{25}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{26}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{27}$$

The bias-corrected first and second-moment estimates are denoted by $\hat{m}_t$ and $\hat{v}_t$, respectively, in equations 28 and 29. The learning rate α and a small constant ϵ are used to prevent division by zero. The updated weight at time step t is represented by θ_t.

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{28}$$

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{29}$$

The proposed LSTM network's suitability for the ADAM optimization algorithm is attributed to its adaptability and capacity to manage sparse gradients. The approach exhibits computational efficiency, necessitates minimal memory, and broadly applies to diverse deep learning models, encompassing non-convex optimization problems. The utilization of the ADAM optimization algorithm guarantees the swift convergence and proficient performance of the proposed LSTM network.

IV. IMPLEMENTATION & RESPONSE MECHANISM

This section presents the experimental setup of the implementation process of the proposed Deep-IDS. One of the unique features of the proposed IDS is balancing the detection rate and false alarm rate to make it more effective and user-friendly. This has been done by experimenting with the response mechanism, which is also a part of this section.

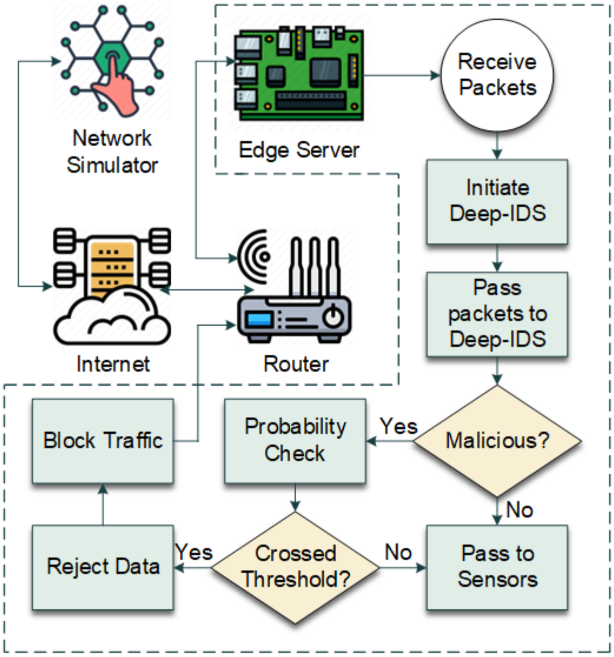


FIGURE 5. Overview of the implementation of the proposed Deep-IDS.

A. EXPERIMENTAL SETUP

The proposed Deep-IDS has been implemented and analyzed in an experimental setup. Experimenting with the proposed Deep-IDS in commercial or organizational networks raises security and ethical concerns. A separate testbed has been created to implement the proposed IDS. Python and the TensorFlow library were used to implement the proposed Deep-IDS. We used the Cooja Network Simulator (CNS) and conducted experiments in the Contiki Operating System (COS). As the IoT, a Raspberry Pi 4 Model B fitted with 4 GB of main memory has been utilized. The experimental setup is illustrated in figure 5 [69].

B. RESPONSE MECHANISM

The Deep-IDS is operated from the edge server. The sensors are capable of receiving data from the Internet using the server. The principle above also holds in data transfer process. The edge server is positioned as an intermediary between the sensors and the router. Upon receiving a packet, the edge server activates the Deep-IDS and then transfers the packets to it. Upon detecting an intrusion, the IDS proceeds to assess the likelihood associated with the specific incursion. If the probability exceeds 70%, the Deep-IDS system will decline the data and transmit a traffic block signal to the router. The router functions to obstruct the origin of the unauthorized access. If no incursion is detected, the Deep-IDS transfers the data to the sensors.

The threshold of the response mechanism has been selected by analyzing the FAR and Detection Rate (DR), illustrated in figure 6. It shows the FAR and the DR at various threshold levels, ranging from 5 to 100. As the threshold increases, we observe a general trend of decreasing FAR and DR. At a threshold of 5, the system demonstrates a high DR of 99.99%, accompanied by a significantly high FAR of 57.58%. As the threshold reaches 50, the DR marginally decreases to 98.41% while the FAR experiences a substantial reduction to 10.02%. Beyond this point, the decrease in FAR becomes less significant while the DR continues to decline more noticeably. At a threshold of 100, the FAR drops to 0%, but the DR also reduces considerably to 85.17%. This analysis highlights a trade-off between the system's detection rate and the false alarm rate as the threshold increases. To optimize system performance, selecting a threshold that balances the need for a high detection rate while minimizing false alarms is crucial. The FAR and DR trade-off is optimized at the 70% threshold.

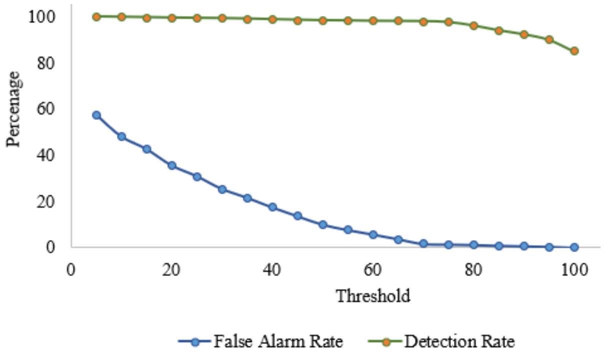


FIGURE 6. Threshold selection.

