IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 5 April 2024, accepted 22 May 2024, date of publication 27 May 2024, date of current version 4 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3405628

RESEARCH ARTICLE

DL-ADS: Improved Grey Wolf Optimization Enabled AE-LSTM Technique for Efficient Network Anomaly Detection in Internet of Thing Edge Computing

J. MANOKARAN[1] AND G. VAIRAVEL[2], (Senior Member, IEEE)
[1]Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, Kattankulathur, Chennai, Tamil Nadu 603203, India
[2]Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, Tiruchirappalli, Tamil Nadu 621105, India

Corresponding authors: G. Vairavel (vairavelg@protonmail.com) and J. Manokaran (manoraj3@gmail.com)

ABSTRACT The Internet of Things (IoT) technology has begun to proliferate in recent years, which simultaneously increases the number of attacks. Owing to the massive volume and multi-dimensional data in IoT, anomaly detection leads to low prediction accuracy and a high false alarm rate. Further, there is a deficit of real-world test datasets for anomaly detection. This work aims to generate a novel real-time anomaly detection dataset and proposes an efficient anomaly detection model using an Improved Grey Wolf Optimization (IGWO)-enabled Long Short-Term Memory (LSTM) network in IoT edge scenarios. Dataset generation is carried out using a testbed setup containing Raspberry Pi 4 and sensors connected by a lightweight Message Queuing Telemetry Transport (MQTT) protocol. An autoencoder is used for feature reduction as it can investigate the input characteristics without sacrificing vital information. The LSTM classifier parameters, such as learning rate, optimizer, and batch size, are tuned precisely using IGWO techniques. The experimental results disclose that the proposed model achieves an accuracy of 99.11% for the testbed dataset, which is better than recent models. To confirm the generalizability of our model, the CICIDS 2017, DS2OS, and MQTTset standard datasets are applied explicitly. The developed model outcomes are statistically verified using the Wilcoxon signed rank test.

INDEX TERMS Anomaly detection, autoencoder, improved grey wolf optimization, Internet of Things security, LSTM networks, MQTT, Wilcoxon signed rank test.

I. INTRODUCTION

A. BACKGROUND

The Internet of Things is an emerging paradigm that can create a better connection between machines and humans and machine-to-machine using various sensing and actuating devices [1]. Recently, the growing prevalence of IoT technology has made our lives comfortable and more productive in many domains, such as smart agriculture, Industry 4.0, smart homes, connected cars, smart cities, and e-healthcare [2]. Owing to the expansion of notable technologies, such as 5G communication, blockchain, explainable artificial intelligence (XAI), and edge computing, IoT technology has been drastically increasing. According to the International Data Corporation (IDC) report, approximately 42 billion intelligent gadgets will be used by 2025, generating 79.4 ZB of data and \$3 trillion in revenue [3]. The sum of connected IoT gadgets from 2020-2025 is shown in Figure 1. (a) [4].

As IoT grows, it also attracts potential threats from different intruders, causing IoT security and privacy difficulties. Unlike traditional network security, IoT security faces many challenges, as it has multi-dimensional data, massive complex data, and resource-constrained IoT devices. A Bitdefender report states that smart home networks will experience eight attacks every 24 hours by 2024. Figure 1 (b)

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

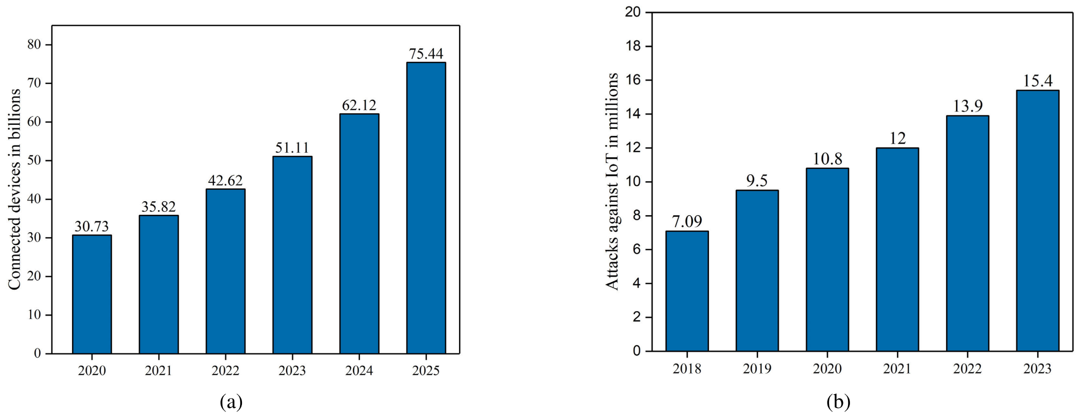


FIGURE 1. (a) Billions of connected smart gadgets in the world (2020-2025) [4] (b) Number of attacks registered against IoT devices in 2018-2023 [5].

depicts the number of attacks registered against IoT devices in 2018-2023 [5].

An intrusion detection system (IDS) is crucial to improve the reliability of IoT networks. The primary objective of an IDS is to recognize and respond to security breaches. The IDS can be grouped into signature-based and anomaly-based methods using detection methodology. In the signature-based approach, accurate patterns are built from known attacks and identify attacks with high accuracy. However, signature-based IDS cannot identify recent zero-day attacks. Finding patterns and distinguishing them from usual traffic patterns are the fundamental concepts of anomaly-based detection. Unlike signature-based systems, anomaly-based systems are more accurate against zero-day and metamorphic attacks, making them most suitable for IoT networks. However, the high false positive rate is an ongoing concern [6], [7].

Researchers have used various machine learning (ML) and DL algorithms in recent years to increase the performance of an IDS. Decision Tree (DT) [8], Self-Organizing Map (SOM) [9], K-Nearest Neighbors (KNN) [8], Logistic Regression (LR) [8], Support Vector Machine (SVM) [8], and Random Forest (RF) [8] are the most frequently used ML algorithms for anomaly detection. Nonetheless, applying conventional ML algorithms to large, noisy, and complex platforms is challenging since they primarily rely on manually extracted features and lack labeled training datasets. Artificial neural networks (ANN) were used principally to develop the cutting-edge ML paradigm known as the DL algorithm. The DL algorithms have some advantages over ML algorithms due to its various capabilities, including high-level attribute extraction, perfect hidden pattern detection, self-learning, and attribute reduction.

The DL algorithm is a combination of several neural networks (NN), including Convolutional Neural Networks (CNNs) [10], AutoEncoder (AEs) [11], Deep Belief Networks (DBNs) [12], Deep Neural Networks (DNNs) [13], and Recurrent Neural Networks (RNNs) [14], each of which has specific skills and qualities. In addition, a sequence of layers is used to find appropriate high-level features rather than manually selecting them from raw input data. These DL algorithms have been effectively used in various domains, such as image processing, sentiment analysis, speech processing, and health care. In a real-time scenario, most of the extracted network data are immaterial and noisy, which causes the classifier to perform with less forecast accuracy. Thus, dimensionality reduction algorithms, such as Principal Component Analysis (PCA), CNN, and AE, are essential for selecting appropriate data, and here, we have used AE as a feature reduction technique.

Though the DL algorithm-based IDS performs better, its accuracy rate, false alarm rate (FAR), and over-fitting problems can still be improved. Hyper-parameter selection is critical in enhancing accuracy, accelerating learning, and decreasing the FAR for complex and nonlinear network traffic problems. Compared with the standard parameter, the tuned parameter performs better for the DL algorithm. To optimize the DL model's hyper-parameters, we propose evolutionary algorithms as an optimization task [15]. In recent years, grey wolf optimization (GWO) has been developed as a promising swarm intelligence algorithm for deciding many optimization concerns by imitating grey wolf behavior [16]. Further, to enhance the performance, we integrate the Elimination Mechanism (EM) and Opposition-Based Learning (OBL) in the traditional GWO algorithm.

B. MOTIVATION AND CONTRIBUTIONS

Recently, many IDS have used ML algorithms with reasonable predictability. Still, modern fields like Industry 5.0, smart health care, retail industries, and intelligent banking systems produce an enormous amount of data with critical features where ML algorithms are less desirable. Hence, the usage of the DL algorithms is preferred to the ML algorithms. However, in the DL algorithm, the optimum selection of parameters, like rate of learning, choice of optimizer, number of hidden units, and selecting a number

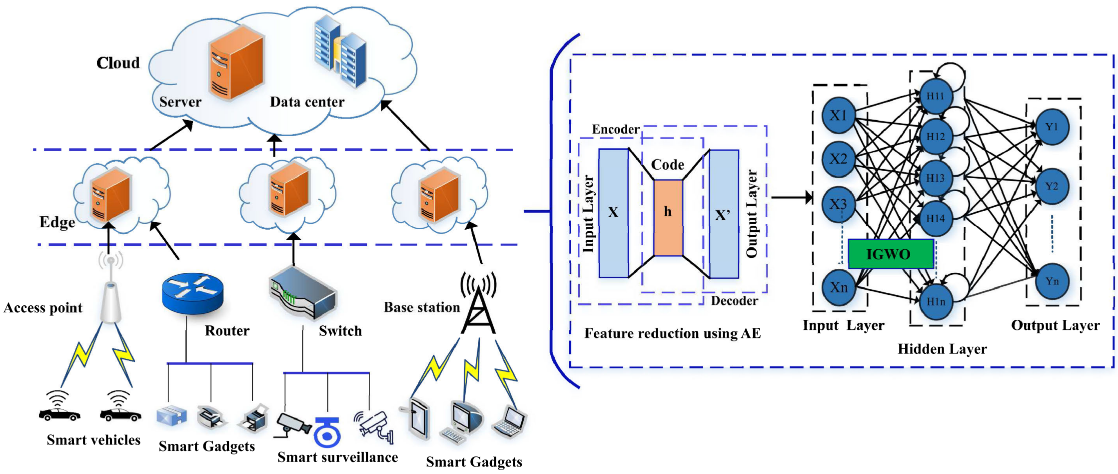


FIGURE 2. The developed DL-ADS employment at IoT Edge.

of layers, is challenging. Furthermore, most existing works use standard datasets and conventional protocols to train the proposed models [17], [18]. NSL KDD [19] and UNSW NB 15 [20] are the most used datasets in recent studies for evaluation. These datasets have limitations, like no IoT telemetry data, no device-level implantation, and a lack of recent IoT attacks. Moreover, conventional protocols may cause considerable latency and use a lot of network resources. We fill these gaps by developing an optimized DL-based anomaly detection model and generating a novel real-time MQTT testbed dataset to train and validate our developed model. The developed model's generalizability is proved by applying it to three different datasets, namely the CICIDS 2017, DS2OS, and MQTTset datasets. Finally, we evaluate the proposed model using parameters like accuracy, FAR, precision, sensitivity, ROC curve, and F1-score metrics. Figure 2 illustrates the construction and placement of the developed model in the IoT edge scenario; here, anomaly detection is performed on the edge device and sent to the cloud, which may further improve the data security and reduce the latency.

This paper aims to create an intelligent IDS using an optimized DL algorithm at IoT edge scenario. Below is a summary of this research's ***key contributions***

(i) After designing an experimental setup and creating many IoT attack scenarios, we generate a novel IDS dataset to train the proposed model.

(ii) Using Autoencoder, the input features are reduced and given as the input of the developed IGWO-optimized LSTM network for anomaly detection.

(iii) The intelligent model is constructed with tuned DL algorithms, and the prediction ability is assessed with a testbed dataset and then associated with the standard dataset (DS2OS, CICIDS 2017, and MQTTset).

(iv) We statistically examine the performance of the developed model with the traditional LSTM network using the Wilcoxon pairwise test.

The remaining paper is organized into five sections. Section II highlights the recent prominent DL-based studies for IDS at IoT. Section III describes the architectural designs, DL techniques, and IGWO techniques used for our developed model. Section IV explains the hardware implementation of our proposed model and the result outcomes. We also perform a comparative analysis of result using the contemporary methods. Lastly, section V concludes the study.

II. RELATED WORKS

Recently, the vast development of cyber-attacks has made the IoT security extremely vulnerable to assault. The employment of DL algorithms and swarm intelligent techniques offer a variety of strategies to identify these assaults, according to the current research. Table 1 discusses some relevant research on IDS using DL algorithms based on different feature selections and various datasets.

In [21], Liu et al. offered a hybrid ADS model that merges the advantages of both ML and DL algorithms. Initially, the clustered RF algorithm rapidly divides the data into normal and attacked data. The attacked data is further categorized into various attack types using CNN and LSTM algorithms. The proposed system achieved 85.24% and 99.91% accuracies for the NSL-KDD and CIC-IDS2017 datasets, respectively. Sahu et al. [22] designed a novel IDS using a CNN + LSTM algorithm for IoT environments. The CNN algorithm is used for attribute extraction, and the LSTM algorithm is used for classification. The testbed dataset generation is a significant contribution by the authors. The authors concluded that the proposed algorithm attained more than 96% detection accuracy.

Huma et al. [23] developed a hybrid DL algorithm for intrusion detection in IoT networks. This algorithm combines DRNN with multilayer perceptron (MLP) algorithm. Authors tested their algorithm using the DS2OS and UNSW NB 15 datasets. Mushtaq et al. [24] offered a two-stage anomaly prediction model with AEs and LSTMs. The AE algorithm

TABLE 1. An overview of DL algorithms for identifying anomalies in IoT networks.

