IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 30, 2022, accepted April 12, 2022, date of publication May 18, 2022, date of current version June 17, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3176317

Design and Development of RNN Anomaly Detection Model for IoT Networks

IMTIAZ ULLAH, (Member, IEEE), AND QUSAY H. MAHMOUD, (Senior Member, IEEE)
Department of Electrical, Computer and Software Engineering, Ontario Tech University, Oshawa, ON L1G 0C5, Canada

Corresponding author: Imtiaz Ullah (imtiaz.ullah@ontariotechu.net)

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

ABSTRACT Cybersecurity is important today because of the increasing growth of the Internet of Things (IoT), which has resulted in a variety of attacks on computer systems and networks. Cyber security has become an increasingly difficult issue to manage as various IoT devices and services grow. Malicious traffic identification using deep learning techniques has emerged as a key component of network intrusion detection systems (IDS). Deep learning methods have been a research focus in network intrusion detection. A Recurrent Neural Network (RNN) is useful in a wide range of applications. First, this paper proposes a novel deep learning model for anomaly detection in IoT networks using a recurrent neural network. Long Short Term Memory (LSTM), BiLSTM, and Gated Recurrent Unit (GRU) techniques are used to implement the proposed model for anomaly detection in IoT networks. A Convolutional Neural Network (CNN) can analyze input features without losing important information, making them particularly well suited for feature learning. Next, a hybrid deep learning model was proposed using convolutional and recurrent neural networks. Finally, a lightweight deep learning model for binary classification was proposed using LSTM, BiLSTM, and GRU based approaches. The proposed deep learning models are validated using NSLKDD, BoT-IoT, IoT-NI, IoT-23, MQTT, MQTTset, and IoT-DS2 datasets. Compared to current deep learning implementations, the proposed multiclass and binary classification model achieved high accuracy, precision, recall, and F1 score.

INDEX TERMS Internet of Things, anomaly detection, recurrent neural network, convolutional neural network, LSTM, BiLSTM, GRU.

I. INTRODUCTION

The fast expansion of the Internet has facilitated the development of the Internet of Things. A common element contributing to this development is the ease with which IoT devices are available, affordable, and convenient in everyday lives. Due to the fast advancement of wireless communication technologies, developers have built extremely low cost IoT nodes that support data collection, data analysis, and wireless transmission [1]. IoT is a network of linked physical devices and sensors that enables information sharing via the Internet. IoT networks have become particularly effective in collecting, analyzing, reporting, and predicting information for use in future plans. IoT networks are made up of various technologies, including protocols, software, and sensor elements. An IoT architecture is a collection of multiple components such as sensors, actuators, protocols, cloud services, and layers of an IoT communication network.

Each architecture is divided into several layers that enable network administrators to detect, analyze, and monitor the IoT system's consistency. There is no single agreed upon design for IoT, with numerous designs proposed by various manufacturers. The primary types of architecture are three, four, and five layer structures. A three layer structure, comprised of perception, network, and application layers, is the most basic configuration for IoT implementation. The perception layer is equipped with sensors, actuators, and computational hardware to detect and collect information from the environments. The physical layer handles tasks such as setting a frequency, manipulating the signal, encrypting the signal, and transmitting and receiving data. This layer has several issues, including power consumption, security, and compatibility. IoT devices are connected to other smart objects, network equipment, and services via the network layer. This layer receives data from the perception layer and passes the data to the application for analytics and smart services. The network layer needs to deal with reliability, network capacity, energy usage, and security.

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

The application layer provides user specific services and applications [2], [3].

The application, data processing, network, and perception layers constitute a four layer architecture. The application layer describes all IoT applications and connects end IoT devices to the network. Application layer grants access to different services depending on the information gathered by sensors. The data processing layer receives information from the perception layer and ensures that only data from legitimate users is transmitted by being secure against cyberattacks. The network layer connects the real and virtual worlds by gathering data from sensors and transferring it to other network devices and networks. The perception layer, also known as the sensor layer, identifies and collects data from IoT devices. An IoT network will likely consist of sensors of various types. The perception layer should be able to identify between these diverse sensors and their various operation methods.

A five layer IoT architecture adds the processing and business layers to the three layer model design. The perception and application layers serve the same purpose as the three layered architecture in this architecture. The transport layer sends and receives data from IoT sensors between the perception and processing layers. The processing layer, also known as the middleware layer, manages the storage, analysis, and processing of large data volumes delivered by the transport layer. The business layer oversees the IoT system's general administration, including applications, business models, and user privacy. The IoT network can be a Low Power Wide Area Network (LPWAN), Local and Personal Area Network (LAN / PAN), cellular network, or mesh network [4].

Recently, the Industrial Internet of Things (IIoT) has emerged as the most rapidly developing revolutionary technology, with the ability to digitize and integrate numerous sectors, resulting in significant economic benefits for all relevant stakeholders. IIoT offers significant potential for developing a wide range of industrial applications, but they are also vulnerable to cyberattacks and need higher levels of security. The wide range of sensors in the IIoT networks creates a significant volume of data, which has caught the interest of hackers worldwide. When it comes to protecting IIoT applications from cyberattacks, the intrusion detection system, which monitors network traffic and identifies network behavior, is regarded as one of the most important security measures [5]. As the IoT networks technology continues to advance, cyber attack detection measures are becoming more important in assuring the security of IoT networks. However, with the continuous development of IoT network traffic, standard IDSs are incapable of rapidly and reliably identifying complex and varied IoT network attacks, particularly those using low frequency attacks [6].

IoT devices and cloud computing are among the expanding Internet connected services, making it increasingly difficult to prevent cyber attacks. Cyberattacks have evolved into severe threats to security and privacy, as their influence on IoT devices would result in financial losses and potentially risk human life. Network intrusion detection is important for monitoring and identifying potential threats, events, and breaches. Security systems, such as firewalls and intrusion detection systems, are vulnerable to modern cyber threats since existing techniques are focused on static attack signatures and cannot recognize new attack variants [7].

IoT is a valuable target for cybercriminals because of its significant economic impact and widespread influence on our lives. Cybersecurity has gone to the top of the priority list for IoT infrastructure. Even though cybersecurity has been studied for years, the growing IoT networks and the introduction of innovative threats have rendered conventional measures ineffective. The study conducted by Tsimenidis *et al.* [8] provides a comprehensive assessment of deep learning models that have been developed for IoT intrusion detection. Deep learning has been used for IoT cybersecurity, and models have categorized its unique contributions to establishing efficient IoT intrusion detection systems in a detailed and organized evaluation.

It is becoming more concerning for most service providers as the number of computer networks and Internet threats grows. It has motivated the development and implementation of IDSs to help prevent and mitigate threats caused by network intruders. An important role in detecting network cyberattacks and abnormalities has been performed and continues to be played by intrusion detection systems. Researchers have proposed numerous intrusion detection techniques to counter the threats posed by network intruders. However, most previously proposed intrusion detection systems have high rates of false alarms [9].

The primary aim of this paper is to design and develop an RNN based anomaly detection model for IoT networks. Kernel, bias, and activity regularizers were implemented in RNN models. Layer normalization was utilized in these models, which optimizes and stabilizes the learning process. Layer normalization and activity regularization layers were used to develop novel RNN and CNNRNN based models for multiclass and binary classification. In contrast to batch normalization, the layer normalization method normalizes the activations of the previous layer for each given sample in a batch individually rather than across the entire batch. Activity regularization changes the cost function, which is based on input activities. These models were aided in learning weak features by incorporating an activity regularization layer into their design. A model is vulnerable to overfitting, and significant changes will be required during training to avoid overfitting. Overfitting of the RNN models was assessed using the dropout layer, early stopping, and 5-fold cross-validation techniques.