V. PERFORMANCE EVALUATION

This section presents the performance evaluation of the Deep-IDS developed in this paper. It starts with the evaluation matrices. Then, the confusion matrix was analyzed to evaluate classification performance. After that, the intrusion detection rate and the IDS's response time have been assessed.

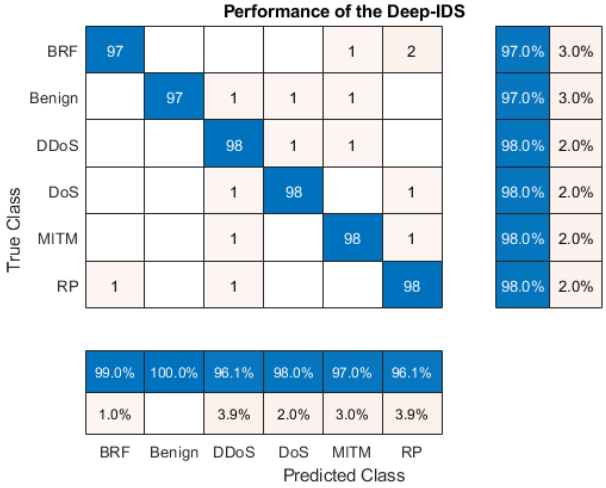
A. EVALUATION METRICS

The state-of-the-art machine learning evaluation metrics have been used in this paper to evaluate the performance of the proposed Deep-IDS [70]. These evaluation metrics are calculated using True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values from the confusion matrix illustrated in figure 7 [71]. The accuracy, precision, recall (sensitivity), and F1 Score are defined by equations 30, 31, 32, and 33, respectively [72]. Another evaluation metric is FAR, which is the False Positive Rate (FPR).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{30}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{31}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{32}$$



True Class \ Predicted Class	BRF	Benign	DDoS	DoS	MITM	RP		
BRF	97				1	2	97.0%	3.0%
Benign		97	1	1	1		97.0%	3.0%
DDoS			98	1	1		98.0%	2.0%
DoS			1	98		1	98.0%	2.0%
MITM			1		98	1	98.0%	2.0%
RP	1		1			98	98.0%	2.0%
	99.0%	100.0%	96.1%	98.0%	97.0%	96.1%		
	1.0%		3.9%	2.0%	3.0%	3.9%		

FIGURE 7. Confusion matrix analysis.

TABLE 2. Performance of the proposed Deep-IDS.

Category	TP	FP	FN	Recall	Precision	F1_Score
Brute Force	97	1	3	0.970	0.990	0.980
Benign	97	0	2	0.970	1.000	0.985
DDoS	98	1	2	0.980	0.990	0.985
DoS	98	3	2	0.980	0.960	0.970
MITM	98	1	2	0.980	0.990	0.985
Replay	98	3	2	0.980	0.960	0.970

$$\text{F1 Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \tag{33}$$

B. CONFUSION MATRIX ANALYSIS

The confusion matrix illustrated in figure 7 demonstrates the performance of the proposed Deep-IDS to classify network traffic into one of the six classes. Upon evaluating the system's performance using Accuracy, Recall, Precision, and F1-Score, it is evident that it performs well in identifying these categories. The system achieves an overall accuracy of 97.67%. The recall values for BRF and Benign instances are 0.970, while DDoS, DoS, MITM, and RP attacks have a higher recall of 0.980.

The precision values indicate the system's effectiveness in identifying true positives. BRF and MITM attacks have a precision of 0.990, while Benign instances have the highest precision of 1.00. DDoS attacks have a precision of 0.990, and DoS and RP attacks have a slightly lower precision of 0.960. The F1-Scores, representing the harmonic mean of recall and precision, are 0.980 for BRF, 0.985 for Benign and DDoS, 0.970 for DoS and RP, and 0.985 for MITM attacks. In summary, the Deep-IDS exhibits high accuracy, recall, precision, and F1 scores, indicating its strong performance in classifying different types of intrusions. The overall performance for each class of the Deep-IDS is listed in table 2.

The exceptional results obtained from the confusion matrix analysis of Deep-IDS underscore its effectiveness in the real-time identification and classification of network intrusions. The high recall and precision values across all categories, particularly for critical intrusion vectors such as DDoS, MITM, and BRF attacks, affirm the system's robustness and reliability in distinguishing between benign and malicious traffic with minimal error. The system's ability to achieve a perfect precision score of 1.00 for benign instances highlights its capability to accurately identify legitimate network activities, thus reducing the likelihood of false positives that could disrupt normal network operations. Moreover, the high F1 scores indicate a balanced performance between recall and precision, ensuring that the system is not only accurate but also consistent in its intrusion detection capabilities. This balance is crucial for maintaining network security and integrity, especially in dynamic and complex IoT environments where the cost of false negatives or positives can be substantial. These metrics provide compelling evidence of its potential to significantly enhance cybersecurity defenses against an evolving landscape of cyber threats.