Study	Year	Model	Feature Extraction method	Dataset	Limitations
[21]	2021	K- Means + RF, CNN+LSTM	NIL	NSL KDD, CIC-IDS2017	i) Low accuracy rate ii) Training time mainly depends on the Spark parameters.
[22]	2021	LSTM	CNN	Testbed	Usage of single dataset .
[23]	2021	HDRaNN	NIL	DS2OS, UNSW NB 15	i) No model comparison ii) Features reduction.
[24]	2022	LSTM	AE	NSL KDD	Low classification accuracy, DR, and FAR.
[25]	2022	ChCSO-driven Deep LSTM	CNN	NSL KDD, BoT-IoT	Old dataset, Limited attacks.
[26]	2022	B- Stacking	PCA	NSL KDD, CIC-IDS2017	No parameter tuning, Optimum selection of Principal components is necessary.
[27]	2022	LSTM	CNN	NSL KDD, CIC-IDS2017	No parameter tuning, Optimum selection of Principal components is necessary.
[28]	2022	RNN	CNN	NSLKDD, IoT-NI,BoT-IoT, MQTT, IoT-23, IoT-DS2, MQTTset	No parameter tuning
[29]	2022	WILS-TRS	NIL	Simulated	No feature reduction
[30]	2022	FSO-LSTM-IDS	CNN	Simulated	i) Limited attacks ii) Limited data samples
[31]	2022	Stacking	KPCA+RNN, KPCA+LSTM, KPCA+GRU	Testbed	Optimum selection of Principal components is necessary.
[32]	2023	CNN-LSTM	NIL	UNSW-NB15, X-IIoTID	No feature reduction
[33]	2023	LSTM-AE	NIL	CICIDS2017, CICDIS2018	No feature reduction
[34]	2023	ELSTM-RNN	LPPSO	NSL KDD, UNSW NB15, CICIDS2017, CICDIS2018	No parameter tuning
[35]	2023	LSTM	IBGJO	CICIDS 2017	Increasing time complexity
[36]	2023	LSTM-GRU	PSO-GA	CICIDS 2017	Low convergence speed
[37]	2024	CNN-BiLSTM	TDBO	UNSW NB 15	Not tested on large dataset
[38]	2024	DCGAN-BiLSTM	NIL	BOT-IoT, IoT23, UNSW NB15, ToN-IoT	No feature reduction
[39]	2024	Ensemble (GRU, BiLSTM, SAE)	EPOA	UNSW NB15, SEMCOM	Increasing time complexity

is applied for attribute reduction, and the LSTM algorithm is utilized for classification. The developed system achieved an accuracy of 89% and a FAR of 11%. Deore and Bhosale [25] presented a robust network IDS using an optimization-enabled DL algorithm. The CNN algorithm is used for attribute reduction, and the LSTM algorithm is used for classification. Further, the LSTM algorithm is optimized using the novel Chimp Chicken Swarm Optimization (ChCSO) algorithm to enhance accuracy.

Roy et al. [26] offered a lightweight ADS model for IoT networks using an ensemble learning algorithm. The proposed model comprises three stages: segmentation, feature reduction, and classification. In segmentation, data are clustered into smaller groups. PCA technique is used for dimensionality deduction, which reduces the complexity and increases the speed of the proposed system. Classification is performed using a newly proposed B-stacking algorithm that combines boosting and stacking. Halbouni et al. [27] created a hybrid IDS using the CNN-LSTM algorithm. CNN is applied for attribute extraction, and an LSTM network is employed for classification. For the CICIDS 2017 and UNSW NB 15 datasets, the proposed system achieved 99.64% and 94.53% detection accuracy, respectively.

Ullah and Mahmoud [28] presented an RNN-based IDS for industrial IoT (IIoT) networks, where the CNN is applied for feature extraction and LSTM, BiLSTM, and Gated Recurrent Unit (GRU) are utilized for anomaly prediction. The authors applied seven open-source datasets, compared their results, and concluded that the developed model achieved a higher accuracy and F1 score. Jothi and Pushpalatha [29] and Alqahtani [30] proposed a novel optimization-enabled DL algorithm for IoT IDS. A novel dataset is created using the OMENT++ simulation software for training and testing, which provides a significant contribution to these studies. In [29], Whale-integrated LSTM is used for classification, and in [30], Firefly Swarm Optimization (FSO) enabled

LSTM is used for classification. The authors concluded that the proposed models have demonstrated a superior tradeoff between prediction and response time, making them more suitable for creating an intelligent, scalable IDS for IoT networks.

Ravi et al. [31] proposed an intelligent NIDS using DL algorithms. RNN, LSTM, and GRU algorithms extract the significant features and are reduced by the kernel PCA algorithm. The stacking algorithm is used for classification, SVM and RF algorithms are used as base learners, and LR is used as a Meta- classifier. Altunay and Albayrak [32] suggested an IDS for an IIoT environment using CNN-LSTM algorithms. The authors developed three anomaly detection models: CNN, LSTM, and hybrid CNN+LSTM. An empirical result shows that the hybrid CNN+LSTM performs better than the remaining. Hnamte et al. [33] developed a bi-stage NIDS system using AE-LSTM networks. The model was trained using the CICIDS 2017 and CICIDS 2018 datasets and attained an accuracy of 99.99% and 99.10%. Donkolet al. [34] suggested an enhanced IDS using a Likely point PSO (LPPSO) + hybrid LSTM-RNN techniques. The LPPSO algorithm solves the over-fitting problem and optimizes feature selection. Compared with the existing ML and RNN-based models, the developed model has a higher detection rate and a shorter execution time.

Hanafi et al. [35] offered an IDS system using a novel optimization-enabled DL algorithm for an IoT network. The author used an Improved Binary Golden Jackal Optimization (IBGJO) technique for optimum attribute selection and the LSTM network for classification. The developed system achieved an accuracy of 98.21% for the CICIDS 2017 dataset. Kahtani et al. [36] developed an IDS using hybrid optimization and DL algorithms. The fusion of particle swarm optimization (PSO) and genetic algorithms (GA) is used for optimum attribute selection, and the LSTM-GRU algorithm is utilized for classification. The developed system achieved an accuracy of 98.86% for the CICIDS 2017 dataset. Li et al. [37] enhanced the performance of intrusion detection through an optimized fusion-DL model. Improved Dung Beetle Optimization Algorithm (TDBO) is used for feature selection and parameter optimization and the CNN-BiLSTM network is used for classification. An empirical result shows that the proposed TDBO-CNN-BiLSTM performs better than the remaining. The complexity of the model is relatively high compared to current methods.

Mishra et al. [38] offered an ADS model using a stacked ensemble of DL algorithms. Deep Convolutional GAN (DCGAN) and BiLSTM networks are combined for model training. The standard parameter tuning method is applied for parameter optimization of the DL algorithm. Chander and Upendra Kumar [39] developed an ADS using an optimized ensemble learning algorithm for the IIoT environment. From the complex and high-dimensional IIoT data, the Enhanced Pelican Optimization Algorithm (EPOA) optimally selects the related features. The GRU, BiLSTM, and AE are used as a base learner. The parameters of the DL algorithms are tuned using the seagull optimization method. Using the voting ensemble concept the base learners are combined. An empirical result shows that the proposed method performs better than the standard ADS model.

The above survey identifies several limitations in the existing DL-based IDS in IoT networks. Most studies have used standard datasets, such as NSL KDD, CICIDS 2018, UNSW NB 15, and CICIDS 2017. Still, there is a need for more research in real-time dataset generation and significance test verification of the developed model in IoT networks. Implementing many developed models using a single and standard open-source online dataset makes it impossible to demonstrate the generalization and detection of recent attacks. Further, most current studies use a bi-classification method that cannot observe the different behaviors of anomalies. Many IDS models do not identify the parameter optimization of the DL algorithm, resulting in reduced detection accuracy, increased training time, and decreased effectiveness for complex IoT scenarios. Moreover, conventional protocols (TCP, UDP, and HTTP) may cause considerable latency and utilize more network resources.

This study shows significant contribution, distinguishing it from previous studies. i) Using a testbed setup containing Raspberry Pi 4, personal computers (PCs), and sensors interconnected using an MQTT protocol, we generate a novel real-time testbed dataset to evaluate our proposed model. ii) A multi-classification technique helps to identify network anomalous behaviors more accurately. iii) Using an autoencoder, the critical features are reduced and the anomalies are identified effectively by the proposed IGWO-tuned LSTM network.

III. PROPOSED ANOMALY DETECTION MODEL

In this section, we explain the DL-based anomaly detection model, which consists of several self-ruling stages. Figure 3 shows the architectural design flow of a network IDS. A real-time experimental testbed setup is used to initially generate and preprocess a novel dataset. The RN-SMOTE algorithm is utilized to overcome the imbalance issue in the anomaly dataset. The input features are reduced using the autoencoder algorithm and trained using the LSTM classification algorithm. Further, with the help of newly proposed IGWO techniques, the parameters are optimized to enrich the IDS model's performance.

A. DATASET GENERATION AND DESCRIPTION

The literature survey shows that a limited online open-source IoT dataset are available to train and test the IDS models. In our proposed work, we create an experimental testbed setup using a Raspberry Pi and sensors for novel dataset generation. Table 2 compares the existing IDS dataset with our experimentally generated dataset. The CICIDS 2017 and CICIDS 2018 datasets do not contain real-time devices and IoT telemetry data.

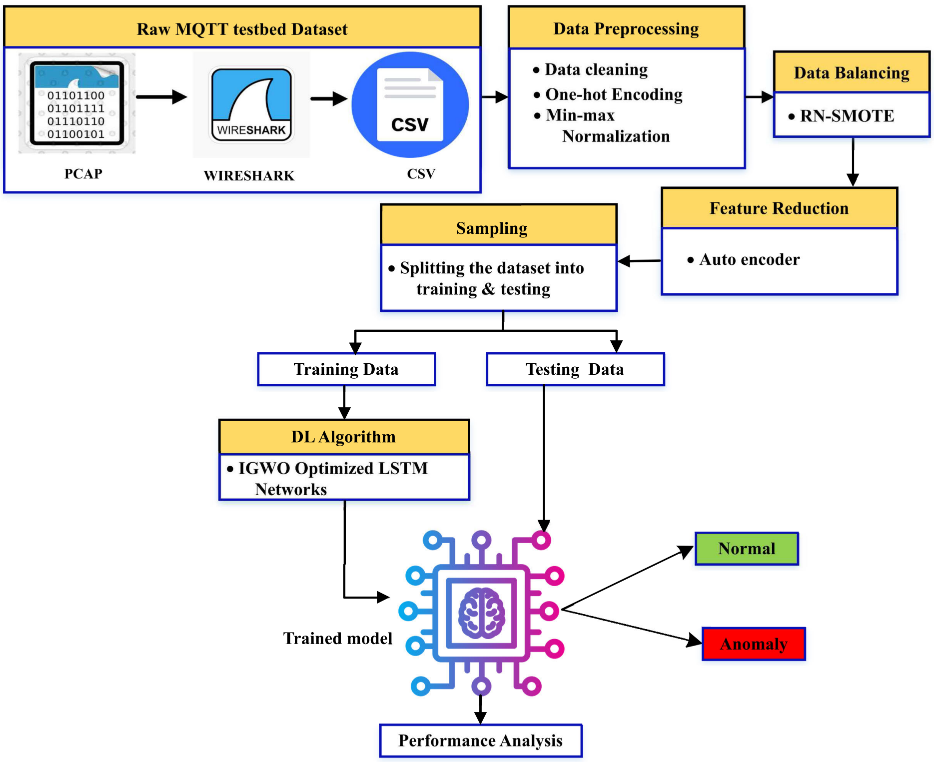


FIGURE 3. Proposed methodology.

TABLE 2. Comparison of available IDS datasets with proposed dataset.

Dataset	Year	Real time devices	Label	Device level implementation	IoT telemetry data	Multiple attackers
CICICDS 2017 [40]	2017	X	✓	✓	X	✓
CICICDS 2018 [40]	2018	X	✓	✓	X	✓
ISCX [40]	2018	✓	✓	X	X	✓
TON-IoT [41]	2019	✓	✓	X	✓	✓
MQTTset [42]	2020	✓	✓	X	✓	✓
MQTT-IDS2020 [43]	2021	X	X	X	✓	✓
Proposed	2024	✓	✓	✓	✓	✓

The ISCX, TON-IoT, and MQTTset datasets have no device-level implementation. Further, the recent MQTT-IDS 2020 has no real-time and label data. Our generated dataset has several advantages, including real-time IoT devices and device-level implementation.

1) FRAME WORK FOR ANOMALY DETECTION

A typical IoT topology has a group of actual sensors (temperature, light, and motion) connected to IoT devices through the Internet for information exchange and processing. The proposed framework is created using eight Raspberry Pi [44] (RPi) (namely RP1-RP8), eight sensors (namely S1-S8), and two personal computers, as shown in Figure 4. Here, RP1-RP4 act as publisher, RP5-RP8 act as subscriber, one PC is assigned as an MQTT broker, and one PC is assigned as an attacker. The MQTT broker acts as the network's edge device and performs all the detection on that edge device. Table 3 tabulates the IP address of each device in the framework. Table 4 shows the hardware used for our dataset generation process.