First, class weights are utilized to address class imbalances in datasets during the training phase. The weights assigned to classes were determined by the number of instances of each class, so a minority class with a small number of instances will receive a high weight. Next, new synthetic samples were created using the borderline SMOTE algorithm.

The random state controls the algorithm's randomization. The borderline SMOTE is being used to ensure the training set is balanced. Features extracted from pcap files are generalized features applicable to any IoT network. The NSLKDD, BoT-IoT, IoT Network Intrusion, IoT-23, MQTT, MQTTset, and IoT-DS2 datasets evaluated the proposed multiclass and binary classification models. The contributions of this paper are:

- Design of an anomaly detection model for IoT networks using a recurrent neural network.
- Design of an anomaly detection model for IoT networks using convolutional and recurrent neural networks.
- A lightweight anomaly detection model for IoT networks using a recurrent neural network.
- Performance improvements of multiclass and binary classification models.

The remainder of this paper is organized as follows: Section II provides a discussion of the related work. Section III explains the design and development of the recurrent and convolutional neural networks models for anomaly detection in IoT networks. Section IV discusses the data collecting process for evaluating the proposed models. Section V presents the evaluation results, followed by discussion and comparison of results in Section VI. Finally, Section VII concludes the paper and offers recommendations for future research.

II. RELATED WORK

Millions of IoT devices are embedded in smart cities, enabling important applications such as smart homes, autonomous vehicles, and communications. The smart city is based on millions of heterogeneous sensors that do not support traditional security frameworks. Several manufacturers use inadequate protection mechanisms for their devices and fail to upgrade their firmware in response to recently discovered operational security breaches. To achieve comprehensive management of sensor operating systems while maintaining perfect security, smart cities need a common architecture that combines soft computing and deep learning [10].

While many machine learning methods have been used to identify anomaly based invasions, relatively few attempts have been made to classify recurrent neural networks. The IoT devices have evolved fast in recent years, and cyber cyberattacks on IoT devices are becoming common. It has become more necessary to have an effective approach for identifying malicious attacks in the IoT domain to reduce security threats on IoT devices. Alkahtani *et al.* [11] developed a hybrid deep learning approach based on CNN and LSTM for detecting botnet attacks on nine IoT devices. Their methodology proved effective in identifying botnet assaults from various IoT devices with an average accuracy of 90%.

Autonomous Vehicles (AVs) are prone to safety and security issues, risking human life. The Internet of Vehicles (IoVs) is a network of manually operated vehicles connected to the Internet. If cyber attackers get access to these vehicles, they might be utilized for malicious purposes. Khan *et al.* [12] have developed a multistage intrusion detection system to identify intrusions in AVs and the Internet of Vehicles (IoVs) while minimizing the number of false alarms. The proposed framework uses a BiLSTM architecture to detect intrusions from AVs network gateways and communication networks. Additionally, the suggested system can detect zero day outbreaks in networks of IoVs.

Smart home network IoT devices are susceptible to sophisticated botnet assaults. Popoola *et al.* [13] examine the performance of RNNs in properly classifying network traffic samples belonging to minority groups in severely unbalanced network traffic data. To learn hierarchical representations of highly unbalanced network traffic data with different degrees of abstraction, many layers of RNN are stacked. To effectively capture the classifying properties of severely unbalanced network traffic samples, the stacked RNN model was used instead of the RNN model. The SRNN model also showed excellent generalization abilities when recognizing network traffic samples from minority groups.

The number of computer controlled automobiles is increasing at an alarming rate worldwide. Even though this improves the driving experience, it introduces a new security vulnerability in the automobile business. Desta *et al.* [14] suggested an LSTM based intrusion detection system built on arbitration ID sequences. They were only able to get an accuracy of 60% using this strategy. Applying this finding to a real car would result in a large number of false negatives; they designed a second strategy that utilizes log loss as an anomalous indicator.

Malicious traffic identification using deep learning techniques has become a crucial aspect of network intrusion detection research. Most effective IDS require packets to be classified into specific flows before analysis, which causes processing delays. In order to identify malicious traffic at the packet level, Wang *et al.* [15] offer a deep learning strategy, employing hierarchical networks, which can learn the properties of communication using basic data packets. They also explored how data balance affects classification performance and time efficiency between the LSTM and GRU models.

Hao *et al.* [16] utilize the encoder to automatically process and analyze network packets to get properties that appropriately reflect the network packets. The variant gated recurrent units dynamically understand data packet content and header attributes to significantly increase the IDS detection rate. The experimental findings from the ISCX2012 dataset indicate that intrusion detection using the proposed variant gated recurrent units provides a greater level of accuracy and detection rate. Using binary weights and activation functions, their proposed model provides a greater representation of the data than the original raw data, which helps to minimize the amount of memory and access time. An overview of recent developments in deep learning for intrusion detection is presented in Table 1. In Table 1, DR mean detection rate, Acc represents the accuracy, F1

TABLE 1. An overview of the related work in deep learning for intrusion detection.