TABLE 3. Intrusion detection rate and response time.

Intrusion	Intrusion	Detected	Detection Rate	Response Time (s)
Brute Force	50	49	98	1.46
DDoS	50	48	96	2.02
DoS	50	49	98	1.98
MITM	50	48	96	1.05
Replay	50	48	96	0.92

C. DETECTION RATE AND RESPONSE TIME ANALYSIS

The proposed Deep-IDS achieves an overall accuracy of 97.67%. The detection rate plays a significant role in evaluating its performance. The detection rate, defined by equation 34, determines the robustness of the proposed Deep-IDS.

$$\text{Detection Rate, R} = \frac{\text{Detected Intrusion}}{\text{Total Intrusion}} \times 100 \tag{34}$$

An experiment with known intrusions and random network data has been conducted to identify the detection rate. In this experiment, there are a total of 250 intrusions and 1000 benign data sequences. The 250 intrusions were randomly injected with benign data. There is no specific pattern of benign and malicious sequence combinations. It is a 180-minute lengthy experiment. Along with the detection rate, the response time was also measured. The findings of this experiment are listed in table 3.

The data in table 3 provide insights into the performance of the proposed Deep-IDS when dealing with different types of intrusions, specifically BRF, DDoS, DoS, MITM, and RP attacks. The detection rates for BRF and DoS attacks are the highest at 98%, with 49 out of 50 intrusions detected in both cases. Meanwhile, the system performs slightly less effectively for DDoS, MITM, and RP attacks, achieving a detection rate of 96% by detecting 48 out of 50 intrusions. The response time varies across different intrusion types, with RP attacks having the fastest response

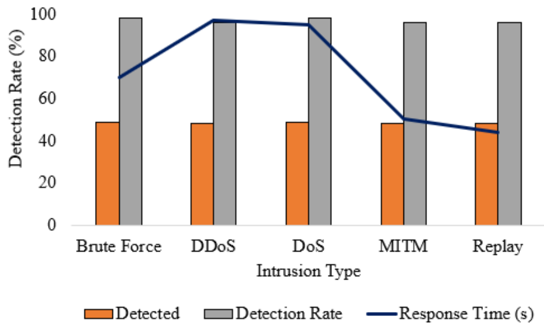


FIGURE 8. The detection rate and response time.

time of 0.92 seconds, followed closely by MITM attacks at 1.05 seconds. BRF intrusions have a slightly longer response time of 1.46 seconds, while DoS and DDoS attacks require even more time, with response times of 1.98 and 2.02 seconds, respectively. The detection rate and the response time have been illustrated in figure 8.

Figure 8 demonstrates the IDS's impressive performance across various attack types. The response time is fast enough to consider it real-time intrusion detection.

D. PERFORMANCE COMPARISON

The Table 4 presents the performance comparison between the proposed Deep-IDS and other similar approaches. The proposed Deep-IDS demonstrates superior performance in intrusion detection compared to several existing approaches, as evidenced by a comprehensive performance comparison. Deep-IDS outperforms other notable systems with an accuracy of 97.67%, precision of 97.67%, a recall of 98.17%, and an F1-Score of 97.91%. For instance, Ashiku et al. [17] reported an accuracy of 94.40% but did not provide figures for precision, recall, or F1-Score, indicating a narrower focus on accuracy alone. Similarly, Musleh et al. [18] achieved a commendable balance with 92.40% accuracy, 89.10% precision, and a 92% F1-Score, yet fell short of the comprehensive performance metrics offered by Deep-IDS. Notably, Logeswari et al. [19], and Mebawondu et al. [20] presented systems with significantly lower overall performance metrics, highlighting the advanced capabilities of Deep-IDS in handling various intrusion types effectively. Elsayed et al. [21] and Kumar et al. [23] also proposed systems with high accuracy rates of 96.56% and 97.45%, respectively, but neither matched the balanced performance across all metrics achieved by Deep-IDS. This comparison underscores the robustness and efficiency of Deep-IDS in accurately detecting and classifying intrusions, setting a new benchmark in the field of cybersecurity for IoT systems.

VI. LIMITATION & FUTURE SCOPE

Every system has some limitations. The proposed Deep-IDS is no exception. It suffers from several limitations, which have been presented in this section.

TABLE 4. The performance comparison of the proposed system with other similar systems (here NA means Not Available).

Author	Accuracy	Precision	Recall	F1-Score
Ashiku et al. [17]	94.40%	NA	NA	NA
Musleh et al. [18]	92.40%	89.10%	94.20%	92%
Logeswari et al. [19]	82.20%	82.84%	82.63%	82.84%
Mebawondu et al. [20]	76.96%	79.80%	NA	NA
Abdelkhalek et al. [21]	83.50%	74%	94%	82%
Elsayed et al. [22]	96.56%	97.30%	NA	97.35%
Kumar et al. [23]	97.45%	96.95%	NA	94.43%
Proposed	**97.67%**	**97.67%**	**98.17%**	**97.91%**

A. COST

The proposed Deep-IDS uses a Raspberry Pi 4 Model B with 4GB primary memory. It is a headless computer and an expensive device [73]. It has been designed to perform many other sophisticated computations. That is why it is an expensive device. In this experiment, it was used only for intrusion detection. An embedded system exclusively designed for the Deep-IDS would reduce the implementation cost. However, developing an embedded system is beyond this research's scope. It creates a new opportunity to conduct further research to make IDS hardware cost-effective.