TABLE 3. Device IP address details.

Device ID	Device	IP Address	Action
RP1	Raspberry Pi 4	192.168.0.101	Publisher
RP2	Raspberry Pi 4	192.168.0.102	Publisher
RP3	Raspberry Pi 4	192.168.0.103	Publisher
RP4	Raspberry Pi 4	192.168.0.104	Publisher
RP5	Raspberry Pi 4	192.168.0.105	Subscriber
RP6	Raspberry Pi 4	192.168.0.106	Subscriber
RP7	Raspberry Pi 4	192.168.0.107	Subscriber
RP8	Raspberry Pi 4	192.168.0.108	Subscriber
PC1	Computer/edge	192.168.0.115	Broker
PC2	Computer	192.168.0.120	Attacker

A router interconnects the publisher, subscriber, and MQTT broker to monitor and control the device. We have created five scenarios for dataset generation: normal, basic connect flooding Denial-of-Service (BCF-DoS), BCF Distributed Denial-of-Service (BCF-DDoS), SYN Flooding Attack (SYN-DoS), and SYN Flooding Attack (SYN-DDoS). We extract the information for these normal and attack scenarios by capturing raw traffic PCAP files using tcpdump tools and then applying the MQTT traffic filter with Tshark.

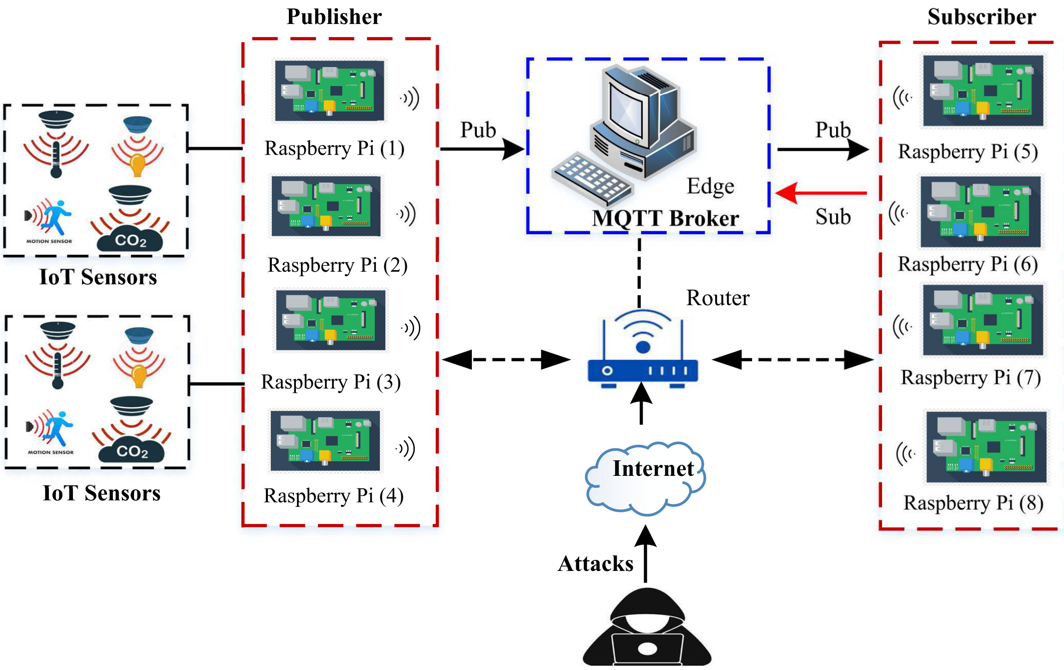


FIGURE 4. IoT Testbed used for dataset generation.

TABLE 4. Hardware devices used for the IoT testbed.

Device	Specification	Description
Raspberry Pi 4	1.8 GHz 64-bit quad-core processor, Full size HDMI, RAM-2GB LPDDR4-SDRAM, Gigabit Ethernet, Powers supply 5 V/2.5 A, Two USB 2 ports, Two USB 3 ports	i) 4 RPi's are connected with sensors and act as a publisher to send message. ii) 4 RPi's act as a subscribers to receive the published message. iii) This can be used in home, industry, and transport system.
MQTT broker (PC)	Intel 8-core i7 CPU and 8 GB RAM, Ubuntu OS 64.	To receive messages from publishers and transmit them to subscribers.
Attack PC	I5 processor: 8GB RAM, Windows 10 Pro 64-bit OS	It is used to generate four different attacks
Temperature & Humidity Sensor (DHT 11)	Temperature: 0c to 50 c Humidity Range: 20% to 90%	It is used to measure the temperature. This can be employed in smart home, Industry 4.0, smart city.
Pressure Sensor (BMP 180)	Pressure sensing range:300-1100 hPa	It measures the intensity according to the amount of light hitting on it. This can be employed in Industry 4.0
Co2 sensor (MH-Z19)	0 to 5% VOL	It used to measure the carbon amount in air. This can be employed in smart home, Industry 4.0

Figures 5(a) and 5(b) show the details of sniffed packets of tcpdump in wireshark for obtaining the individual packet details for normal and attack scenarios. The extracted features and the distribution of samples are tabulated in Table 5 and Table 6.

TABLE 5. Extracted features from testbed setup.

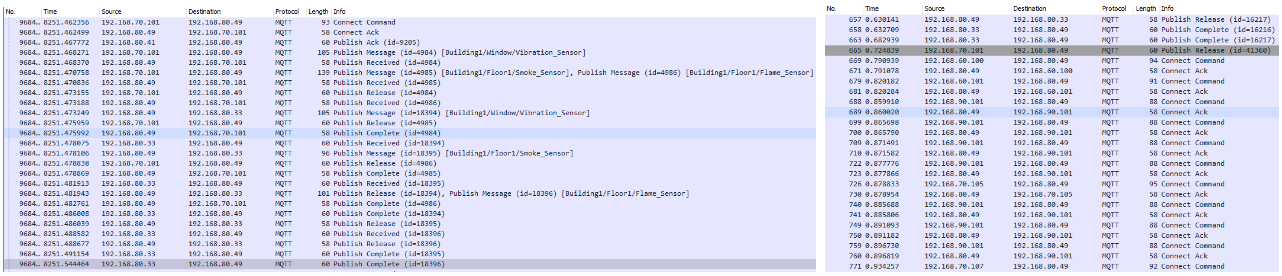
S.No	Features	S.No	Features
1	mqtt.message type	17	tcp.flags.syn
2	mqtt.qos	18	mqtt.kalive
3	mqtt.con flag.qos	19	tcp.flags.reset
4	mqtt.sub.qos	20	tcp.flags.ack
5	epoch time	21	mqtt.conflag.cleansess
6	ip.proto	22	mqtt.username-len
7	ip.source	23	mqtt.passwd-len
8	frame.len	24	mqtt.retain
9	frame.time-delta-displayed	25	mqtt.conflag.retain
10	frame.time-relative	26	mqtt.conflag.willflag
11	stream index	27	mqtt.willmsg-len
12	frame length	28	mqtt.willtopic-len
13	tcp.analysis.initial.rtt	29	mqtt.topic-len
14	tcp.time-relative	30	mqtt.len
15	tcp.seg.len	31	mqtt.conack.val
16	tcp.window-size		

TABLE 6. Testbed dataset record distributions.

Traffic Labels	Training records	Testing records
BCF-DoS	42370	10593
BCF-DDoS	188777	47194
SYN-DoS	83062	20766
SYN-DDoS	398089	99522
Normal	245786	61447

2) ATTACK MODEL

For the dataset generation, we have considered the most vulnerable MQTT DoS attack. DoS attacks exhaust the resources of the target and deny legitimate users to access the resources. Here, four types of DoS attacks are applied to the IoT network



(a) (b)

FIGURE 5. Wireshark sniffed packets for dataset generation (a) Normal (b) Attack.

for dataset generation, namely BCF-DoS, BCF-DDoS, SYN-DoS, and SYN-DDoS.

Algorithm 1 Pseudocode for BF DoS Attack [45]

Input: Broker-IP, MQTT port Number, Username, Password, Cc-connect packet count
Initializations: :
Broker-IP ← 192.168.0.115;
MQTT port Number ← 8000;
S_{id} ← MQTT-connect (1,5000);
R = CONNECT (Broker-IP, MQTT port Number, S_{id});
    Username: Manokaran;
    Password: Admin@123;
    Username-password-set(Username, Password);
C=[];
For i=1 to 11000 **do**
C[i] ← S_{id};
    Cc=C[i];
Cc[i-1].CONNECT (Broker-IP, MQTT port Number, S_{id})

B. DATA PRE-PROCESSING

Data preprocessing is a crucial processing step for any DL-based modelling, including data mining, churn prediction, and intrusion detection. The standard preprocessing procedures are data normalization, data encoding, and data cleaning. During the data cleaning stage, we eliminate irrelevant and redundant data and the empty cells are replaced with zero values. During the data encoding process, the system converts non-numeric features into numeric attributes. One frequent method for working with non-numeric features is "one-hot encoding." The output feature of the one-hot encoding is either 1's or 0's. To accelerate the speed and diminish the precision loss of the DL techniques, a popular Min-Max normalization is executed, and the old samples (χ_{Min}, χ_{Max}) are converted to new samples (F_{min}, F_{max}) using Equation (1) [46].

$$\chi_i' = F_{min} + (F_{max} - F_{min}) * (\frac{\chi_i - \chi_{min}}{\chi_{max} - \chi_{min}}) \quad (1)$$

C. DATA BALANCING

Class imbalance is one of the significant problems in the design of the DL-based IDS model. The difference between the data point count in the majority and minority classes is high, which results in the class imbalance problem. In our previous study [47], we compared various data-balancing techniques and found that the SMOTE algorithm performed well in most cases. Here, we used an improved version of the SMOTE algorithm, named RN-SMOTE, to balance the dataset [48].

1) RN-SMOTE

RN-SMOTE is a hybrid oversampling technique that combines DBSCAN and SMOTE algorithms. Initially, the unbalanced data are clustered using the DBSCAN method, and the outliers are eliminated. Then, we apply the SMOTE techniques to the clustered samples. Here, new samples are generated in the minority groups to balance the majority groups. Equation 2 indicates the calculation of the new sample ϵ_{new}. Figure 6 shows the working idea of the RN-SMOTE algorithm.

$$\epsilon_{new} = \epsilon + rand(0, 1) \times (\overline{\epsilon} - \epsilon) \quad (2)$$

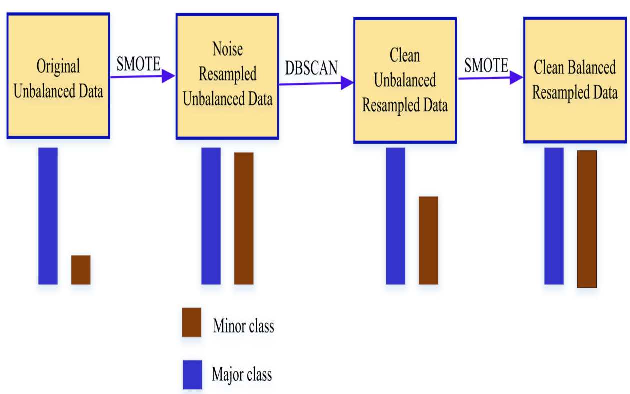


FIGURE 6. RN-SMOTE algorithm sample generation process [48].

D. FEATURE REDUCTION

Feature reduction and extraction are the primary processing steps in the model design of DL-based IDS. It plays a significant role in reducing the complexity and increasing the system's accuracy.

1) AUTOENCODER (AE)

The proposed method uses an autoencoder as a feature extraction technique. The AE is an unsupervised neural network that learns optimal encoding and decoding of information. Figure 7 illustrates the structure of AE network. There are five layers in AE: input, output, hidden, encoder, and decoder. Initially, the data is sent to an input layer (IL), where the encoder module compresses and encodes it into the hidden layer (HL). Then, the decoder reconstructs the original representation at the output layer (OL) after decoding the compressed encoding. AE aims to minimize data dimensions while maintaining the integrity of the pertinent information to prevent input repetition at the OL. AE discovers the data representation in lower dimensions by removing the noisy and uninformative characteristics that are not helpful for classification [49].

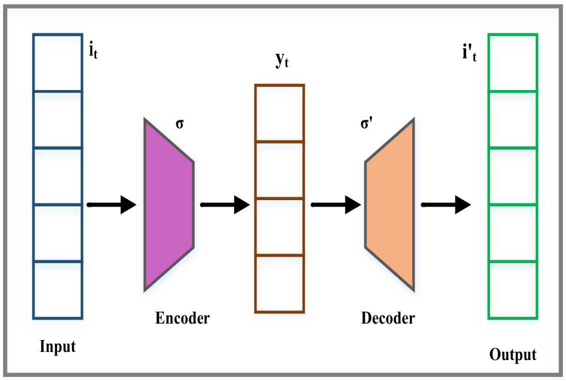


FIGURE 7. Autoencoder structure [49].