Article	Year	Model	Dataset	Classification	Performance
Wang *et al.* [17]	2017	CNN	KDD	Binary	DR=97.66
Wang *et al.* [18]	2017	CNN	ISCX2012	Multiclass	Acc=99.69
Yin *et al.* [19]	2017	RNN	NSLKDD	Multiclass	Acc=83.28
Meng *et al.* [20]	2017	LSTM	NSLKDD	Binary	Acc=98.85
Diro *et al.* [21]	2018	LSTM	AWID	Binary	Acc =98.22
Roy *et al.* [22]	2018	BiLSTM	UNSW-NB15	Binary	Acc=95.71
Yang *et al.* [23]	2018	DBN	NSLKDD	Multiclass	Acc=82.08
Xu *et al.* [24]	2018	BGRU+MLP	KDD99	Multiclass	Acc=99.84
Xu *et al.* [24]	2018	BGRU+MLP	NSLKDD	Multiclass	Acc=99.24
Wu *et al.* [25]	2018	RNN	NSLKDD	Multiclass	Acc=81.29
Naseer *et al.* [26]	2018	LSTM	NSLKDD	Multiclass	Acc=89.00
Ding *et al.* [27]	2018	LSTM	NSLKDD	Multiclass	Acc=73.18
Ding *et al.* [27]	2018	CNN	NSLKDD	Multiclass	Acc=80.13
Hwang *et al.* [28]	2019	LSTM	ISCX2012	Binary	Acc =99.99
Moreton *et al.* [29]	2019	LSTM, GRU	Personal Dataset	Multiclass	Acc =96.08
Nguyen *et al.* [30]	2019	GRU	Personal Dataset	Multiclass	Acc=95.60
Li *et al.* [31]	2019	GRU	Personal Dataset	Multiclass	F1 =80.30
Ferrag *et al.* [32]	2019	RNN	BoT-IoT	Multiclass	Acc =98.20
Arivud *et al.* [33]	2019	CNN	NSLKDD	Binary	Acc =99.67
Chouhan *et al.* [34]	2019	CNN	NSLKDD	Multiclass	Acc=89.41
Vinayakumar *et al.* [35]	2019	DNN	KDD99	Multiclass	Acc=92.90
Vinayakumar *et al.* [35]	2019	DNN	KDD99	Binary	Acc=93.00
Faker *et al.* [36]	2019	DNN	UNSW-NB15	Binary	Acc=99.19
Faker *et al.* [36]	2019	DNN	UNSW-NB15	Multiclass	Acc=97.04
Faker *et al.* [36]	2019	DNN	CICIDS2017	Binary	Acc=97.73
Liu *et al.* [37]	2019	CNN-LSTM	NSLKDD	Binary	Acc=98.90
Anani *et al.* [38]	2019	LSTM	KDD99	Binary	Acc=99.43
Anani *et al.* [38]	2019	GRU	KDD99	Binary	Acc=99.06
Anani *et al.* [38]	2019	Bi-LSTM	KDD99	Binary	Acc=82.20
Li *et al.* [39]	2019	GRU	NSLKDD	Binary	Acc=82.87
Sokolov *et al.* [40]	2019	GRU	Gas Pipeline	Binary	Acc=91.70
Hao *et al.* [16]	2019	GRU	ISCX 2012	Binary	Acc=99.90
Ge *et al.* [41]	2019	DNN	BoT-IoT	Multiclass	Acc=98.09
Kim *et al.* [42]	2020	CNN-LSTM	CISC-2010	Binary	Acc=91.54
Kim *et al.* [42]	2020	CNN-LSTM	CICISC-2017	Binary	Acc=93.00
Roopak *et al.* [43]	2020	CNN-LSTM	CICIDS2017	Binary	Acc=99.03
Jiang *et al.* [44]	2020	LSTM	KDD99	Binary	Acc=98.94
Malik *et al.* [45]	2020	LSTM-CNN	CICIDS2017	Multiclass	Acc=98.60
Susilo *et al.* [46]	2020	CNN	BoT-IoT	Binary	Acc =91.00
Ge *et al.* [47]	2020	DNN	BoT-IoT	Multiclass	Acc =99.79
Ferrag *et al.* [48]	2020	RNN	BoT-IoT	Multiclass	Acc =98.37
Wang *et al.* [15]	2020	CNN, GRU	Multiple	Binary	Acc=99.42
Hassan *et al.* [49]	2020	CNN–WDLSTM	UNSW-NB15	Binary	Acc=97.17
Aldhaheri *et al.* [50]	2020	SNN	BoT-IoT	Multiclass	Acc=98.73
Ferrag *et al.* [48]	2020	RNN	BoT-IoT	Binary	Acc=98.31
Liu *et al.* [51]	2021	DSSTE+LSTM	NSLKDD	Multiclass	Acc=81.78
Ashraf *et al.* [52]	2021	LSTM Autoencoder	UNSW-NB15	Binary	Acc=98.00
Borisenko *et al.* [53]	2021	LSTM	CIC-IDS2018	Multiclass	Acc=94.00
Hai *et al.* [54]	2021	LSTM	CICIDS2017	Binary	Acc=99.55
Pooja *et al.* [55]	2021	BiLSTM	KDD	Binary	Acc =99.70
Jia *et al.* [56]	2021	IE-DBN	KDD	Multiclass	Acc =98.12
Jia *et al.* [56]	2021	IE-DBN	NSLKDD	Multiclass	Acc =98.79
Biswas *et al.* [57]	2021	LSTM-GRU	BoT-IoT	Binary	Acc =99.76
Biswas *et al.* [57]	2021	LSTM-GRU	NSL	Binary	Acc =99.14
Laghrissi *et al.* [58]	2021	LSTM	KDD99	Binary	Acc=98.88
Imrana *et al.* [9]	2021	BiLSTM	NSLKDD	Binary	Acc=94.26
Imrana *et al.* [9]	2021	BiLSTM	NSLKDD	Multiclass	Acc= 91.36
ElSayed *et al.* [59]	2021	CNN	InSDN	Binary	Acc = ~97.50
ElSayed *et al.* [59]	2021	CNN	InSDN	Multiclass	Acc = ~97.50
Joshi *et al.* [60]	2021	ANN	CTU-13	Binary	Acc = 99.94
Sethi *et al.* [61]	2021	Reinforcement	NSLKDD	Multiclass	Pr=96.50
Khan *et al.* [12]	2021	LSTM	UNSWNB-15	Binary	Acc =98.88
Alyasiri *et al.* [62]	2021	GE	MQTT	Binary	Acc=97.94
Hussain *et al.* [63]	2021	DT	MQTTset	Binary	Acc=99.47
Vaccari *et al.* [64]	2021	RF	MQTTset	Binary	Acc=99.68

represents the F1 score, and Pr represents precision. When data is unavailable, it is denoted by a "-".

Recent years have seen a tremendous influence on industrial production because of the fast development and widespread use of new technologies, resulting in smart manufacturing (SM). However, industrial systems based on the IoT are currently one of the most targeted sectors for various cyberattacks. An anomaly detection approach is proposed by Huong *et al.* [65] to identify industrial control systems cyberattacks. The framework outperforms existing time series data detection solutions in terms of detection performance. In the Industrial IoT (IIoT), a vast quantity of data processing takes place in the cloud and at the edge to perform various types of analytics. IIoT routing attacks can be detected using Nayak *et al.* [66] deep learning based routing attack detection technique. The proposed solution uses parallel learning and detection to facilitate deep learning on IIoT devices with limited processing power. The parallel model output is tested in an IIoT network to compare the performance of distributed and centralized threat detection in an RPL network. Training time is significantly reduced when a parallel GAN model is used.

Although various relevant research has utilized deep learning for NIDS, most of these techniques fail to consider the impact of overfitting when deep learning algorithms are implemented. Therefore, the anomaly detection system's resilience may be compromised, making detecting zero day cyberattacks less effective. Convolutional neural networks and a novel regularizer technique were proposed by Elsayed *et al.* [59] to categorize network flow traffic into normal and attack categories. Additionally, they propose a lightweight CNN model with fewer features without sacrificing model performance significantly. Advanced information and communication technologies have facilitated the dissemination of a large quantity of information, which continues to grow day by day through the Internet and the creation of new added value via Internet based activities. Increasing numbers of diverse connecting points with high computing capability have expanded cyber security concerns. Biswas *et al.* [57] have suggested a new deep learning strategy to distinguish malicious botnet traffic from normal traffic.

The widespread availability of Internet services around the globe has presented a significant challenge to service providers in terms of protecting their systems, particularly against new breaches and threats. The GRU is the most effective model for botnet detection; however, it is computationally costly, yet it can handle large amounts of data and identify sequences efficiently. Deep learning approaches based on the LSTM algorithm were developed by Laghrissi *et al.* [58] for the purpose of detecting attacks. They utilized PCA and mutual information to reduce the dimensionality of the data and select the best features. These techniques were also evaluated in terms of performance and processing time.

The battery performance of IoT devices is a major concern, as the devices consume a significant amount of energy when connected. IoT devices might also contain important network data, raising major privacy and security risks. Botnet attacks are significant threats to IoT smart devices. Ashraf *et al.* [67] secure IoT networks against botnet cyberattacks using statistical learning for botnet detection. Louk *et al.* [68] address a gap in the literature by showing the importance of ensemble models for detecting potential attacks in a cyber physical power system. They balanced the dataset by employing oversampling and undersampling techniques. Oversampling and undersampling were beneficial in a boosting ensemble model but were ineffective in a bagging ensemble model. Ensemble learners have outperformed single learners in a wide variety of applications, including the cybersecurity area. However, the majority of previously published works continue to produce unsatisfactory outcomes due to insufficient ensemble design. Nkenyereye *et al.* [69] demonstrate the efficacy of stacking ensemble models for anomaly detection, where a deep neural network is utilized as a basic learner model. The suggested model's effectiveness and DNN model are experimentally compared using a variety of performance criteria.