B. ADVERSARIAL MACHINE LEARNING (AML) ATTACK

The proposed Deep-IDS demonstrates outstanding performance as an intrusion detector. However, no counter-measurement has been taken for the AML attack. The CIC-IDS2017 is a public dataset. Anyone can access and analyze the data to prepare sequences for an AML attack [74]. Even though AML attacks have been drawing significant attention lately, they are not within the proposed Deep-IDS context. However, it opens the door to conducting more experiments to defend against AML attacks and secure the proposed Deep-IDS.

C. REAL-WORLD EXPERIMENT

The proposed Deep-IDS has been experimented with in a testbed that resembles a real-world scenario. As a result, the performance of the proposed system is considered a realistic result. However, a testbed does not encompass a large perimeter like a realistic environment [75]. It is a significant limitation of the proposed system.

D. CYBER-PHYSICAL SYSTEM SECURITY

This paper focuses on intrusion detection only. However, a system is always vulnerable to cyber-physical intrusion [11] unless necessary counter-measurement is taken. Anyone with access to the edge server has the scope to make the entire system vulnerable by gaining unauthorized access or altering the IDS's parameters.

The proposed Deep-IDS's existing limitations are the future scopes of further research. Eventually, more weaknesses of this system will be discovered, and in subsequent research, those weaknesses will be strengthened. This is how, through continuous improvement, the proposed Deep-IDS will be an efficient, effective, and unique IDS.

VII. CONCLUSION

The Deep-IDS presented in this paper is an innovative solution to detect five types of intrusion in real-time with an average of 97.67%. The methodology presented in this paper has been developed to abridge the research gaps identified through a comprehensive literature review. Unlike most of the research conducted in this field, this paper justifies the Deep Learning algorithms used in this paper through a rigorous characteristics analysis. This paper has presented the design and implementation of an effective LSTM network that detects five types of intrusion with an average detection rate of 96.8%. The average detection time is only 1.49 seconds. The well-engineered network architecture incorporated with the He weight initialization method has made it possible. It has been further optimized through the ADAM optimization algorithm. Furthermore, The appropriate sequence generation method and other dataset processing techniques have contributed to the outstanding performance of the IDS developed in this paper. That is why it detects intrusion in real-time with 97.67% precision, 98.17% recall, and 97.91% F1-Score. The innovative trade-off analysis between the detection rate and false alarm rate has introduced a uniqueness to this paper. An innovative response mechanism has been developed and presented in this paper, which is the roadmap to apply the detection made by the LSTM network. It prevents the compromised IoT node from receiving any further malicious traffic. As a result, it can no longer be used as an access point by cybercriminals. It makes a significant impact in strengthening the IoT node security as well as the security of the whole interconnected network.

REFERENCES

[1] S. Fraihat, S. Makhadmeh, M. Awad, M. A. Al-Betar, and A. Al-Redhaei, "Intrusion detection system for large-scale IoT NetFlow networks using machine learning with modified arithmetic optimization algorithm," *Internet Things*, vol. 22, Jul. 2023, Art. no. 100819.

[2] A. Behura, A. Singh, and S. Nayak, "Integration of cloud, iot, and ai for smart services," in *Fostering Cross-Industry Sustainability With Intelligent Technologies*. Hershey, PA, USA: IGI Global, 2024, pp. 162–182.

[3] M. K. Kagita, N. Thilakarathne, T. R. Gadekallu, P. K. R. Maddikunta, and S. Singh, "A review on cyber crimes on the Internet of Things," in *Deep Learning for Security and Privacy Preservation in IoT*. Singapore: Springer, 2022, pp. 83–98.

[4] F. Nie, W. Liu, G. Liu, and B. Gao, "M2VT-IDS: A multi-task multi-view learning architecture for designing IoT intrusion detection system," *Internet Things*, vol. 25, Apr. 2024, Art. no. 101102.

[5] S. Sharma, V. Kumar, and K. Dutta, "Multi-objective optimization algorithms for intrusion detection in IoT networks: A systematic review," *Internet Things Cyber-Phys. Syst.*, vol. 4, pp. 258–267, Jan. 2024.

[6] S. Wang, W. Xu, and Y. Liu, "Res-TranBiLSTM: An intelligent approach for intrusion detection in the Internet of Things," *Comput. Netw.*, vol. 235, Nov. 2023, Art. no. 109982.

[7] C. Ni and S. C. Li, "Machine learning enabled industrial IoT security: Challenges, trends and solutions," *J. Ind. Inf. Integr.*, vol. 38, Mar. 2024, Art. no. 100549.