The mathematical mapping between an AE's input layer and output layer is explained as follows: Consider the given testbed generated data i_t input vector having m dimensional data, which is represented as $[i_1, i_2, i_3 \cdots i_m]$. In encoding operation, the hidden layer y_t representations are as follows,

$$y_t = \sigma(Wi_t + b) \tag{3}$$

where σ is the non-linear activation function, W is the weight matrix between the IL and HL, and b is the bias vector. In the decoder section, reconstruct the original data from the hidden layer as follows,

$$i'_t = \sigma'(W' y_t + b') \tag{4}$$

where W' is the weight matrix between the HL and OL. After then, the original data and the rebuilt result are compared, and the error is sent back across the network to update the weights. The goal of AE training is to reduce the reconstruction error, which can be expressed as the cost function c.

$$J_{AE}(c) = \frac{1}{p} \sum_{i=1}^{p} L[i_t, i'_t] \tag{5}$$

where p denotes the input feature, i_t represent the input feature at t^{th} interval, and i'_t is the reproduced output feature. $L[i_t, i'_t]$ is a reconstruction error that can be calculated by mean square error. The error value is alternatively written as,

$$L[i_t, i'_t] = \left\| i_t - i'_t \right\|^2 \tag{6}$$

$$L[i_t, i'_t] = -\frac{1}{p} \sum_{i=1}^{p} [i_t \log i'_t + (1 - i_t) log(1 - i'_t)] \tag{7}$$

Reconstruction errors are smaller when output i'_t is closer to input i_t, which implies that y_t is an effective low-dimensional feature representation.

E. RECURRENT NEURAL NETWORKS (RNN)

RNNs, which have feedback in their internal memory, are widely used in sequence learning problems. It finds application in areas such as NLP, data mining, security analysis, and intrusion detection. It also allows the network to incorporate feedback on the results of the time step before the current time step. Utilizing RNN has the benefit of providing memory cells that can operate on short-term and long-term memories. Therefore, RNN uses past information to estimate future details [50]. Figure 8 shows the detailed construction of RNN. Owing to the vast volume and complex nature of IoT data, this network cannot remember the earlier information in an ideal way, which raises a significant issue called the vanishing gradient problem. We employed an LSTM network to address this issue, which is explained in the next section.

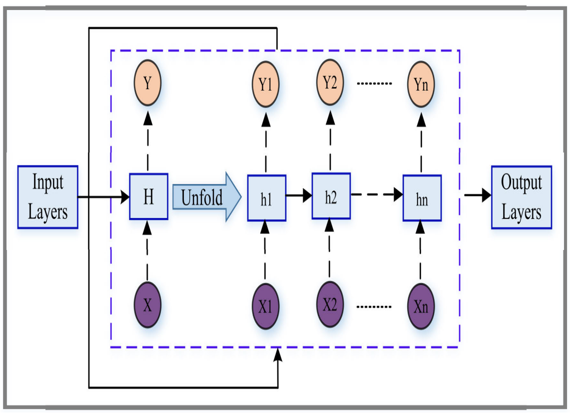


FIGURE 8. RNN structure.

1) LONG SHORT-TERM MEMORY NETWORK

LSTM is the most usable DL algorithm for various domains because of its adaptability in memory and suitability for large databases. Figure 9 shows the structure of the LSTM network, which has three stages, namely, the input gate (I.G.), the output gate (O.G.), and the forget gate (F.G.). Every gate depends on the state of the previous time step and the current input signal. Here, the gates use a sigmoid layer and a multiplication operation. The output gate specifies the new states, the forget gate determines the information to be removed from unit state C, and the input gate chooses the data needs to be stored [51], [52].

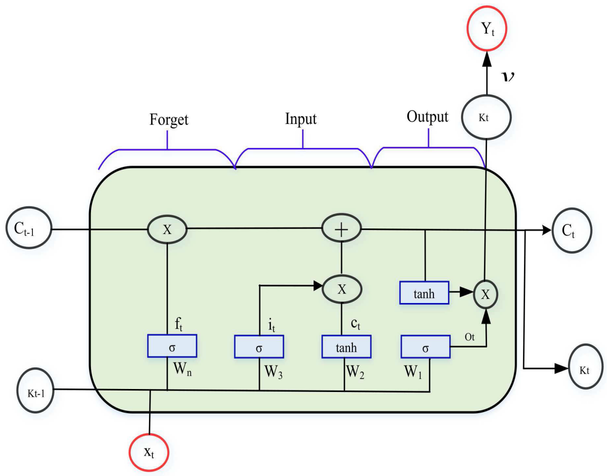


FIGURE 9. LSTM structure [51].

The mathematical working of LSTM cell is shown in Equation (8) to (13),

$$f_t = \sigma(W_n[K_{t-1}, x_t] + b_f) \tag{8}$$
$$i_t = \sigma(W_i[K_{t-1}, x_t] + b_i) \tag{9}$$
$$O_t = \sigma(W_o[K_{t-1}, x_0] + b_0) \tag{10}$$
$$\widehat{C}_t = tanh(W_c[K_{t-1}, x_0] + b_c) \tag{11}$$
$$C_t = f_t * C_{t-1} + i_t * \widehat{C}_t \tag{12}$$
$$K_t = O_t.tanh(C_t) \tag{13}$$

where f_t denotes the output of forgetting gate, C_{t-1}, C_t and $\widehat{C}_t$ denotes single memory, W_n, W_i, W_o, W_c are the weight matrixs, and b_f, b_i, b_o, b_c are the bias values, σ represents the sigmoid activation function.

F. GREY WOLF OPTIMIZATION (GWO)

Grey wolf optimizer is a swarm-based optimization technique motivated by wolves and developed by Mirjalili [16]. The author explored how the GWO algorithm can be used to solve complex engineering problems. It strictly follows the hierarchy order, which is shown in Figure 10. The author demonstrated that the performance of the GWO algorithm is superior to various popular meta-heuristic optimizers, namely Particle Swarm Optimization (PSO), Aquila Optimizer (AO), Ant Colony Optimization (ACO), and Dung Beetle Optimizer (DBO). PSO is a well-known intelligent searching technique based on the idea of bird predation. The actions of Aquila's in the wild when attempting to capture their prey serve as the model for AO algorithm. Ants' foraging behavior serves as the model for ACO algorithm. The idea behind the DBO techniques came from the ways that dung beetles reproduce, search, and roll balls. These optimization techniques are all limited by premature convergence and local entrapment.

The mathematical functions of GWO are listed below, Let the position of the four wolves are represented in examination space as $P_\alpha, P_\beta, P_\delta, P_\omega$. The initial process is prey encircling, which is represented in Equation (14)-(18).

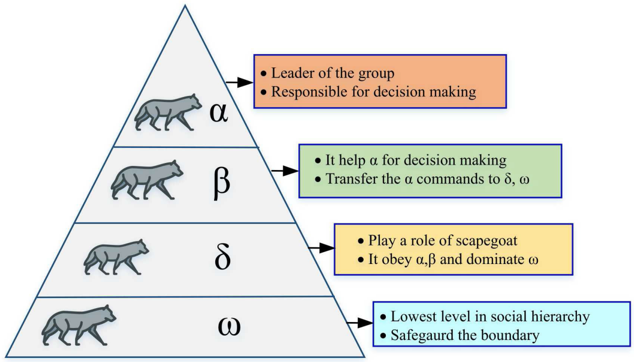


FIGURE 10. Hierarchical levels of GWO algorithm [16].

$$\vec{D} = \left|\vec{C} \cdot \vec{P}_{prey}(t) - \vec{P}(t)\right| \tag{14}$$
$$\vec{P}(t+1) = \vec{P}_{prey}(t) - \vec{A} \cdot \vec{D} \tag{15}$$

where $\vec{P}(t)$, $\vec{P}_{prey}(t)$ indicates the position vector of wolves and prey at current iteration. $\vec{A}, \vec{C}$ are coefficient vectors, calculated as shown below,

$$\vec{C} = 2\vec{a} \cdot rand_1 - \vec{a} \tag{16}$$
$$\vec{A} = 2 \cdot rand_2 \tag{17}$$
$$\vec{a} = 2 - \left(\frac{2 * t}{Iteration_{max}}\right) \tag{18}$$

$rand_1, rand_2$ indicate random value between [0, 1]. $\vec{a}$ is a vector whose value decreases from two to zero, bringing the wolves closer to the prey with each iteration. The position renovating of wolves is shown in Figure 11. The prey position update is calculated using Equation 19

$$\vec{P}_{t+1} = \frac{\vec{P}_1 + \vec{P}_2 + \vec{P}_3}{3} \tag{19}$$

where,

$$\vec{P}_1 = \vec{P}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha, \ \vec{P}_2 = \vec{P}_\beta - \vec{A}_2 \cdot \vec{D}_\beta, \ \vec{P}_3 = \vec{P}_\delta - \vec{P}_3 \cdot \vec{D}_\delta \tag{20}$$

$$\vec{D}_\alpha = \left|\vec{C}_1 \cdot \vec{P}_\alpha - \vec{P}\right| \tag{21}$$
$$\vec{D}_\beta = \left|\vec{C}_2 \cdot \vec{P}_\beta - \vec{P}\right| \tag{22}$$
$$\vec{D}_\delta = \left|\vec{C}_3 \cdot \vec{P}_\delta - \vec{P}\right| \tag{23}$$

Usually, researchers optimize the objective function of GWO by abstracting it as a random search problem in multidimensional space. For every wolf, the fitness value is be calculated in accordance to the fitness function. Equation 24 shows the fitness function of our parameter tuning problem and our aim is to minimize the fitness function [53].

$$Fitness = Error\ rate = 1 - Accuracy \tag{24}$$

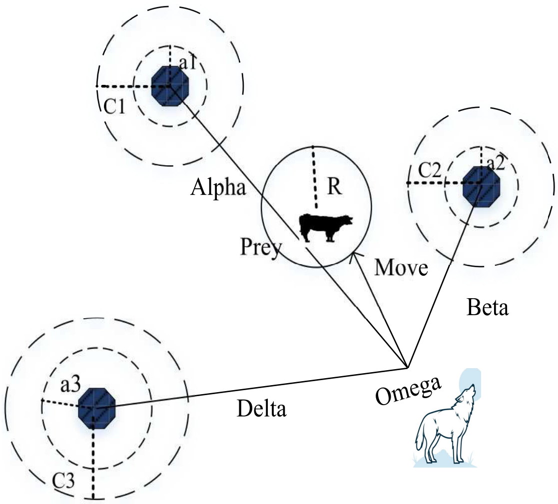


FIGURE 11. Position updating of GWO [16].

1) IMPROVED GREY WOLF OPTIMIZATION

Though GWO excels at solving many optimization problems, it is possible that GWO may converge prematurely to suboptimal solutions when applied to problems with high dimensions. We propose an IGWO algorithm instead of a traditional GWO algorithm to balance exploration and exploitation properly. It has two modifications, like i) Elimination Mechanism [54] ii) Opposition-Based Learning method [55]. The work flow and algorithm are shown in Figures 12 and 13.

First, adjust the algorithm using the EM principle to prevent the wolves from entering the local optimum. Next, eliminate R wolves with the lowest fitness value after each algorithm iteration. In the meantime, we generate R new wolves using the OBL technique to subdue the issues of computational expensiveness and time consumption.

Let $m \in [x,y]$ be a real number. Its opposite m^o is represented as follow:

$$m^o = lb_i + ub_i - m \tag{25}$$

The same principles apply to E-multi-dimensional search spaces as well. Let $q = \{m_1, m_2, \ldots m_E\}$, and $m_i \{1 \leq i \leq E\}$ is a range of lb_i (lower boundary) and ub_i (upper boundary). The opposite point $\vec{q}^0 = \left\{\vec{m}_1^0, \vec{m}_2^0, \ldots \vec{m}_E^0\right\}$ can be obtained using equation (26)

$$\vec{m}_i^0 = lb_i + ub_i - m_i \tag{26}$$

2) COMPLEXITY ANALYSIS

The computational complexity of the IGWO algorithm is covered in this section. Let PC represents the size of the wolf population, E_{dim} is the problem's dimension, and $Iteration_{max}$ is the maximum number of iterations. At the end of every cycle, the least wealthy R wolves are eliminated, and R new wolves are generated in accordance with the OBL method. The wolf selection procedure requires O(PC), the position-updating technique requires O(PC $\times$ E_{dim}), and the OBL evaluation function requires O(PC). The total computing complexity of each iteration is therefore O(PC $\times$ E_{dim}). The computing complexity of the complete iteration is O(PC $\times$ E_{dim} $\times$ $Iteration_{max}$). As a result, it is consistent with the conventional GWO and the computational complexity has not raised [56]. In our proposed model the number of dimensions (E_{dim}) are minimized using AE algorithm and this reduces the complexity while detecting anomaly in IoT-edge scenarios.

The number of computations involved in the proposed IGWO-AE-LSTM is greater than the traditional models available for network anomaly detection in the IoT, which increases the time delay. The experimental results show that the performance (accuracy, precision, recall, and F1 score) of the proposed DL-ADS is superior compared to the current state-of-the-art methods. So there is a performance-time delay complexity trade-off for obtaining high-accuracy anomaly detection in IoT edge scenarios.