A two-level hybrid anomalous activity detection model has been proposed to detect intrusions in IoT networks, which detects abnormal activity at level1 and analyses the identified anomalous activity at level2 [70]. The level-2 model selects relevant features using Recursive Feature Elimination (RFE), oversampling using Synthetic Minority Over Sampling Technique (SMOTE) and cleaning the data using Edited Nearest Neighbors (ENN). A three-layer system was proposed to detect intrusions in a smart grid system [71]. The proposed structure includes an IDS in each Home Area Network (HAN) and Neighborhood Area Network (NAN) and many IDS sensors in the WAN. In modeling anomaly based intrusion detection systems, feature selection is important. A filter based feature selection methodology was proposed for anomaly based intrusion detection systems that leverage information gain by considering each feature's consistency, dependency, content, and distance [72]. Additionally, using an industrial control system dataset, the suggested model for feature selection was evaluated for anomaly detection in SCADA networks [73].

It is challenging to extract valuable information from network traffic to identify possible anomalies. Many different network flow features were investigated to address this challenge [74], [75]. A feed forward neural network technique for identifying anomalous activity in IoT networks based on flow and control flags features has been presented [76]. The model was assessed for multiclass and binary classification using a variety of IoT network intrusion datasets. Using conditional GANs to create realistic distributions for a given feature set, a framework for identifying anomalies in IoT networks was proposed, which overcomes data imbalance by using conditional GANs to detect abnormalities in IoT networks [77].

Since IoT devices are increasingly being used in critical infrastructures and cyber physical systems, there has been a significant increase in research efforts to develop efficient defenses against cyber attacks. IoT networks can be protected from a wide range of cyberattacks using a framework called Boost Defense, developed by Al-Haija *et al.* [78]. A strong classifier for identifying and categorizing cyberattacks in IoT networks was built using AdaBoost, decision trees, and substantial data engineering approaches. Anomaly NIDS are lightweight and versatile to construct profiles for normal and malicious behavior using a variety of ways. Al-Haija *et al.* [79] used machine learning approaches to design and evaluate an anomaly based IoT NIDS. It was modeled as supervised multiclass learning, where a classification function was developed to map a collection of labels to ten classes.

III. PROPOSED MODEL

A. RECURRENT NEURAL NETWORK

Neural networks have the ability to improve many aspects of our daily lives. An artificial neural network with a sequential information structure is known as a Recurrent Neural Network (RNN). They are referred to as recurrent because they execute the same function on each sequence element, with the outcome depending on prior calculations. RNNs are loop based networks that enable data preservation. Long Short Term Memory (LSTM) network is a kind of RNN that can learn long term dependencies. Hochreiter *et al.* [80] introduce the LSTM network. The LSTM network performed very well across a broad range of issues and is now extensively utilized. LSTM is designed to prevent long term dependence [81]. Each recurrent neural network comprises a chain of repeating neural network modules. A recurrent neural network includes loops, enabling information to be retained in the network. In Fig.1, a simple recurrent neural network with loops is shown. The neural network in Fig.1, A examines the input x_t and then generates output h_t. A loop enables data to be transferred from one network phase to the next. LSTM is expressly intended to prevent the issue of long term dependence. Each recurrent neural network comprises a chain of repeating neural network modules. Table 2 shows a list of symbols used to help understand the various concepts presented in the following sections.

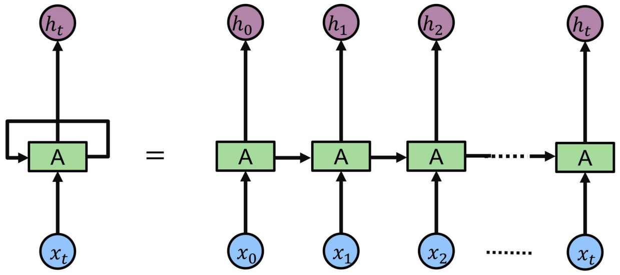


FIGURE 1. Simple recurrent neural network with loops characteristic.

TABLE 2. List of symbols.

Symbol	Meaning
f_t	*fortgot gate*
i_t	*input gate*
o_t	*output gate*
σ	*sigmoid function*
$tanh$	*tanh function*
w_x	*weight for respective gate(x)neurons*
h_{t-1}	*output of prvious LSTM block*
x_t	*input at current timestamp*
b_x	*biases for the repective gate(x)*
C_{t-1}	*LSTM previous memory content*
$\tilde{C}_t$	*LSTM current memory content*
C_t	*LSTM new memory content*
r_t	*GRU reset gate*
z_t	*GRU updated gate*
$\tilde{h}_t$	*GRU current memory content*
h_t	*LSTM or GRU final memory content at current time step*

The first stage of LSTM is to determine which information from the cell state will be discarded. A sigmoid layer, known as the "forget gate layer," makes this determination. The sigmoid layer examines the values in h_{t-1} and x_t and returns a value between 0 and 1 for each value in the cell state C_{t-1}. A 1 indicates "completely retain," whereas a 0 indicates "entirely discard." The forget layer functioning of the LSTM is shown in Fig. 2, and the operation of forgets gate layer is represented by (1). It is necessary to determine what additional information will be stored in the cell state due to the input gate layer decision. The input gate layer, also known as the sigmoid layer, determines the values to update. Equation (2) represents the operation of the input gate layer, and the input gate layer functioning of the LSTM is shown in Fig. 3. A tanh layer generates a vector of potential nominee values, $\tilde{C}_t$ that may be included in the state. A tanh layer operation is represented by (3). Equation (4) updates the cell state C_{t-1} into the new C_t, and (4) function is resented in Fig. 4.

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

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{2}$$

$$\tilde{C}_t = tanh(W_C.[h_{t-1}, x_t] + b_C) \tag{3}$$

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \tag{4}$$

The last step is to determine the output. The output result will be based on the current state of the cell, but it will be a simplified form of it. First, a sigmoid layer is implemented that sets the output of the cell state based on the values received. Equation (5) represents this operation. As shown in (6), the system implements a cell state to the tanh function and then multiplies it by the output of the sigmoid gate. This process ensures that only the values that have been selected to return are sent. The output h_t operation is presented in (6)

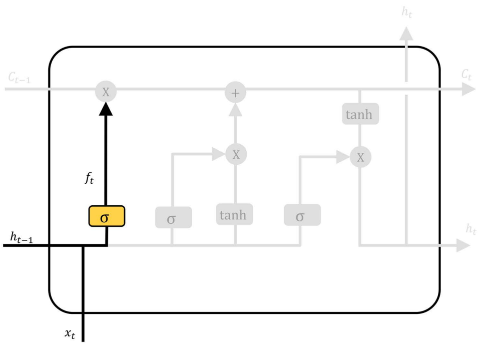


FIGURE 2. LSTM forget layer operation.

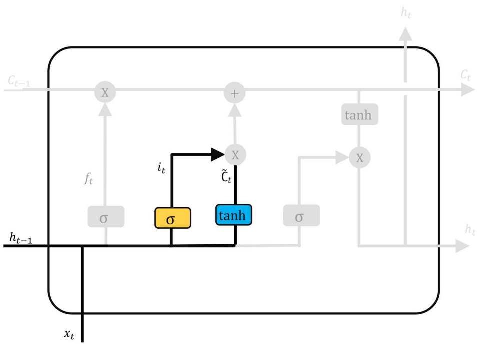


FIGURE 3. LSTM input gate layer operation.