[8] M. Nouman, U. Qasim, H. Nasir, A. Almasoud, M. Imran, and N. Javaid, "Malicious node detection using machine learning and distributed data storage using blockchain in WSNs," *IEEE Access*, vol. 11, pp. 6106–6121, 2023.

[9] E. K. Boahen, B. E. Bouya-Moko, F. Qamar, and C. Wang, "A deep learning approach to online social network account compromisation," *IEEE Trans. Computat. Social Syst.*, vol. 10, no. 6, pp. 3204–3216, Dec. 2023.

[10] W. I. Khedr, A. E. Gouda, and E. R. Mohamed, "FMDADM: A multi-layer DDoS attack detection and mitigation framework using machine learning for stateful SDN-based IoT networks," *IEEE Access*, vol. 11, pp. 28934–28954, 2023.

[11] S. Achar, N. Faruqui, M. Whaiduzzaman, A. Awajan, and M. Alazab, "Cyber-physical system security based on human activity recognition through IoT cloud computing," *Electronics*, vol. 12, no. 8, p. 1892, Apr. 2023.

[12] C. Chethana, P. K. Pareek, V. H. C. de Albuquerque, A. Khanna, and D. Gupta, "Deep learning technique based intrusion detection in cyber-security networks," in *Proc. IEEE 2nd Mysore Sub Sect. Int. Conf. (MysuruCon)*, Oct. 2022, pp. 1–7.

[13] S. W. Azumah, N. Elsayed, V. Adewopo, Z. S. Zaghloul, and C. Li, "A deep LSTM based approach for intrusion detection IoT devices network in smart home," in *Proc. IEEE 7th World Forum Internet Things (WF-IoT)*, Jun. 2021, pp. 836–841.

[14] M. Ahsan, N. Rifat, M. Chowdhury, and R. Gomes, "Intrusion detection for IoT network security with deep neural network," in *Proc. IEEE Int. Conf. Electro Inf. Technol. (eIT)*, May 2022, pp. 467–472.

[15] N. Yadav, S. Pande, A. Khamparia, and D. Gupta, "Intrusion detection system on IoT with 5G network using deep learning," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–13, Mar. 2022.

[16] A. M. Banaamah and I. Ahmad, "Intrusion detection in IoT using deep learning," *Sensors*, vol. 22, no. 21, p. 8417, Nov. 2022.

[17] L. Ashiku and C. Dagli, "Network intrusion detection system using deep learning," *Proc. Comput. Sci.*, vol. 185, pp. 239–247, Jan. 2021.

[18] D. Musleh, M. Alotaibi, F. Alhaidari, A. Rahman, and R. M. Mohammad, "Intrusion detection system using feature extraction with machine learning algorithms in IoT," *J. Sensor Actuator Netw.*, vol. 12, no. 2, p. 29, Mar. 2023.

[19] G. Logeswari, S. Bose, and T. Anitha, "An intrusion detection system for SDN using machine learning," *Intell. Autom. Soft Comput.*, vol. 35, no. 1, pp. 867–880, 2023.

[20] J. O. Mebawondu, O. D. Alowolodu, J. O. Mebawondu, and A. O. Adetunmbi, "Network intrusion detection system using supervised learning paradigm," *Sci. Afr.*, vol. 9, Sep. 2020, Art. no. e00497.

[21] A. Abdelkhalek and M. Mashaly, "Addressing the class imbalance problem in network intrusion detection systems using data resampling and deep learning," *J. Supercomput.*, vol. 79, no. 10, pp. 10611–10644, Jul. 2023.

[22] R. A. Elsayed, R. A. Hamada, M. I. Abdalla, and S. A. Elsaid, "Securing IoT and SDN systems using deep-learning based automatic intrusion detection," *Ain Shams Eng. J.*, vol. 14, no. 10, Oct. 2023, Art. no. 102211.

[23] P. Kumar, G. P. Gupta, and R. Tripathi, "TP2SF: A trustworthy privacy-preserving secured framework for sustainable smart cities by leveraging blockchain and machine learning," *J. Syst. Archit.*, vol. 115, May 2021, Art. no. 101954.

[24] M. Saied, S. Guirguis, and M. Madbouly, "Review of artificial intelligence for enhancing intrusion detection in the Internet of Things," *Eng. Appl. Artif. Intell.*, vol. 127, Jan. 2024, Art. no. 107231.

[25] H. C. Altunay and Z. Albayrak, "A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks," *Eng. Sci. Technol., Int. J.*, vol. 38, Feb. 2023, Art. no. 101322.

[26] S. A. Bakhsh, M. A. Khan, F. Ahmed, M. S. Alshehri, H. Ali, and J. Ahmad, "Enhancing IoT network security through deep learning-powered intrusion detection system," *Internet Things*, vol. 24, Dec. 2023, Art. no. 100936.

[27] R. Chaganti, W. Suliman, V. Ravi, and A. Dua, "Deep learning approach for SDN-enabled intrusion detection system in IoT networks," *Information*, vol. 14, no. 1, p. 41, 2023.