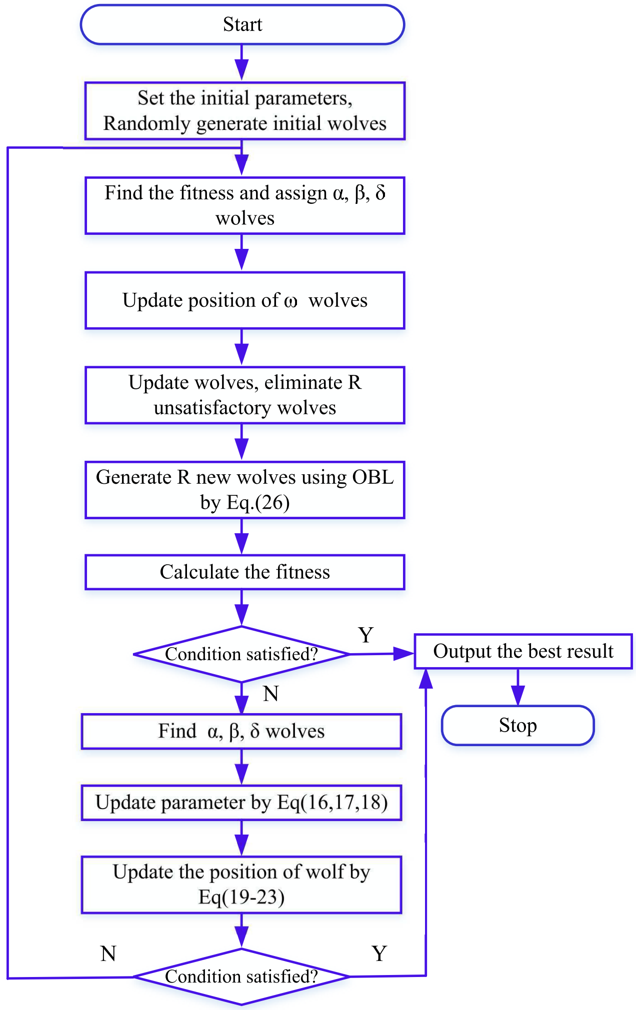


FIGURE 12. Work-flow of the developed IGWO algorithm.

```
Proposed IGWO algorithm
Initialize: The positions of the wolves, Pi (i=1, 2,…, n)
Initialize: The GWO parameters a, A, and C
    Find the fitness of each searching wolf
    Pα, Pβ, Pδ – Three best searching wolf
    While (t < Iteration_max)
        For each individual do
            Update the position of the current individual by equation (19)
        end
        Update a, A, and C
        Calculate the fitness of each individual
        For (i=0; i< W_best; i++)
            For (j=0; j<E; j++)
                Eliminate R less effective wolves
                Create R new wolves using equation (26)
            end
        end
        Evaluate the fitness function
      While (t<Iteration_max)
          Find Pα, Pβ, Pδ
          Update a, A, and C using equation (16 - 18)
          For (i=0; i< W_best ; i++)
                Update the position using equation (19 - 23)
          end
          t=t+1
      end
    end
    Return Pα
```

FIGURE 13. Pseudocode of IGWO algorithm.

IV. PERFORMANCE ASSESSMENT

This section explains the evaluation of the developed DL-based anomaly detection system. We evaluate the developed model using standard metrics such as accuracy, FAR, precision, ROC curve, recall, and F1 score. Data associated with anomaly is represented by the first four entries, while normal data is represented by the fifth entry in the class label. Figure 14 (a) demonstrates the confusion matrix for multi-classification with five class labels. Here, N_{11} to N_{44} represent the number of attack data predicted as attacks, and N_{55} represents the number of normal data envisioned as normal. Figure 14 (b) shows the confusion matrix for the bi-classification problem. The evaluation parameters are calculated using (Equations 27 to 31). The aim of the proposed model is to increase the average accuracy [56].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{27}$$

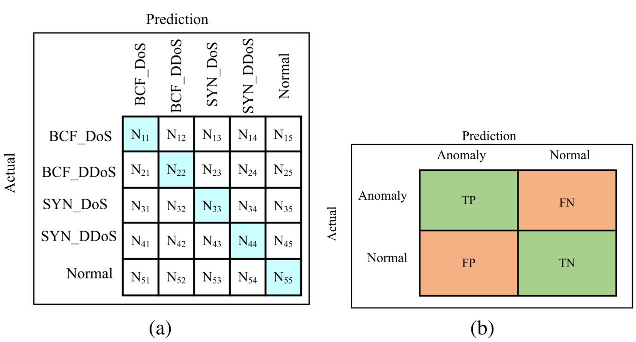


FIGURE 14. (a) Confusion matrix for multi-classification (b) Confusion matrix for bi-classification.

$$Precision = \frac{TP}{TP + FP} \tag{28}$$

$$Recall = \frac{TP}{TP + FN} \tag{29}$$

$$F1 - score = \frac{2 * P * R}{P + R} \tag{30}$$

$$FAR = \frac{FP}{FP + TN} \tag{31}$$

A. EXPERIMENTAL FRAME-UP AND INVESTIGATION

The developed model's performance is analyzed using different frame-up, such as in frame-up I; the performances of DL algorithms with and without feature reduction techniques are compared based on standard metrics values for all datasets. In frame-up II, the DL algorithm is tuned using the IGWO technique for performance enhancement and evaluation. In frame-up III, the proposed model's performance is compared with the existing methods.

We have implemented our IGWO-AE-LSTM model on a Google's Colaboratory using Scikit-learn and the Keras frameworks. Python-based trials are executed on an Intel Core i7, CPU processor running 64-bit Windows 10 and 64 GB of RAM in order to evaluate the proposed approach. Three classifiers are considered along with four intrusion datasets, which results in 12 combinations of experiments. Additionally, for cross-validation, 20 runs are completed, and the average values are tabulated for analysis. Figure 15 shows the experimental testbed with Raspberry Pi, sensors, PCs, and a router.

1) FRAME UP I

The performance of multi-classification anomaly detection using LSTM network before and after feature reduction of different datasets are shown in Tables 7-14. Table 7 represents the performance of the testbed dataset using the LSTM network. The multi-classification prediction accuracy for different classes are: BCF-DoS 96.88%, BCF-DDoS 96.52%, SYN-DoS 97.07%, SYN-DDoS 95.88%, and Normal 95.82%. Table 8 represents the performance of the CICIDS 2017 dataset using the LSTM network. The dataset has seven classes, and the prediction accuracy for those different classes are: Normal 93.61%, Bot 99.98%, Brute force 98.48%, DOS/DDoS 95.36%, Infiltration 100%, Port scan 96.94% and Web attack 99.97%. Table 9 represents the performance of the DS2OS dataset using the LSTM network. The dataset has eight classes, and the prediction accuracy for those different classes are: DoS 95.02%, Data type 99.90%, Malicious control 99.77%, Malicious operation 99.90%, Scan 99.78%, Spying 99.91%, Wrong setup 99.95% and Normal 94.24%. Table 10 represents the performance of the MQTTset dataset using the LSTM network. The dataset has six classes, and the prediction accuracy for those different classes are: Normal 95.17%, Flood 99.91%, DoS 95.34%, Brute force 98.39%, Malformed 98.45%, and SlowITe 99.99%. Even though the performance of the LSTM prediction model is good, to further reduce the complexity

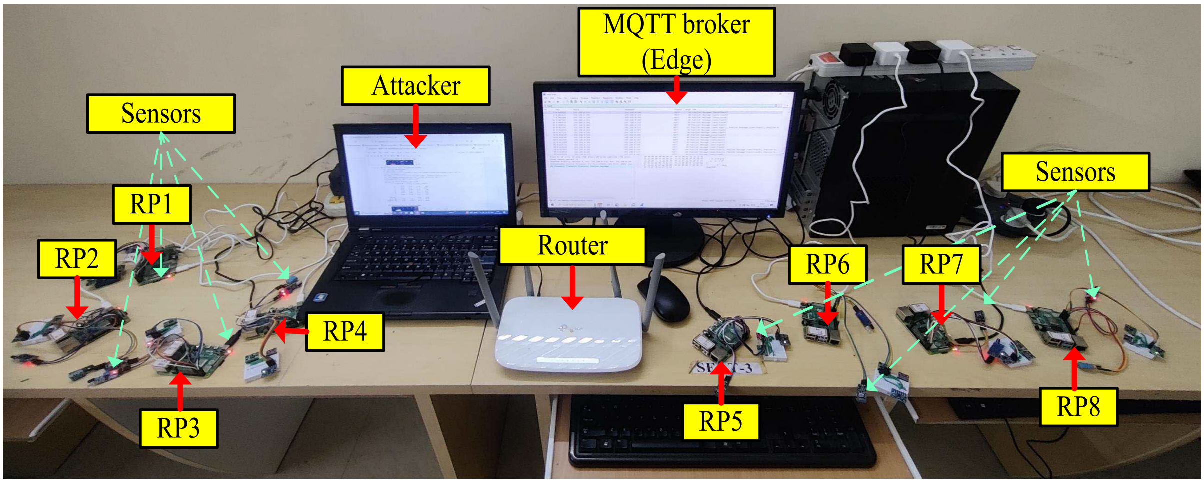


FIGURE 15. Experimental testbed.

and improve precision, recall, and FAR, we have applied the autoencoder network as a feature reduction technique, and the results are tabulated in Tables 11-14.

TABLE 7. Performance of multi-classification testbed dataset using LSTM network without AE.

Classes	Accuracy	Precision	Recall	F-Score	FAR
BCF-DoS	96.88	64.32	66.20	65.25	1.70
BCF-DDoS	96.52	90.81	91.64	91.22	2.28
SYN-DoS	97.07	83.19	82.97	83.08	1.59
SYN-DDoS	95.88	95.03	95.05	95.04	3.53
Normal	95.82	92.34	91.28	91.81	2.61

TABLE 8. Performance of multi-classification CICIDS 2017 dataset using LSTM network without AE.

Classes	Accuracy	Precision	Recall	F-Score	FAR
Normal	93.61	97.33	94.63	95.96	10.50
Bot	99.98	87.91	78.43	82.90	0.01
Brute force	98.48	49.18	67.65	49.89	1.37
DOS/DDoS	95.36	83.05	82.63	82.84	2.64
Infiltration	100	50	100	66.67	0.00
Port scan	96.94	69.04	82.75	75.27	2.21
Web attack	99.97	93.75	67.16	78.26	0.01

TABLE 9. Performance of multi-classification DS2OS dataset using LSTM network without AE.

Classes	Accuracy	Precision	Recall	F-Score	FAR
DoS	95.02	52.25	72.89	57.58	3.97
Data Type	99.90	45.45	43.86	44.64	0.05
Malicious Control	99.77	54.36	55.12	54.17	0.12
Malicious Operation	99.90	75.58	80.75	78.08	0.06
Scan	99.78	73.17	77.57	75.41	0.12
Spying	99.91	74.16	62.03	67.59	0.03
Wrong Setup	99.95	36.36	67.69	46.68	0.04
Normal	94.24	98.46	95.56	96.99	11.76

TABLE 10. Performance of multi-classification MQTTSet dataset using LSTM network without AE.

Classes	Accuracy	Precision	Recall	F-Score	FAR
Normal	95.17	97.46	92.75	95.05	2.41
Flood	99.91	98.96	51.63	67.86	0.01
DoS	95.34	90.90	97.96	94.30	6.37
Brute force	98.39	79.77	84.81	82.21	0.99
Malformed	98.45	88.53	61.01	72.24	0.27
SlowITe	99.99	99.99	99.89	99.95	0.01

Tables 11-14 show the performance of multi-classification anomaly detection using an LSTM network with 16 selected dimensional features. There is a significant effect of feature reduction on different datasets. The multi-classification performance of the testbed dataset using LSTM network without feature reduction technique are 96.66%, 96.52%, 97.07%, 95.88%, and 95.82% for different classes. Similarly, the performances achieved using the LSTM network with feature reduction technique are 99.25%, 98.77%, 99.13%, 98.78%, and 98.46% for different classes. The multi-classification performance of the MQTTset dataset using LSTM network without feature reduction technique are 95.17%, 99.91%, 95.34%, 98.39, 98.45%, and 99.99% for different classes. Similarly, the performances achieved using the LSTM network with feature reduction technique are 98.42%, 99.96%, 98.34%, 99.42%, 99.54%, and 99.99% for different classes. The above discussions infer that the performance of the LSTM network with feature selection is superior to the performance of the LSTM network without feature selection technique, irrespective of the dataset. Compared to Tables 7-10 of the full features, Tables 11-14 of the selected features have better performances in terms of accuracy, precision, recall, F-score, and FAR. Figure 16 shows the performance comparison of the AE-LSTM model with different datasets (Testbed, CICIDS 2017, DS2OS, and MQTTSet).