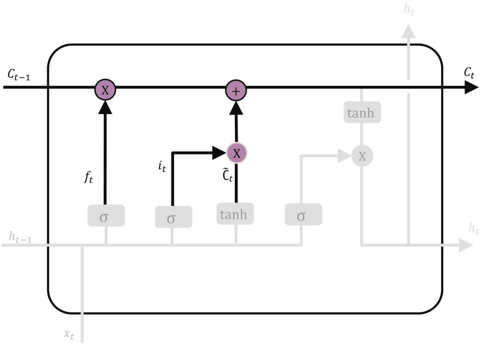


FIGURE 4. LSTM cell state operation.

and Fig. 5.

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t \times \tanh(C_t) \tag{6}$$

Cho *et al.* [82] developed a slight variant of LSTM called Gated Recurrent Unit (GRU). The GRU has two gates, while the LSTM has three gates. The LSTM input and forget gates are combined into an ''update gate'' in GRU. The GRU eliminated the cell state and transferred information via the hidden state. The update gate works the same way as the LSTM forget and input gates. It determines which data should be discarded and which should be included. The reset gate determines how much previous information should be erased from the memory. GRU utilizes fewer tensor operations than LSTM, and as a result, it can be trained faster than the LSTM model. A single GRU cell operation is presented in Fig. 6. Equation (7) represents the reset gate operation, (8) describes the update gate operation, (9) shows the current memory state of the GRU, and (10) represents the final memory state of the GRU.

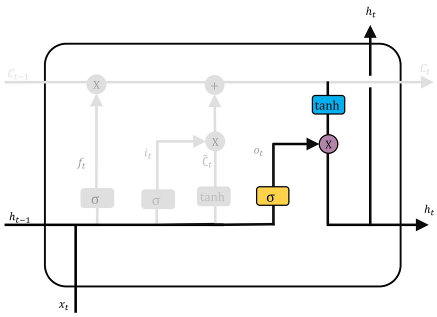


FIGURE 5. LSTM output gate operation.

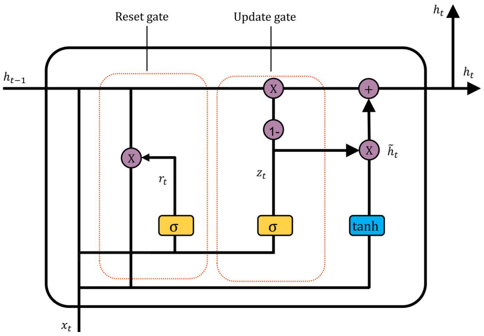


FIGURE 6. GRU cell operation.

$$r_t = \sigma(W_r.[h_{t-1}, x_t]) \tag{7}$$

$$z_t = \sigma(W_z.[h_{t-1}, x_t]) \tag{8}$$

$$\tilde{h}_t = tanh(W.[r_t \times h_{t-1}, x_t]) \tag{9}$$

$$h_t = (1-z_t) \times h_{t-1} + z_t \times \tilde{h}_t \tag{10}$$

BiLSTM is an extension of conventional LSTM that enhances model performance on sequence classification tasks by learning in both directions simultaneously. Because they train two LSTM on the input sequence instead of one, BiLSTM is useful when all timesteps in the input sequence are accessible. BiLSTM considers forward and backward activation to calculate the output.

B. PROPOSED MODEL

Deep learning techniques are gaining popularity due to their ability to detect computer network threats and abnormalities in various applications. A recurrent neural network has shown to be effective in multiple areas. Due to its better capability, this article presents a model based on a recurrent neural network. This paper design and develop LSTM, BiLSTM, and GRU models for anomaly detection in IoT networks. The model consists of an input layer, output layer, and four recurrent, activation, normalization, activity regularization, and dropout layers. Overfitting is a significant concern for deep learning models. Kernel, bias, and activity regularizers were used at the LSTM, BiLSTM, and GRU layers to reduce the possibility of overfitting. The kernel regularizer imposes a penalty on the kernel of the layer; the bias regularizer enforces a penalty on the bias of the layer while the output of the layer is penalized by the activity regularizer. The regularizers uses the value of l1-l2 to compute the value for kernel, bias, and activity regularization.

In the proposed model, the activation layer adopted the LeakyReLU activation function. Next, layer normalization is applied, which typically accelerates and stabilizes the learning process by decreasing error rates. A neural network can be encouraged to learn weak features by using an activity regularization layer. The activity regularization support l1, l2, and l1-l2 regularization techniques. The activity regularization layer updates the input activity, which is dependent on a cost function. A recurrent neural network is prone to overfitting and will need significant adjustments to the training dataset to avoid it. By ignoring certain neurons during the training period, a dropout layer mitigates the risk of overfitting. Neuron weights in a neural network settle into their environment as a network learns. Four recurrent, activation, normalization, activity regularization, and dropout layers were used across LSTM, BiLSTM, and GRU models. A dense layer with 512 neurons and a LeakyReLU activation function is utilized before the output layer. The output layer is the last layer of the model, and the number of neurons in this layer is dependent on the number of classes in the dataset.

A recurrent neural network was used to design an anomaly detection model for IoT networks. Same structure was used to construct LSTM, BiLSTM, and GRU models. Fig. 7 shows a layered view of the proposed recurrent neural network LSTM, BiLSTM, and GRU models. Table 3 shows the proposed LSTM, BiLSTM, and GRU models parameters and hyperparameters, including one input layer, four RNN layers, four activation layers, four normalization layers, four activity regularization layers, four dropout layers, one dense layer, and one output layer. The input layer receives network traffic flow with 64 features. First, a 64 × 1 input vector is created to fit the 64 best features selected by the feature selection method [83]. The LSTM model uses 512 units at each LSTM layer. The LSTM model also uses kernel, bias, and activity regularizers at the LSTM layer. These regularizers use l1-l2 function for penalties. The activation layer uses the LeakyReLU activation function, an alternative to Rectified Linear Unit (ReLU) implementation. It does not contain zero-slope sections; LeakyReLU solves the ''dying ReLU'' issue. The LeakyReLU activation function accelerates the rate of learning significantly. It's been demonstrated that having the ''mean activation'' near to 0 speeds up the training process. In contrast to ReLU, LeakyReLU is more ''balanced''; as a result, it can learn more quickly [84]. A model normalization layer may usually aid in accelerating and stabilizing the learning process by reducing error rates. The LSTM model training and validation details are presented in algorithm 1. BiLSTM and GRU models were created using the same parameters and hyperparameters.