[28] O. H. Abdulganiyu, T. Ait Tchakoucht, and Y. K. Saheed, "A systematic literature review for network intrusion detection system (IDS)," *Int. J. Inf. Secur.*, vol. 22, no. 5, pp. 1125–1162, Oct. 2023.

[29] B. Jothi and M. Pushpalatha, "WILS-TRS—A novel optimized deep learning based intrusion detection framework for IoT networks," *Pers. Ubiquitous Comput.*, vol. 27, no. 3, pp. 1285–1301, Jun. 2023.

[30] P. Sanju, "Enhancing intrusion detection in IoT systems: A hybrid metaheuristics-deep learning approach with ensemble of recurrent neural networks," *J. Eng. Res.*, vol. 11, no. 4, pp. 356–361, Dec. 2023.

[31] A. A. E. Donkol, A. G. Hafez, A. I. Hussein, and M. M. Mabrook, "Optimization of intrusion detection using likely point PSO and enhanced LSTM-RNN hybrid technique in communication networks," *IEEE Access*, vol. 11, pp. 9469–9482, 2023.

[32] N. Patel, S. Trivedi, and N. Faruqui, "An innovative deep neural network for stress classification in workplace," in *Proc. Int. Conf. Smart Comput. Appl. (ICSCA)*, Feb. 2023, pp. 1–5.

[33] M. A. Alsoufi, S. Razak, M. M. Siraj, I. Nafea, F. A. Ghaleb, F. Saeed, and M. Nasser, "Anomaly-based intrusion detection systems in IoT using deep learning: A systematic literature review," *Appl. Sci.*, vol. 11, no. 18, p. 8383, 2021.

[34] N. Faruqui, M. A. Yousuf, M. Whaiduzzaman, A. K. M. Azad, A. Barros, and M. A. Moni, "LungNet: A hybrid deep-CNN model for lung cancer diagnosis using CT and wearable sensor-based medical IoT data," *Comput. Biol. Med.*, vol. 139, Dec. 2021, Art. no. 104961.

[35] D. Mandic and J. Chambers, *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability*. Hoboken, NJ, USA: Wiley, 2001.

[36] S. Trivedi, N. Patel, and N. Faruqui, "Bacterial strain classification using convolutional neural network for automatic bacterial disease diagnosis," in *Proc. 13th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence)*, Jan. 2023, pp. 325–332.

[37] N. Faruqui, M. A. Yousuf, M. Whaiduzzaman, A. Azad, S. A. Alyami, P. Lió, M. A. Kabir, and M. A. Moni, "SafetyMed: A novel IoMT intrusion detection system using CNN-LSTM hybridization," *Electronics*, vol. 12, no. 17, p. 3541, Aug. 2023.

[38] S. Trivedi, N. Patel, and N. Faruqui, "A novel lightweight lung cancer classifier through hybridization of DNN and comparative feature optimizer," in *Proc. Int. Conf. Hybrid Intell. Syst.* Cham, Switzerland: Springer, 2022, pp. 188–197.

[39] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of CNN and RNN for natural language processing," 2017, *arXiv:1702.01923*.

[40] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," 2014, *arXiv:1406.1078*.

[41] N. K. Manaswi and N. K. Manaswi, "RNN and LSTM," in *Deep Learning With Applications Using Python: Chatbots and Face, Object, and Speech Recognition With TensorFlow and Keras*, 2018, pp. 115–126.

[42] A. Awajan, "A novel deep learning-based intrusion detection system for IoT networks," *Computers*, vol. 12, no. 2, p. 34, Feb. 2023.

[43] M. Alazab, A. Awajan, H. Alazzam, M. Wedyan, B. Alshawi, and R. Alturki, "A novel IDS with a dynamic access control algorithm to detect and defend intrusion at IoT nodes," *Sensors*, vol. 24, no. 7, p. 2188, Mar. 2024.

[44] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2017, pp. 1643–1647.

[45] H. S. Gill and B. S. Khehra, "An integrated approach using CNN-RNN-LSTM for classification of fruit images," *Mater. Today, Proc.*, vol. 51, pp. 591–595, Jan. 2022.

[46] Z. Tan and M. Karakose, "Comparative study for deep reinforcement learning with CNN, RNN, and LSTM in autonomous navigation," in *Proc. Int. Conf. Data Analytics Bus. Industry, Way Towards Sustain. Economy (ICDABI)*, Oct. 2020, pp. 1–5.

[47] S. Aktar and A. Yasin Nur, "Towards DDoS attack detection using deep learning approach," *Comput. Secur.*, vol. 129, Jun. 2023, Art. no. 103251.

[48] S. Kouadri, C. B. Pande, B. Panneerselvam, K. N. Moharir, and A. Elbeltagi, "Prediction of irrigation groundwater quality parameters using ANN, LSTM, and MLR models," *Environ. Sci. Pollut. Res.*, vol. 29, no. 14, pp. 21067–21091, Mar. 2022.

[49] E. Ahmadzadeh, H. Kim, O. Jeong, N. Kim, and I. Moon, "A deep bidirectional LSTM-GRU network model for automated ciphertext classification," *IEEE Access*, vol. 10, pp. 3228–3237, 2022.