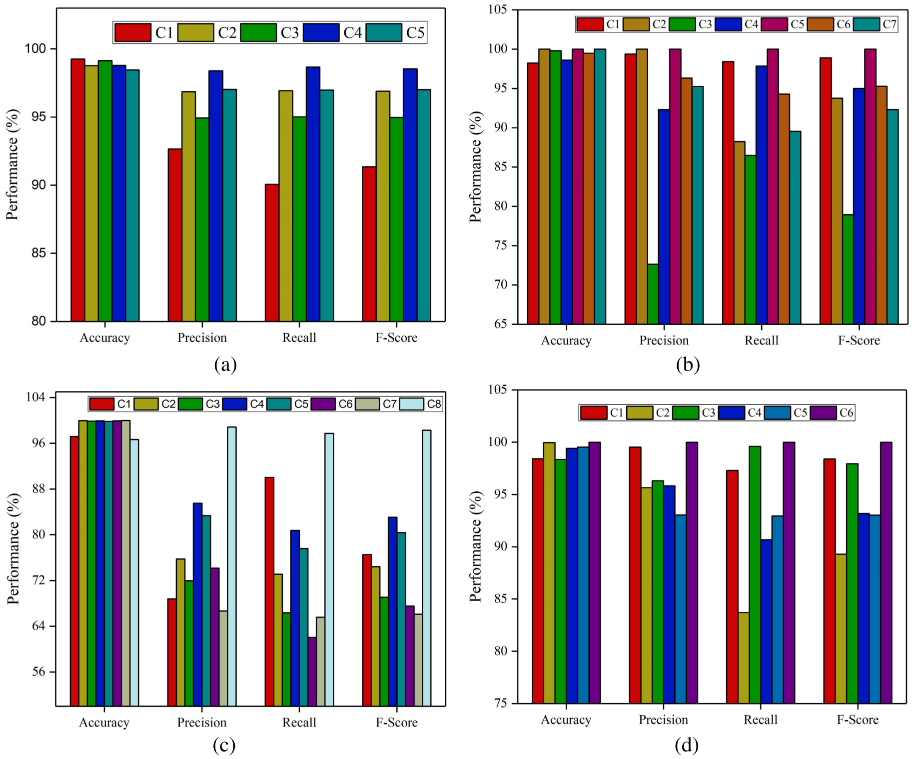


FIGURE 16. Multi-classification performance comparison using AE-LSTM network (a) Testbed (b) CICIDS 2017 (c) DS20S (d) MQTTset.

TABLE 11. Performance of multi-classification testbed dataset using LSTM network with AE.

Classes	Accuracy	Precision	Recall	F-Score	FAR
BCF-DoS	99.25	92.66	90.07	91.35	0.33
BCF-DDoS	98.77	96.85	96.93	96.89	0.77
SYN-DoS	99.13	94.93	95.01	94.97	0.48
SYN-DDoS	98.78	98.39	98.67	98.53	1.14
Normal	98.46	97.03	96.97	97.01	1.02

TABLE 12. Performance of multi-classification CICIDS 2017 dataset using LSTM network with AE.

Classes	Accuracy	Precision	Recall	F-Score	FAR
Normal	98.22	99.37	98.41	98.89	2.54
Bot	99.99	99.99	88.24	93.75	0.01
Brute force	99.78	72.62	86.47	78.94	0.16
DOS/DDoS	98.60	92.30	97.84	94.99	1.28
Infiltration	100	100	100	100	0.00
Port scan	99.47	96.31	94.26	95.27	0.22
Web attack	99.99	95.24	89.55	92.31	0.01

2) FRAME UP II

In frame-up II, the AE-LSTM network is optimized using the IGWO technique for performance enhancement and evaluation. The initial parameter values of the GWO algorithm based on the contemporary method are listed in Table 15. Figure 17 shows the diverse probable values of hyper-parameter and the nominated optimal parameter of the LSTM network for all datasets, namely testbed, CICIDS 2017, DS2OS, and MQTTset.

TABLE 13. Performance of multi-classification DS2OS dataset using LSTM network with AE.

Classes	Accuracy	Precision	Recall	F-Score	FAR
DoS	97.19	68.79	90.02	76.54	2.03
Data Type	99.95	75.76	73.10	74.40	0.02
Malicious Control	99.85	71.95	66.37	69.05	0.06
Malicious Operation	99.93	85.53	80.75	83.07	0.03
Scan	99.84	83.33	77.57	80.35	0.07
Spying	99.91	74.16	62.03	67.59	0.03
Wrong Setup	99.98	66.67	65.57	66.12	0.01
Normal	96.67	98.84	97.72	98.28	9.87

TABLE 14. Performance of multi-classification MQTTSet dataset using LSTM network with AE.

Classes	Accuracy	Precision	Recall	F-Score	FAR
Normal	98.42	99.53	97.30	98.40	0.46
Flood	99.96	95.65	83.70	89.29	0.04
DoS	98.34	96.31	99.60	97.93	2.48
Brute force	99.42	95.84	90.65	93.17	0.18
Malformed	99.54	93.04	92.95	93.01	0.24
SlowITe	99.99	99.99	99.99	99.99	0.01

TABLE 15. **Tuning values of GWO.**

S.No	Items	Values
1	Number of search wolves	12
2	Number of iterations	400
3	Clusters names	α, β, δ
4	Range of search space	[0,1]
5	Coefficient vector	2

Model	Hyperparameters	Range value	Best parameter value D1	D2	D3	D4
LSTM	Activation Function	{Sigmod, tanh, ReLU}	ReLU	ReLU	ReLU	Sigmod
	Epochs	{50,100,150,200,250}	250	200	150	100
	Batch Size	{32,64,128,256,512}	128	128	32	64
	Optimizer	{Adam, rmsprop, SGD}	Adam	Adam	SGD	Adam
	Layers	{1,2,3,4,5,6}	4	3	4	2
	Learning rate	{0.1,0.01,0.001,0.0001}	0.001	0.002	0.001	0.01

FIGURE 17. **The description of hyper parameters search and their optimal values on each dataset.**

a: DISCOVERING THE BEST GWO ITERATIONS

In DL models, determining the maximum number of algorithm iterations necessary to achieve a stable and low error rate is a crucial. Thoroughly testing the model is necessary to ensure consistent performance over many iterations. The minimum error rate evolution and the number of iterations for the suggested model using various datasets are presented in Figure 18. All datasets show a decrease in error rate with increasing iterations, as depicted in the figure. The optimal iteration points for the testbed, CICIDS 2017, DS2OS, and MQTTset datasets are 80, 78, 40, and 58, respectively.

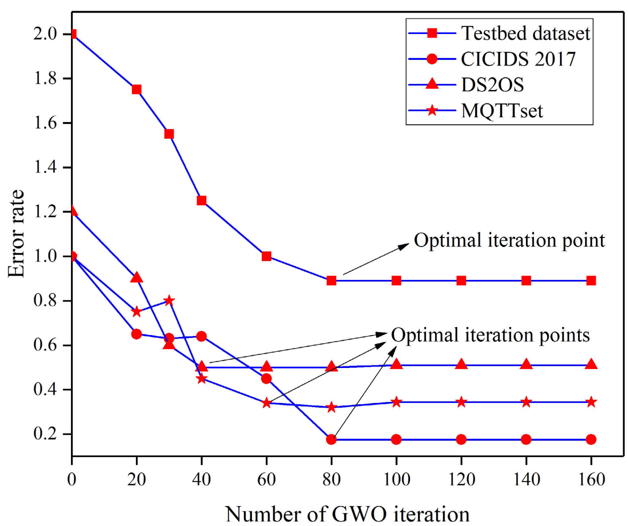


FIGURE 18. **Error rate convergence with iteration of IGWO.**

Tables 16-19 display the performance of the parameter-tuned AE-LSTM network for the different datasets. Table 16 shows the performance of the testbed dataset with IGWO optimized AE-LSTM network. Compared to Table 11 of the traditional LSTM network, the accuracy value is increased for different classes, like BCF-DoS 99.25% to 99.39%, BCF-DDoS 98.77% to 98.96%, SYN-DoS 99.13% to 99.42%, SYN-DDoS 98.78% to 99.04%, and Normal 98.46% to 98.81%. Further, precision, recall, F-score, and FAR also show better performance for the optimized AE-LSTM network compared to the normal AE-LSTM network. Tables 17 and 19 show the performance of the IGWO-AE-LSTM network for different datasets, such as the CICIDS 2017, DS2OS and MQTTset. Here also, the proposed model has the highest detection accuracy, precision, recall, F-score, and FAR for different classes. Figure. 19 compares the proposed model's performance for the Testbed, CICIDS 2017, DS2OS, and MQTTset datasets. From the figure, the proposed IGWO-AE-LSTM algorithm performs better in all parameters than other algorithms.

TABLE 16. **Performance of multi-classification testbed dataset using Optimized AE-LSTM network.**

Classes	Accuracy	Precision	Recall	F-Score	FAR
BCF-DoS	99.39	95.07	91.03	93.01	0.22
BCF-DDoS	98.96	97.45	97.25	97.35	0.62
SYN-DoS	99.42	96.85	96.39	96.62	0.30
SYN-DDoS	99.04	98.80	98.89	98.85	0.86
Normal	98.81	97.26	98.13	97.67	0.95

TABLE 17. **Performance of multi-classification CICIDS 2017 dataset using Optimized AE-LSTM network.**

Classes	Accuracy	Precision	Recall	F-Score	FAR
Normal	99.49	99.87	99.49	99.68	0.51
Bot	99.99	99.99	94.12	96.97	0.01
Brute force	99.96	97.07	93.85	95.43	0.01
DOS/DDoS	99.58	97.34	99.64	98.48	0.43
Infiltration	100	100	100	100	0.00
Port scan	99.90	99.02	99.17	99.10	0.06
Web attack	99.99	95.52	95.52	95.52	0.01

TABLE 18. **Performance of multi-classification DS2OS dataset using Optimized AE-LSTM network.**

Classes	Accuracy	Precision	Recall	F-Score	FAR
DoS	98.35	89.10	97.61	93.02	0.98
Data Type	99.97	90.91	73.10	81.04	0.01
Malicious Control	99.89	85.51	68.37	74.73	0.03
Malicious Operation	99.94	92.86	80.75	86.38	0.01
Scan	99.84	85.11	77.57	81.16	0.06
Spying	99.91	75.86	65.03	68.25	0.03
Wrong Setup	99.98	76.10	69.79	70.48	0.01
Normal	97.91	98.98	98.87	98.82	5.49

The performance of the IGWO-AE-LSTM model in terms of training accuracy and testing accuracy for anomaly detection are elaborately shown in Figures 20 (a) and (b), and the training loss and testing loss for anomaly detection are shown in Figures 20 (c) and (d). Overall, the figures show an increase in the accuracy of training and validation and a decrease in the loss of training and validation till 250 epochs. Since there was no improvement above

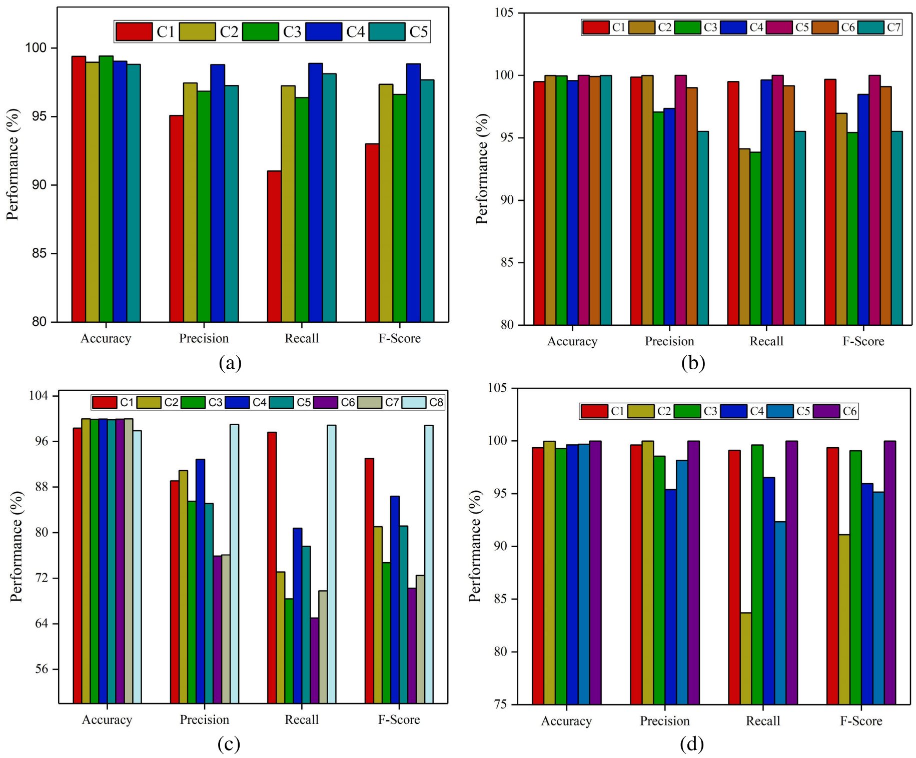


FIGURE 19. Performance comparison of proposed method (a) Testbed (b) CICIDS 2017 (c) DS2OS (d) MQTTset.

TABLE 19. Performance of multi-classification MQTTset dataset using Optimized AE-LSTM network.