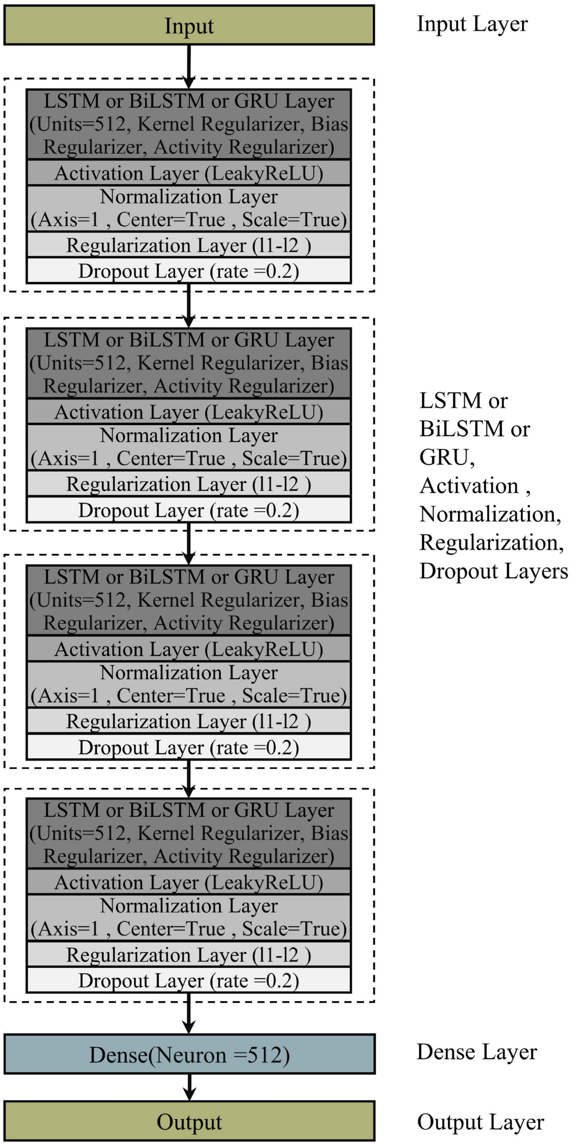

Input
Input Layer
LSTM or BiLSTM or GRU Layer (Units=512, Kernel Regularizer, Bias Regularizer, Activity Regularizer)
Activation Layer (LeakyReLU)
Normalization Layer (Axis=1 , Center=True , Scale=True)
Regularization Layer (l1-l2)
Dropout Layer (rate =0.2)
LSTM or BiLSTM or GRU, Activation , Normalization, Regularization, Dropout Layers
Dense(Neuron =512)
Dense Layer
Output
Output Layer

FIGURE 7. Proposed LSTM, BiLSTM, and GRU models layer's view, parameters, and hyperparameters for multiclass classification.

Layer normalization has the potential to stabilize the hidden state dynamics of a recurrent neural network. To avoid introducing additional dependencies across training instances, unlike batch normalization, layer normalization calculates normalization statistics from the inputs neurons of the hidden layer [85]. Regularization and dropout layers were utilized to reduce the likelihood of overfitting. The activity regularization layer makes changes to the input activity that is dependent on the cost function. l1-l2 factors serve as a baseline for the activity regularization layer functionality. The dropout layer improves overfitting prevention by changing some input units to 0 at a frequency equal to the dropout rate during the training process. This implies that their impact on downstream neuron activity is eliminated temporally on the

Algorithm 1 LSTM Model

```
Input: X, y ← Network Flows
Output: metrics
Set LSTM, activation, normalization, regularization, and
   dropout layers parameters to define the LSTM model (LM)
Model ←LM
Initialize batch size, optimizer, learning rate, epochs
for epochs= 1 to n do
   while early stopping criteria= false
       train Model
       validate Model
       monitor='loss'
       adjust loss function using sparse categorical
       cross entropy
   end while
end for
Evaluate Model
Perform prediction using Model
Calculate metrics
return metrics
```

forward pass, and any weight changes are not transferred to the cell on the backward trip.

BiLSTM is a sequence processing model with two LSTMs: one processing input in the forward direction and the other LSTM processing in the backward direction. The BiLSTM model provides higher predictions than the normal LSTM model due to the extra data training. Long training time is needed for the BiLSTM model. The proposed BiLSTM model consists of an input layer, four BiLSTM layers, four activation layers, four normalization layers, four activity regularization layers, four dropout layers, one dense layer, and one output layer, as shown in Table 3.

The GRU was created to address the vanishing gradient issue inherent in the conventional recurrent neural network. GRU may also be seen as a variant of the LSTM since they are both constructed similarly and, in certain instances, provide equally good outcomes. GRU uses update and reset gates to overcome the vanishing gradient issue of a conventional RNN. To put it another way, two vectors determine what information should be sent to the output device. The proposed GRU model also consists of an input layer, four GRU layers, four activation layers, four normalization layers, four activity regularization layers, four dropout layers, one dense layer, and one output layer. Table 3 summarizes the proposed recurrent neural network model.

A Convolutional Neural Network (CNN) is an algorithm used in deep learning that takes an input picture, assigns significance to distinct image elements, and can distinguish between different image elements. A convolutional neural network requires significantly less preprocessing than other deep learning classification methods. Recent research has shown that a convolutional neural network can produce outstanding speech recognition and image identification results. The advantages of a convolutional neural network can be fully revealed if intrusion detection problems are transformed into image recognition problems. A convolutional neural network can effectively capture spatial and temporal correlations related to intrusion detection. A smaller number of parameters and reusable weights make this design better suited to the problem of intrusion detection than other deep learning approaches [86].

TABLE 3. LSTM, BiLSTM and GRU models parameters and hyperparameters for multiclass classification.

Layer	Layer Name	Layer	Configuration
Input	Input Layer	1	64 Input features
Hidden layers	LSTM or BiLSTM or GRU	4	Units=512 Kernel regularizer, Bias regularizer, Activity regularizer
	Activation	4	LeakyReLU(alpha=0.2)
	Layer Normalization	4	Axis=1, Center=True, Scale=True
	Regularization	4	l1= 0.0001, l2= 0.0001
	Dropout	4	Dropout rate =0.2

Classification	Dense	1	Neuron =512,
			Activation=ReLU

Output	Output	1	The number of neurons is equal to the number of classes in the dataset, Activation=SoftMax
Hyperparameters	Early Stopping (monitor=loss, patience=5, verbose = 1), optimizers= Adam, Loss function= sparse categorical cross entropy, Learning rate=0.001, Batch size=120, epochs= 100 to 500.		

Convolutional and recurrent neural networks were used to design a model for anomaly detection in IoT networks. Convolutional neural network learns input features without losing essential information, making them ideal for prediction. Figure 8 shows a layered view of the proposed convolutional neural network based LSTM, BiLSTM, and GRU models. Table 4 shows the parameters and hyperparameters of a convolutional neural network based LSTM, BiLSTM, and GRU models for multiclass classification. The convolutional layer was created using a CNN1D model. Convolution is computed in 1D using temporal access, with the kernel moving in just one direction. CNN1D requires two dimensional input, and output data is regularly used to process time series data. First, an input vector 64 × 1 is created to correspond to the 64 best features selected by the feature selection method. The convolutional layer is combined with the activation, normalization, regularization, and dropout layers. The activation layer uses LeakyReLU activation function with an alpha value of 0.2. The layer normalization uses a feature axis for normalization. The activity regularization layer uses l1-l2 functions, and the dropout layer randomly drops neurons to reduce the chance of overfitting. A pooling layer reduces the dimension of a complex feature and the computational resources required to analyze the data. The model employs an average pooling layer.

The activation, normalization, regularization, and dropout layers are combined with two LSTM, BiLSTM, or GRU