[50] N. Faruqui, M. A. Yousuf, P. Chakraborty, and M. S. Hossain, "Innovative automation algorithm in micro-multinational data-entry industry," in *Cyber Security and Computer Science*, Dhaka, Bangladesh. Cham, Switzerland: Springer, 2020, pp. 680–692.

[51] L. Yu, R. Zhou, R. Chen, and K. K. Lai, "Missing data preprocessing in credit classification: One-hot encoding or imputation?" *Emerg. Markets Finance Trade*, vol. 58, no. 2, pp. 472–482, Jan. 2022.

[52] A. Thakkar and R. Lohiya, "Attack classification using feature selection techniques: A comparative study," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 1, pp. 1249–1266, Jan. 2021.

[53] V. Benedetti, H. Casini, and P. J. Martinez, "Mutual information of generalized free fields," *Phys. Rev. D, Part. Fields*, vol. 107, no. 4, Feb. 2023, Art. no. 046003.

[54] Y. Zhu, Z. Li, F. Wang, and L. Li, "Control sequences generation for testing vehicle extreme operating conditions based on latent feature space sampling," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 4, pp. 2712–2722, Apr. 2023.

[55] N. Faruqui, M. A. Yousuf, F. A. Kateb, M. A. Hamid, and M. M. Monowar, "Healthcare as a service (HAAS): CNN-based cloud computing model for ubiquitous access to lung cancer diagnosis," *Heliyon*, vol. 9, no. 11, Nov. 2023, Art. no. e21520.

[56] K. Prateek, S. Maity, and R. Amin, "An unconditionally secured privacy-preserving authentication scheme for smart metering infrastructure in smart grid," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 2, pp. 1085–1095, Mar. 2023.

[57] A. Gerodimos, L. Maglaras, M. A. Ferrag, N. Ayres, and I. Kantzavelou, "IoT: Communication protocols and security threats," *Internet Things Cyber-Phys. Syst.*, vol. 3, pp. 1–13, Jan. 2023.

[58] S. Sharma, C. Ganguly, and S. De, "Effect of polarization on RF signal transmission over two-ray channel," in *Proc. Nat. Conf. Commun. (NCC)*, Feb. 2023, pp. 1–6.

[59] K. Prateek, N. K. Ojha, F. Altaf, and S. Maity, "Quantum secured 6G technology-based applications in Internet of Everything," *Telecommun. Syst.*, vol. 82, no. 2, pp. 315–344, Feb. 2023.

[60] W. Lee, S. J. Stolfo, P. K. Chan, E. Eskin, W. Fan, M. Miller, S. Hershkop, and J. Zhang, "Real time data mining-based intrusion detection," in *Proc. Inf. Survivability Conf. Expo.*, 2001, vol. 1, pp. 89–100.

[61] Y. Cai, W. Hua, H. Chen, G. Edward Suh, C. De Sa, and Z. Zhang, "Structured pruning is all you need for pruning CNNs at initialization," 2022, *arXiv:2203.02549*.

[62] K. D. Humbird, J. L. Peterson, and R. G. Mcclarren, "Deep neural network initialization with decision trees," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1286–1295, May 2019.

[63] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2015, pp. 1026–1034.

[64] A. Fred Agarap, "Deep learning using rectified linear units (ReLU)," 2018, *arXiv:1803.08375*.

[65] N. R. Goodman, "Statistical analysis based on a certain multivariate complex Gaussian distribution (an introduction)," *Ann. Math. Statist.*, vol. 34, no. 1, pp. 152–177, Mar. 1963.

[66] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[67] A. Lydia and S. Francis, "Adagrad—An optimizer for stochastic gradient descent," *Int. J. Inf. Comput. Sci.*, vol. 6, no. 5, pp. 566–568, 2019.

[68] F. Zou, L. Shen, Z. Jie, W. Zhang, and W. Liu, "A sufficient condition for convergences of Adam and RMSProp," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11119–11127.

[69] N. Faruqui, M. A. Kabir, M. A. Yousuf, M. Whaiduzzaman, A. Barros, and I. Mahmud, "Trackez: An IoT-based 3D-object tracking from 2D pixel matrix using MEZ and FSL algorithm," *IEEE Access*, vol. 11, pp. 61453–61467, 2023.

[70] S. Bera, T. Dey, D. Das Adhikary, S. Guchhhait, U. Nandi, N. Faruqui, and B. Paul, "Identification of mental state through speech using a deep learning approach," in *Proc. Doctoral Symp. Hum. Centered Comput.* Singapore: Springer, 2023, pp. 43–53.

[71] S. Trivedi, N. Patel, and N. Faruqui, "NDNN based U-Net: An innovative 3D brain tumor segmentation method," in *Proc. IEEE 13th Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, Oct. 2022, pp. 0538–0546.

[72] L. P. O. Paula, N. Faruqui, I. Mahmud, M. Whaiduzzaman, E. C. Hawkinson, and S. Trivedi, "A novel front door security (FDS) algorithm using GoogleNet-BiLSTM hybridization," *IEEE Access*, vol. 11, pp. 19122–19134, 2023.