Classes	Accuracy	Precision	Recall	F-Score	FAR
Normal	99.37	99.62	99.11	99.36	0.38
Flood	99.97	99.99	83.70	91.12	0.01
DoS	99.28	98.55	99.62	99.08	0.95
Brute force	99.64	95.39	96.53	95.96	0.21
Malformed	99.69	98.15	92.34	95.16	0.06
SlowITe	99.99	99.99	99.99	99.99	0.01

250 epochs, we decided to stop the experiments. A categorical cross-entropy loss function was used in this study, which is mathematically described as an Equation. (32):

$$L_{(c,p)} = -\sum_{j=1}^{n} C_j \, ln \, (P)_j \tag{32}$$

where c denotes the true value while the predicted value is $\widehat{y}$, n denotes the number of classes, and p denotes the probability distribution of j^{th} observed value. The ROC curve is drawn between TPR and FPR from Equation 33 and 34. Figure 21 shows the multi-classification ROC curve of our proposed system for different datasets (Testbed, CICIDS 2017, DS2OS, and MQTTSet). The AUC values of the different classes are denoted individually in the Figure 21.

$$TPR = \frac{TP}{TP + FN} \tag{33}$$

$$FPR = \frac{FP}{FP + TN} \tag{34}$$

b: STATISTICAL ANALYSIS BASED ON WILCOXON SIGNED-RANK TEST

Statistical analysis is mainly used to test null and alternative hypotheses. The null hypothesis states that the developed and existing methods do not differ significantly. Meanwhile, the alternative hypothesis claims that these methods are significantly different. The significance of the results attained can be expressed using p-value, where the p-value should be less than 0.05 [57]. We use the Wilcoxon signed-rank test to compare the pairwise differences between the proposed model and the basic LSTM model using the p value [58]. *scipy.stats.wilcoxon*() function in Python is used to find the p value. We performed this test for every dataset using the proposed model and compared models. We used the accuracy value as a statistical evaluation measurement to differentiate between the compared models. It can be observed from Table 20, that the p-value obtained for all four datasets

TABLE 20. Wilcoxon signed-rank test results.

S.No	Dataset	P value
1	CICIDS 2017	0.0270
2	DS2OS	0.0179
3	MQTTset	0.0430
4	Testbed	0.0062

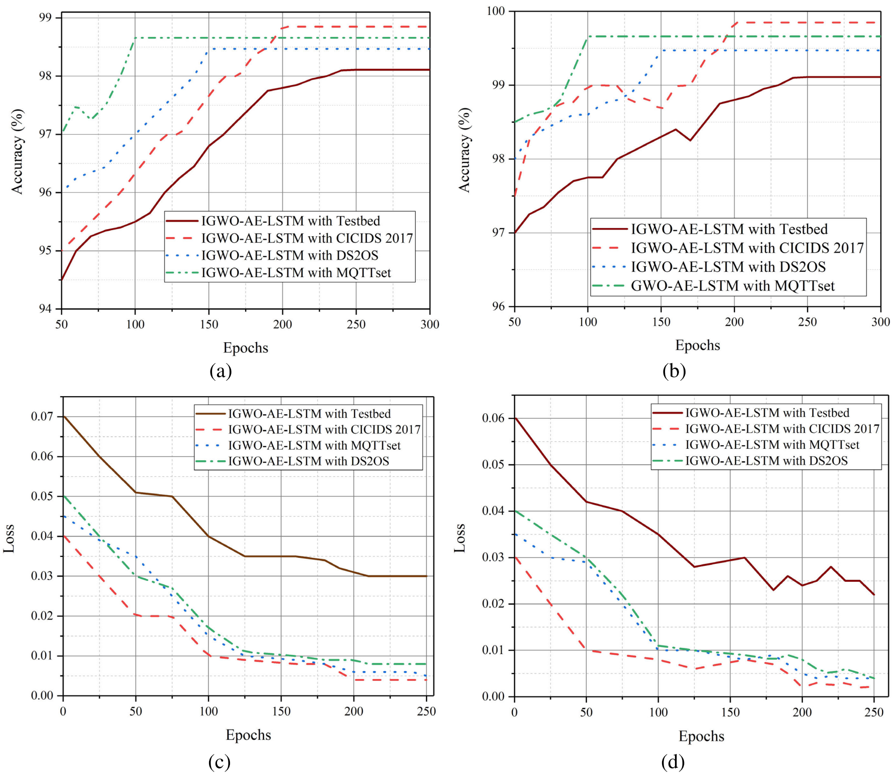


FIGURE 20. Model performance comparison (a) Training accuracy evaluation (b) Testing accuracy evaluation (c) Training loss rate evaluation (d) Testing loss rate evaluation.

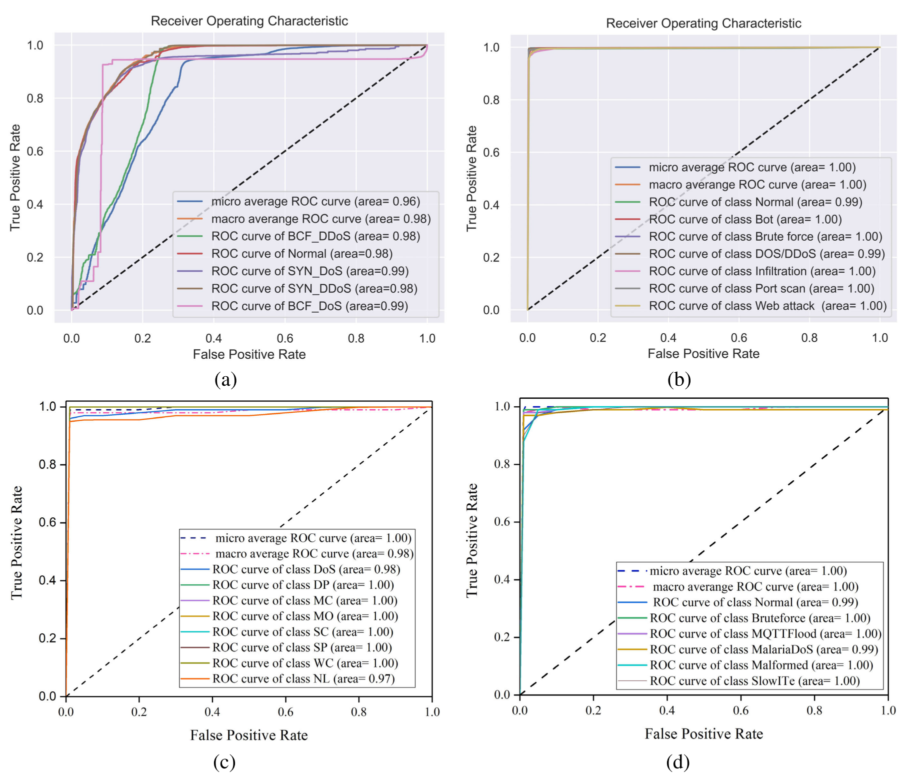


FIGURE 21. ROC curve for each of the attacks in the dataset (a) Testbed (b) CICIDS 2017 (c) DS20S (d) MQTTset.

considered for our study is less than 0.05. Hence, the results attained are statistically significant.

3) FRAME UP III

a: COMPARISON OF PARAMETER OPTIMIZATION METHODS WITH EXISTING TECHNIQUES

Our proposed AE-LSTM model is optimized using some other standard hyper-parameter tuning algorithms like PSO and GWO techniques. Tables 21-24 compare and tabulate the performances of the existing optimizer with different results.

TABLE 21. Performance of the testbed dataset with existing optimizer.

Class	PSO-AE-LSTM		GWO- AE-LSTM		IGWO-AE-LSTM	
	Acc	FAR	Acc	FAR	Acc	FAR
BCF-DoS	99.12	0.35	99.32	0.25	99.39	0.22
BCF-DDoS	98.71	0.73	98.86	0.67	98.96	0.62
SYN-DoS	99.14	0.40	99.35	0.32	99.42	0.30
SYN-DDoS	98.86	1.12	98.96	0.92	99.04	0.86
Normal	98.42	1.24	98.64	1.11	98.81	0.95

TABLE 22. Performance of the CICIDS 2017 dataset with existing optimizer.

Class	PSO-AE-LSTM		GWO- AE-LSTM		IGWO-AE-LSTM	
	Acc	FAR	Acc	FAR	Acc	FAR
Normal	99.06	1.64	99.14	1.55	99.49	0.51
Bot	99.99	0.01	99.99	0.01	99.99	0.01
Brute force	99.94	0.03	99.96	0.01	99.96	0.01
DOSDDoS	99.19	0.63	99.24	0.58	99.58	0.43
Infiltration	100	0.00	100	0.00	100	0.00
Port scan	99.87	0.07	99.89	0.06	99.90	0.06
Web attack	99.99	0.01	99.99	0.01	99.99	0.01

TABLE 23. Performance of the DS2OS dataset with existing optimizer.

Class	PSO-AE-LSTM		GWO- AE-LSTM		IGWO-AE-LSTM	
	Acc	FAR	Acc	FAR	Acc	FAR
DoS	98.00	1.30	98.14	1.10	98.35	0.98
Data Type	99.96	0.01	99.96	0.01	99.97	0.01
Malicious Control	99.86	0.06	99.87	0.05	99.89	0.03
Malicious Operation	99.92	0.03	99.93	0.02	99.94	0.01
Scan	99.76	0.11	99.78	0.11	99.84	0.06
Spying	99.89	0.05	99.90	0.04	99.91	0.03
Wrong Setup	99.97	0.02	99.98	0.01	99.98	0.01
Normal	97.39	5.85	97.59	5.78	97.91	5.49

These tables reveal that higher accuracy is achieved in our proposed model in comparison to the existing studies. We also verified the proposed method using the standard datasets (DS2OS, CICIDS 2017, MQTTset). Additionally, we evaluated the developed IGWO-AE-LSTM method by comparing it with other existing models and verified it on the benchmark dataset. Table 25 reveals that our proposed model is superior to most of the existing anomaly detection models in the IoT field.

TABLE 24. Performance of the MQTTset dataset with existing optimizer.

Class	PSO-AE-LSTM		GWO- AE-LSTM		IGWO-AE-LSTM	
	Acc	FAR	Acc	FAR	Acc	FAR
Normal	99.18	0.50	99.24	0.46	99.37	0.38
Flood	99.97	0.01	99.97	0.01	99.97	0.01
DoS	99.11	1.09	99.11	1.09	99.28	0.95
Brute force	99.57	0.27	99.57	0.27	99.64	0.21
Malformed	99.66	0.08	99.66	0.08	99.69	0.06
SlowITe	99.99	0.01	99.99	0.01	99.99	0.01

TABLE 25. Comparison table of the developed model with present IDS models.

Study	Feature Extraction	Model	Dataset	Accuracy
[21]	Nil	CNN, LSTM	NSL KDD, CICIDS 2017	85.24, 99.91
[22]	CNN	LSTM	Self-generated	96
[23]	Nil	Hybrid DRNN	DS2OS, UNSW NB 15	98, 99
[24]	AE	LSTM	NSL KDD	89
[25]	CNN	ChCSO-LSTM	NSL KDD, BoT-IoT	93.31, 94.25
[26]	AE	B-Stacking	NSL KDD, CICIDS 2017	98.50, 99.11
[27]	CNN	LSTM	CICIDS 2017, UNSW NB 15, WNS DS	99.64, 94.53, 99.67
[28]	CNN	LSTM	NSL KDD, BoT-IoT	99.91, 99.90
[29]	Nil	WILS-TRS	CICIDS 001, UNSW NB 15, NSL KDD	99.30, 99.10, 99.50
[30]	Nil	FSO-LSTM	Real time	98.89
[31]	KPCA	Ensemble DL	SDN-IoT	97
[32]	Nil	CNN-LSTM	UNSW NB 15, X-IIoTID	92.90, 99.80
[33]	Nil	LSTM-AE	CICIDS 2017, CICIDS 2018	99.90, 99.10
[34]	LPPSO	ELSTM-RNN	UNSW NB 15, NSL KDD	97.04, 98.80
		Proposed	Testbed, CICIDS 2017, DS2OS, MQTTset	99.11, 99.85, 99.47, 99.66

V. CONCLUSION

In this work, we have proposed and analyzed an IGWO-enabled AE-LSTM network for anomaly detection in an IoT edge environment. A novel testbed dataset is generated using Raspberry Pi 4 and sensors to train and validate the proposed model. The generated imbalanced data are transformed into balanced data using the RN-SMOTE algorithm. An autoencoder network selects prominent features in the balanced data and train them using an LSTM network for anomaly detection. To further improve the performance and reduce the complexity, we have tuned the parameters of the LSTM network using the proposed IGWO algorithm. Simulation results demonstrate that the proposed technique produces an accuracy of 99.11%, 99.85%, 99.47%, and 99.66% for the testbed, CICIDS 2017, DS2OS, and MQTTset datasets, respectively. We assessed the goodness-of-fit of the developed model using the Wilcoxon signed-rank test. Despite these benefits, the proposed IGWO-AE-LSTM model has certain limitations in terms of higher training time and complexity as compared to traditional ML algorithms and DL algorithms, owing to its sophisticated nature. The future perspective of this study is to incorporate novel lightweight feature selection techniques in the feature selection stage and to utilize an ensemble of DL concepts in the classification stages. Also, we have generated only four IoT attacks using eight Raspberry Pis and eight sensors. The robustness of the dataset can be enhanced by generating more IoT attacks and increasing the number of IoT devices in future studies.

REFERENCES

[1] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A survey on the Internet of Things (IoT) forensics: Challenges, approaches, and open issues," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 1191–1221, 2nd Quart., 2020.

[2] S. Kumar, P. Tiwari, and M. Zymbler, "Internet of Things is a revolutionary approach for future technology enhancement: A review," *J. Big Data*, vol. 6, no. 1, pp. 1–21, Dec. 2019, doi: 10.1186/s40537-019-0268-2.