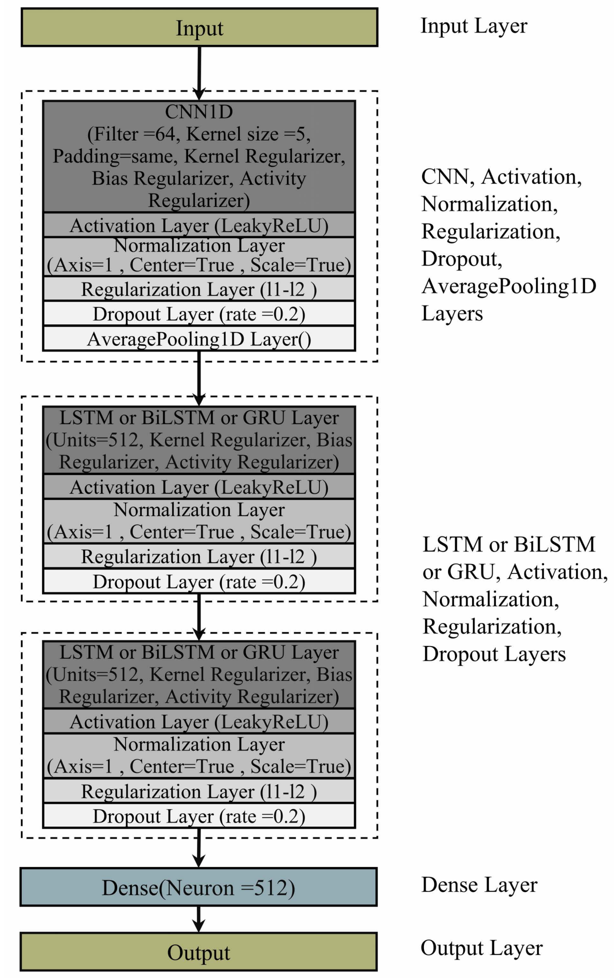

Input
Input Layer
CNN1D
(Filter =64, Kernel size =5, Padding=same, Kernel Regularizer, Bias Regularizer, Activity Regularizer)
Activation Layer (LeakyReLU)
Normalization Layer
(Axis=1 , Center=True , Scale=True)
Regularization Layer (l1-l2)
Dropout Layer (rate =0.2)
AveragePooling1D Layer()
CNN, Activation, Normalization, Regularization, Dropout, AveragePooling1D Layers
LSTM or BiLSTM or GRU Layer
(Units=512, Kernel Regularizer, Bias Regularizer, Activity Regularizer)
Activation Layer (LeakyReLU)
Normalization Layer
(Axis=1 , Center=True , Scale=True)
Regularization Layer (l1-l2)
Dropout Layer (rate =0.2)
LSTM or BiLSTM or GRU Layer
(Units=512, Kernel Regularizer, Bias Regularizer, Activity Regularizer)
Activation Layer (LeakyReLU)
Normalization Layer
(Axis=1 , Center=True , Scale=True)
Regularization Layer (l1-l2)
Dropout Layer (rate =0.2)
LSTM or BiLSTM or GRU, Activation, Normalization, Regularization, Dropout Layers
Dense(Neuron =512)
Dense Layer
Output
Output Layer

FIGURE 8. Proposed convolutional neural network based LSTM, BiLSTM, and GRU models layer's view, parameters, and hyperparameters for multiclass classification.

TABLE 4. Convolutional neural network based LSTM, BiLSTM and GRU models parameters and hyperparameters for multiclass classification.

Layer	Layer Name	Layer	Configuration
Input	Input Layer	1	64 Input features
Hidden layers	CNN1D	1	Filter =64, Kernel size =5, Padding=same, Kernel regularizer, Bias regularizer, Activity regularizer
	LSTM or BiLSTM or GRU	2	Units=512, Kernel regularizer, Bias regularizer, Activity regularizer
	Activation	3	LeakyReLU(alpha=0.2)
	Layer Normalization	3	Axis=1, Center=True, Scale=True
	Regularization	3	l1= 0.0001, l2= 0.0001
	Dropout	3	Dropout rate =0.2

Classification	Dense	1	Neuron =512,
			Activation=ReLU

Output	Output	1	The number of neurons is equal to the number of classes in the dataset, Activation=SoftMax
Hyperparameters	Early Stopping (monitor=loss, patience=5, verbose = 1), optimizers= Adam, loss function= sparse categorical cross entropy, Learning rate=0.001, Batch size=120, epochs= 100 to 500.		

layers. The dense layer receives input from the final dropout layer, and the output of the dense layer is transferred to the output layer. The dense layer uses 512 neurons and LeakyReLU activation function. The number of neurons in the output layer is determined by the number of classes in the dataset. The output layer of the proposed model has four, five, six, ten, and nineteen neurons. The following benefits accrue from the initial model interface being a convolutional neural network: The spatial and temporal correlations associated with an anomaly detection problem can be captured effectively by a convolutional neural network when the optimal filters are used and with fewer parameters and reusable weights, the architecture is more suited to fit the anomaly detection data. The pooling layer lowers computing power by reducing the dimension of the features.

A lightweight binary classification model utilizing a single RNN layer was proposed. A layered view of the binary classification model is presented in Fig. 9. The model consist of an input layer, and the number of inputs to the model is equal to the number of features in the dataset. The hidden layer can be LSTM, BiLSTM, or GRU layer combined with the activation, normalization, activity regularization, and dropout layers. The activation layer uses LeakyReLU action function, the normalization layer uses layer normalization, and the regularization layer uses activity regularization along with the l1-l2 penalty function. Regularization and dropout layers lower the likelihood of the model overfitting. The dense layer use 512 neurons and the LeakyReLU activation function. The dense layer works as a classification layer, and the output layer uses two neurons to classify the data as normal or anomalous. Parameters and hyperparameters of LSTM, BiLSTM, and GRU models for binary classification are described in Table 5.

IV. DATA COLLECTION

BoT-IoT [87], IoT-NI [88], IoT-23 [83], MQTT [83], MQTTset [83], and NSLKDD [89] datasets were used to evaluate the proposed models. The first step included processing pcap files of BoT-IoT [90], IoT Network Intrusion [91], IoT-23 [92], MQTT-IoT-IDS2020 [93], and MQTTset [94] datasets. CICFlowmeter [95] extracts 80 network features from pcap files and stores them as CSV files. The adapted datasets are available at [96]. The testbed architecture and attacks on each dataset are described in detail in the previous paper [83]. It is determined how to label each dataset instance based on pre-defined criteria for each dataset instance. Because the network features flow ID, source IP, source port, destination IP, and timestamp characterize communication inside a specific IoT network; they were removed from all

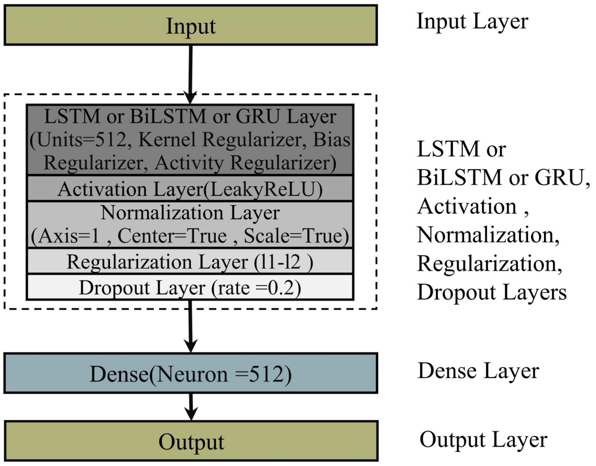

Input
LSTM or BiLSTM or GRU Layer (Units=512, Kernel Regularizer, Bias Regularizer, Activity Regularizer)
Activation Layer(LeakyReLU)
Normalization Layer (Axis=1 , Center=True , Scale=True)
Regularization Layer (l1-l2)
Dropout Layer (rate =0.2)
Dense(Neuron =512)
Output
Input Layer
LSTM or BiLSTM or GRU, Activation , Normalization, Regularization, Dropout Layers
Dense Layer
Output Layer

FIGURE 9. Proposed LSTM, BiLSTM, and GRU models layer's view, parameters, and hyperparameters for binary classification.

TABLE 5. LSTM, BiLSTM and GRU models parameters and hyperparameters for binary classification.