[73] P. Chakraborty, M. A. Yousuf, M. Z. Rahman, and N. Faruqui, "How can a robot calculate the level of visual focus of human's attention," in *Proc. Int. Joint Conf. Comput. Intell. (IJCCI)*. Singapore: Springer, 2020, pp. 329–342.

[74] S. Trivedi, T. Anh Tran, N. Faruqui, and M. M. Hassan, "An exploratory analysis of effect of adversarial machine learning attack on IoT-enabled industrial control systems," in *Proc. Int. Conf. Smart Comput. Appl. (ICSCA)*, Feb. 2023, pp. 1–8.

[75] M. E. Hossain, N. Faruqui, I. Mahmud, T. Jan, M. Whaiduzzaman, and A. Barros, "DPMS: Data-driven promotional management system of universities using deep learning on social media," *Appl. Sci.*, vol. 13, no. 22, p. 12300, Nov. 2023.

SANDEEPKUMAR RACHERLA (Senior Member, IEEE) received the bachelor's degree in computer science and the master's degree in economics from the Birla Institute of Technology and Science, Pilani, Goa, India, and the master's degree in business analytics from The University of Texas at Dallas, Richardson, TX, USA.

He is currently a Data Scientist with Amazon, USA. Previously, he was a Senior Software Engineer at Qualcomm Inc. His research interests include data science and machine learning, wherein he has successfully created, developed, and executed innovative AI and ML solutions in domains, such as people and workforce analytics, pricing, and promotions, saving companies millions of dollars. Apart from his passion for discussing data science, he also enjoys mentoring.

PRATHYUSHA SRIPATHI (Member, IEEE) received the bachelor's degree in computer science from Jawaharlal Nehru Technological University (JNTU), India, the M.B.A. degree in marketing from Indian Institute of Management (IIM), and the master's degree in computational data analytics from Georgia Institute of Technology (Georgia Tech), USA.

She is currently a Data Scientist with Amazon, USA. Previously, she held the position of an Analytics Consultant at Latent View Analytics and a Software Engineer with ACS Solutions. Her research interests include data science and machine learning.

NURUZZAMAN FARUQUI received the B.Sc. degree (Hons.) in electrical and electronics engineering (EEE) from North South University, Bangladesh, in 2016, and the master's degree in information technology from the Institute of Information Technology (IIT), Jahangirnagar University (JU), Bangladesh, in 2018.

He is currently a Senior Lecturer with the Department of Software Engineering (SWE), Daffodil International University, Bangladesh. He is also a YouTuber and the author. He is globally recognized for his educational video content on neural networks using MATLAB. He is also a Research Coordinator with the Department of SWE. He has authored three books so far. His research interests include artificial intelligence, machine learning, deep learning, cloud computing, and image processing.

Mr. Faruqui is a member of The Institution of Engineers, Bangladesh (IEB) and Bangladesh Society for Private University Academics (BSPUA).

MD ALAMGIR KABIR (Member, IEEE) received the master's degree in software engineering from Wuhan University, China, in 2017, and the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, in 2021. He is currently a Postdoctoral Researcher with the Artificial Intelligence and Intelligent Systems Research Group, Mälardalen University, Sweden. He is an inquisitive individual, driven to develop novel techniques that facilitate efficient, explainable, and robust AI. During his graduate studies, he conducted research on various topics in software engineering, including software testing, verification and validation, and software metrics. He has published several research papers in top-tier conferences and journals in his field. At Mälardalen University, his research interests include developing new techniques for artificial intelligence and intelligent systems.

MD WHAIDUZZAMAN (Senior Member, IEEE) received the bachelor's degree in electronics and computer science and the M.Sc. degree in telecommunication and computer network engineering from London, U.K., and the Ph.D. degree from the University of Malaya, Malaysia.

He is currently a Professor with the Institute of Information Technology (IIT), Jahangirnagar University. He also works on ARC-funded projects at the Queensland University of Technology, Australia. His research interests include mobile cloud computing, vehicular cloud computing, fog computing, the IoT, and microservices. He received the *Journal of Network and Computer Applications* (Elsevier) Best Paper Award, Paris, France.

SYED AZIZ SHAH was appointed as an Associate Professor with the Research Centre for Intelligent Healthcare (CIH), Coventry University (CU), in 2020. He is an Interdisciplinary Researcher focusing on advanced radio frequency (RF) sensor design and signal processing using RF and THz sensing, specifically for healthcare applications. He is highly motivated to utilize and develop advanced technologies that address the unmet healthcare challenges of unobtrusive monitoring of older adults' daily activities for health and well-being purposes.

He was a recipient of the U.K. Prestigious Engineering and Physical Sciences Research Council (EPSRC) New Investigator Award. He also received the endorsement of U.K. exceptional talent candidate (Emerging Leader), in 2018, for his pioneering work in the field of wireless sensing for remote patient monitoring. These prestigious awards are given to early-career, world-leading innovators and scientists from the Royal Academy of Engineering.