[3] M. Shirer and C. MacGillivray. (2025). *The Growth in Connected Iot Devices is Expected to Generate 79.4 Zb of Data in 2025, According to a New IDC Forecast*. [Online]. Available: https://www.idc.com/getdoc.jsp

[4] I. Statista. (2015). *Internet of Things (IoT) Connected Devices Installed Base Worldwide From 2015 to 2025 (in Billions)*. [Online]. Available: https://www.statista.com/statistics/471264/iot-number-of-connecteddevicesworldwide/

[5] M. Z. Arshad, H. Rahman, J. Tariq, A. Riaz, A. Imran, A. Yasin, and I. Ihsan, "Digital forensics analysis of IoT nodes using machine learning," *J. Comput. Biomed. Informat.*, vol. 4, no. 1, pp. 1–12, Dec. 2022.

[6] A. Heidari and M. A. Jabraeil Jamali, "Internet of Things intrusion detection systems: A comprehensive review and future directions," *Cluster Comput.*, vol. 26, no. 6, pp. 3753–3780, Dec. 2023.

[7] D. Limon-Cantu and V. Alarcon-Aquino, "Multiresolution dendritic cell algorithm for network anomaly detection," *PeerJ Comput. Sci.*, vol. 7, p. e749, Oct. 2021.

[8] M. J and V. G, "An empirical comparison of machine learning algorithms for attack detection in Internet of Things edge," *ECS Trans.*, vol. 107, no. 1, pp. 2403–2417, Apr. 2022.

[9] X. Qu, L. Yang, K. Guo, L. Ma, M. Sun, M. Ke, and M. Li, "A survey on the development of self-organizing maps for unsupervised intrusion detection," *Mobile Netw. Appl.*, vol. 26, no. 2, pp. 808–829, Apr. 2021.

[10] H. Wang, Z. Cao, and B. Hong, "A network intrusion detection system based on convolutional neural network," *J. Intell. Fuzzy Syst.*, vol. 38, no. 6, pp. 7623–7637, 2020.

[11] G. Andresini, A. Appice, and D. Malerba, "Autoencoder-based deep metric learning for network intrusion detection," *Inf. Sci.*, vol. 569, pp. 706–727, Aug. 2021.

[12] I. Sohn, "Deep belief network based intrusion detection techniques: A survey," *Expert Syst. Appl.*, vol. 167, Apr. 2021, Art. no. 114170.

[13] A. Thakkar and R. Lohiya, "Attack classification of imbalanced intrusion data for IoT network using ensemble learning-based deep neural network," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11888–11895, 2023.

[14] S. Latif, Z. Zou, Z. Idrees, and J. Ahmad, "A novel attack detection scheme for the industrial Internet of Things using a lightweight random neural network," *IEEE Access*, vol. 8, pp. 89337–89350, 2020.

[15] M. Feurer and F. Hutter, "Hyperparameter optimization," in *Automated Machine Learning: Methods, Systems, Challenges* (The Springer Series on Challenges in Machine Learning). Cham, Switzerland: Springer, 2019, pp. 3–33, doi: 10.1007/978-3-030-05318-5_1.

[16] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[17] I. A. Khan, N. Moustafa, D. Pi, W. Haider, B. Li, and A. Jolfaei, "An enhanced multi-stage deep learning framework for detecting malicious activities from autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25469–25478, Dec. 2022.

[18] B. A. Tama, M. Comuzzi, and K.-H. Rhee, "TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system," *IEEE Access*, vol. 7, pp. 94497–94507, 2019.

[19] (2014). *NKDD Dataset for Network-based Intrusion Detection Systems*. [Online]. Available: http://nsl.cs.unb.ca/nsl-kdd/.html

[20] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Nov. 2015, pp. 1–6, doi: 10.1109/MILCIS.2015.7348942.

[21] C. Liu, Z. Gu, and J. Wang, "A hybrid intrusion detection system based on scalable K-means+ random forest and deep learning," *IEEE Access*, vol. 9, pp. 75729–75740, 2021.

[22] A. K. Sahu, S. Sharma, M. Tanveer, and R. Raja, "Internet of Things attack detection using hybrid deep learning model," *Comput. Commun.*, vol. 176, pp. 146–154, Aug. 2021.

[23] Z. E. Huma, S. Latif, J. Ahmad, Z. Idrees, A. Ibrar, Z. Zou, F. Alqahtani, and F. Baothman, "A hybrid deep random neural network for cyberattack detection in the industrial Internet of Things," *IEEE Access*, vol. 9, pp. 55595–55605, 2021.

[24] E. Mushtaq, A. Zameer, M. Umer, and A. A. Abbasi, "A two-stage intrusion detection system with auto-encoder and LSTMs," *Appl. Soft Comput.*, vol. 121, May 2022, Art. no. 108768.

[25] B. Deore and S. Bhosale, "Hybrid optimization enabled robust CNN-LSTM technique for network intrusion detection," *IEEE Access*, vol. 10, pp. 65611–65622, 2022.

[26] S. Roy, J. Li, B.-J. Choi, and Y. Bai, "A lightweight supervised intrusion detection mechanism for IoT networks," *Future Gener. Comput. Syst.*, vol. 127, pp. 276–285, Feb. 2022.

[27] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "CNN-LSTM: Hybrid deep neural network for network intrusion detection system," *IEEE Access*, vol. 10, pp. 99837–99849, 2022.

[28] I. Ullah and Q. H. Mahmoud, "Design and development of RNN anomaly detection model for IoT networks," *IEEE Access*, vol. 10, pp. 62722–62750, 2022.

[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] A. S. Alqahtani, "Retraction note: FSO-LSTM IDS: Hybrid optimized and ensembled deep-learning network-based intrusion detection system for smart networks," *J. Supercomput.*, vol. 78, no. 7, pp. 9438–9455, Feb. 2024.

[31] V. Ravi, R. Chaganti, and M. Alazab, "Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system," *Comput. Electr. Eng.*, vol. 102, Sep. 2022, Art. no. 108156.

[32] 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.

[33] V. Hnamte, H. Nhung-Nguyen, J. Hussain, and Y. Hwa-Kim, "A novel two-stage deep learning model for network intrusion detection: LSTM-AE," *IEEE Access*, vol. 11, pp. 37131–37148, 2023.

[34] 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.

[35] A. V. Hanafi, A. Ghaffari, H. Rezaei, A. Valipour, and B. Arasteh, "Intrusion detection in Internet of Things using improved binary golden jackal optimization algorithm and LSTM," *Cluster Comput.*, Jul. 2023, doi: 10.1007/s10586-023-04102-x.

[36] M. S. Al-kahtani, Z. Mehmood, T. Sadad, I. Zada, G. Ali, and M. ElAffendi, "Intrusion detection in the Internet of Things using fusion of GRU-LSTM deep learning model," *Intell. Autom. Soft Comput.*, vol. 37, no. 2, pp. 2279–2290, 2023.

[37] Y. Li, J. Zhang, Y. Yan, Y. Lei, and C. Yin, "Enhancing network intrusion detection through the application of the dung beetle optimized fusion model," *IEEE Access*, vol. 12, pp. 9483–9496, 2024.

[38] A. K. Mishra, S. Paliwal, and G. Srivastava, "Anomaly detection using deep convolutional generative adversarial networks in the Internet of Things," *ISA Trans.*, vol. 145, pp. 493–504, Feb. 2024.

[39] N. Chander and M. U. Kumar, "Enhanced pelican optimization algorithm with ensemble-based anomaly detection in industrial Internet of Things environment," *Cluster Comput.*, Mar. 2024, doi: 10.1007/s10586-024-04303-y.

[40] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, 2018, pp. 1–11.
[41] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," *IEEE Access*, vol. 8, pp. 165130–165150, 2020.
[42] I. Ullah and Q. H. Mahmoud, "Design and development of a deep learning-based model for anomaly detection in IoT networks," *IEEE Access*, vol. 9, pp. 103906–103926, 2021.
[43] H. Hindy, E. Bayne, M. Bures, R. Atkinson, C. Tachtatzis, and X. Bellekens, "Machine learning based IoT intrusion detection system: An MQTT case study (MQTT-IoT-IDS2020 dataset)," in *Proc. Int. Netw. Conf.*, 2020, pp. 73–84.
[44] R. Pi. (2015). *Raspberry Pi 4 Model B*. [Online]. Available: https://www.raspberrypi.org
[45] H. Siddharthan, T. Deepa, and P. Chandhar, "SENMQTT-SET: An intelligent intrusion detection in IoT-MQTT networks using ensemble multi cascade features," *IEEE Access*, vol. 10, pp. 33095–33110, 2022.
[46] J. Manokaran, G. Vairavel, and J. Vijaya, "A novel set theory rule based hybrid feature selection techniques for efficient anomaly detection system in IoT edge," in *Proc. Int. Conf. Quantum Technol., Commun., Comput., Hardw. Embedded Syst. Secur. (iQ-CCHESS)*, Sep. 2023, pp. 1–6.
[47] J. Manokaran and G. Vairavel, "GIWRF-SMOTE: Gini impurity-based weighted random forest with SMOTE for effective malware attack and anomaly detection in IoT-edge," *Smart Sci.*, vol. 11, no. 2, pp. 276–292, Dec. 2022, doi: 10.1080/23080477.2022.2152933.
[48] A. Arafa, N. El-Fishawy, M. Badawy, and M. Radad, "RN-SMOTE: Reduced noise SMOTE based on DBSCAN for enhancing imbalanced data classification," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 8, pp. 5059–5074, Sep. 2022.
[49] H. D. Nguyen, K. P. Tran, S. Thomassey, and M. Hamad, "Forecasting and anomaly detection approaches using LSTM and LSTM autoencoder techniques with the applications in supply chain management," *Int. J. Inf. Manage.*, vol. 57, Apr. 2021, Art. no. 102282.
[50] S. M. Kasongo, "A deep learning technique for intrusion detection system using a recurrent neural networks based framework," *Comput. Commun.*, vol. 199, pp. 113–125, Feb. 2023.
[51] Y.-C. Wang, Y.-C. Houng, H.-X. Chen, and S.-M. Tseng, "Network anomaly intrusion detection based on deep learning approach," *Sensors*, vol. 23, no. 4, p. 2171, Feb. 2023.
[52] F. Laghrissi, S. Douzi, K. Douzi, and B. Hssina, "Intrusion detection systems using long short-term memory (LSTM)," *J. Big Data*, vol. 8, no. 1, p. 65, Dec. 2021.
[53] T. A. Alamiedy, M. Anbar, Z. N. M. Alqattan, and Q. M. Alzubi, "Anomaly-based intrusion detection system using multi-objective grey wolf optimisation algorithm," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 9, pp. 3735–3756, Sep. 2020.
[54] J.-S. Wang and S.-X. Li, "An improved grey wolf optimizer based on differential evolution and elimination mechanism," *Sci. Rep.*, vol. 9, no. 1, p. 7181, May 2019.
[55] X. Yu, W. Xu, and C. Li, "Opposition-based learning grey wolf optimizer for global optimization," *Knowledge-Based Syst.*, vol. 226, Aug. 2021, Art. no. 107139, doi: 10.1016/j.knosys.2021.107139.
[56] J. Manokaran and G. Vairavel, "IGWO-SoE: Improved grey wolf optimization based stack of ensemble learning algorithm for anomaly detection in Internet of Things edge computing," *IEEE Access*, vol. 11, pp. 106934–106953, 2023.
[57] A. Thakkar and R. Lohiya, "Fusion of statistical importance for feature selection in deep neural network-based intrusion detection system," *Inf. Fusion*, vol. 90, pp. 353–363, Feb. 2023.
[58] M. Barhoush, B. H. Abed-alguni, and N. E. A. Al-qudah, "Improved discrete salp swarm algorithm using exploration and exploitation techniques for feature selection in intrusion detection systems," *J. Supercomput.*, vol. 79, no. 18, pp. 21265–21309, Dec. 2023.

J. MANOKARAN received the bachelor's degree in electronics and communication engineering and the master's degree in applied electronics from Anna University, in 2009 and 2011, respectively. He is currently pursuing the Ph.D. degree with the SRM Institute of Science and Technology, Kattankulathur, Chennai, India. His list of academic achievements includes six publications, three journal articles, and three conference papers. His research interests include the Internet of Things, computational intelligence, machine learning, deep learning, edge computing, cluster computing, and intrusion detection. During his post-graduation, he received the Best Technical Paper Award from the National Level Conference on Computer Intelligent and Application organized by the CARE College of Engineering, Trichy.

G. VAIRAVEL (Senior Member, IEEE) received the bachelor's degree in electronics and communication engineering from Madras University, India, in 2001, and the master's degree in communication systems and the Ph.D. degree in wireless communication from Anna University, Chennai, India, in 2004 and 2014, respectively. He is currently the Dean of Academics and Educational Initiatives with the SRM Institute of Science and Technology, Trichy, India. His research interests include massive MIMO communication systems, antenna design, machine learning, the Internet of Things, and engineering education. He is an active member of the American Society for Engineering Education and the TPACK Special Interest Group. He is a fellow of the Institute of Engineers, India. He is elected as a Member-at-Large of the CDIO International Council. He is serving as an editor, the guest editor for reputed journals, and chaired reputed conferences.