Layer	Layer Name	Layer	Configuration
Input	Input Layer	1	64 Input features
Hidden layers	LSTM or BiLSTM or GRU	1	Units=512 Kernel regularizer, Bias regularizer, Activity regularizer
	Activation	1	LeakyReLU(alpha=0.2)
	Layer Normalization	1	Axis=1, Center=True, Scale=True
	Regularization	1	l1= 0.0001, l2= 0.0001
	Dropout	1	Dropout rate =0.2
Classification	Dense	1	Neuron =512, Activation=ReLU
Output	Output	1	Two neurons, Activation=SoftMax
Hyperparameters	Early Stopping (monitor='loss', patience=5, verbose = 1), optimizers= Adam, loss function= sparse categorical cross entropy, Learning rate=0.001, Batch size=120, epochs= 100 to 500.		

adapted datasets. Non-numeric features are transformed into numeric features. Duplicate instances were produced when the pcap data was converted to CSV files. These redundant instances have been removed from all datasets. Features were normalized within a given range (−1, 1) to eliminate extreme values which will significantly speed up calculations. The mean imputation method fills in missing values in all datasets. It replaces missing values with the average of a variable's remaining values that have missing information. Table 6 presents the BoT-IoT, IoT Network Intrusion, IoT-23, MQTT, and MQTTset datasets with and without redundancy.

The BoT-IoT dataset pcap files were generated by Koroniotis *et al.* [90]. The BoT-IoT dataset testbed comprises virtual machines connected to the network through a LAN and the Internet. The PFSense system establishes a connection between the virtual machines and the Internet. A realistic smart home framework was developed using five IoT devices operated locally and connected to the cloud services using the node-red system for producing network traffic. Normal network traffic is generated with the help of the ostinato tool. The Ubuntu server delivers IoT resources for simulating a real world IoT network, while Kali Linux serves as an attack system. Transmitting IoT messages into the cloud was accomplished using the MQTT protocol. The BoT-IoT dataset category classes are Normal, DDoS, DoS, Scan, and Data Theft. The BoT-IoT dataset instances details are presented in Table 6(a). The DDoS and DoS attack categories were combined to form a single attack class.

The IoT Network Intrusion dataset pcap files were generated by Kang *et al.* [91]. Two IoT devices SKT NGU and EZVIZ Wi-Fi camera were used as victim devices. The wireless network adapter collects the network traffic. Except for the Mirai botnet category, all cyberattacks are constructed of packets collected when modeling cyberattacks that use software such as Nmap. A laptop produced attack packets that were then altered to seem as though they came from an IoT device in the instance of the Mirai botnet. The IoT Network Intrusion dataset category classes are Normal, DoS, MITM, Mirai, and Scan. Table 6(b) shows the details of the IoT Network Intrusion dataset instances.

The IoT-23 dataset was developed by Stratosphere Laboratory CTU University, Czech Republic [92]. There are 20 malicious events and three non-malicious events for IoT devices. The objective of the IoT-23 dataset is to give researchers a large labeled dataset of IoT malware infections to build machine learning models. The IoT-23 dataset's testbed consists of IoT devices and interconnects networks. A standard smart home structure was developed to generate the dataset, consisting of an Amazon Echo device, Philips Hue device, and Somfy door lock IoT devices. There are nine types of attacks in the IoT-23 dataset. The IoT-23 dataset contains twenty different network events that simulate a variety of IoT device use cases. The normal network traffic was also collected by gathering the network traffic of three distinct IoT devices. These three devices are real hardware devices, not a simulation. Like every real IoT network, malicious and normal situations operate with unrestricted Internet connectivity in a managed network. This dataset provides the community with two distinct datasets: the first dataset contains normal and malicious networks, while the other dataset includes only normal IoT network capture. The IoT-23 intrusion dataset's primary advantages are that it effectively simulates a recent trend in IoT network traffic and is one of the few publicly accessible IoT network intrusion datasets. Table 6(c) displays the details of the IoT-23 dataset instances.

The MQTT-IoT-IDS2020 (MQTT) dataset pcap files were created by Hindy *et al.* [93]. Five recorded scenarios are included in the dataset: one normal operation and four attacks operation. The dataset represents a real MQTT IoT network in a typical operating situation. The dataset covers popular MQTT attacks and situations for real world IoT device testing. The MQTT dataset classes are Normal,

TABLE 6. BoT-IoT, IoT Network Intrusion (IoT-NI), IoT-23, MQTT-IoT-IDS2020 (MQTT), and MQTTset datasets classes and instances.

No	Category	With Redundancy	Without Redundancy
(a)	**BoT-IoT dataset classes and instances.**		
0	Normal	105202	77511
1	DDoS	57027372	17420085
2	DoS	37077674	18199716
3	Scan	4734836	4108211
4	Data Theft	454715	445799
(b)	**IoT Network Intrusion (IoT-NI) dataset classes and instances.**		
0	Normal	40073	39851
1	DoS	59391	59391
2	MITM	35377	32909
3	Mirai	415677	366971
4	Scan	75265	72122
(c)	**IoT-23 dataset classes and instances.**		
0	Normal	4313776	4253672
1	Attack	1716778	1699608
2	Mirai	756	756
3	File Download	8035	7707
4	Heartbeat	12895	12648
5	C&C	23981	20612
6	Torii	33858	24492
7	Port Scan	65944863	2999999
8	DDoS	20768988	4619869
9	Okiru	13718252	12908506
(d)	**MQTT-IoT-IDS2020 (MQTT) dataset classes and instances.**		
0	Normal	334318	167159
1	MQTT Bruteforce	2002780	2001972
2	Scan-A	31245	29276
3	Scan-U	33404	27843
5	Sparta	1252259	1217198
(e)	**MQTTset dataset classes and instances.**		
0	Normal	440699	420136
1	Bruteforce	4547	4513
2	MQTTFlood	77793	77756
3	MalariaDoS	11408	11265
5	Malformed	3580	3535
6	SlowITe	3044	3044

MQTT-Bruteforce, Scan-A, Scan-U, and Sparta. Instances of the MQTT dataset that have been constructed are detailed in Table 6(d). Security of IoT networks and devices is crucial due to the increasing number of linked devices and networks. The MQTTset dataset pcap files were generated by Vaccari *et al.* [94]. The testbed for the MQTTset dataset comprises ten IoT sensors in a typical smart home. MQTTset was created with the help of IoT-Flock [97], a network traffic generator capable of simulating IoT devices and networks that use the MQTT and CoAP protocols. The authors connected several IoT sensors to MQTT broker to build a dataset reflective of a real IoT network. The MQTTset dataset category classes are Normal, Bruteforce, MQTTFlood, MalariaDoS, Malformed, and SlowITe. Table 6(e) displays the details of the MQTTset dataset instances.

BoT-IoT, IoT Network Intrusion, IoT-23, MQTT, and MQTTset datasets were combined to make a dataset with several attacks. The new dataset is named IoT-DS2.

TABLE 7. IoT-DS2 dataset classes and instances.

No	Category	BoT-IoT	IoT-NI	IoT-23	MQTT	MQTTset	IoT-DS2
0	Normal	-	-	4253672	-	-	2000000
1	DDoS	17420085	-	-	-	-	500000
2	DoS	-	59391	-	-	-	59391
3	MITM ARP Spoofing	-	32909	-	-	-	32909
4	Mirai	-	366971	-	-	-	366971
5	MQTT Bruteforce	-	-	-	2001972	-	500000
6	Sparta	-	-	-	1217198	-	500000
7	Theft	445799	-	-	-	-	445799
8	Attack	-	-	1699608	-	-	500000
9	C&C	-	-	20612	-	-	20612
10	File Download	-	-	7707	-	-	7707
11	Heartbeat	-	-	12648	-	-	12648
12	Okiru	-	-	12908506	-	-	500000
13	OS Scan	946268	-	-	-	-	500000
14	Port Scan	-	-	2999999	-	-	500000
15	Torii	-	-	24492	-	-	24492
16	MQTT Flood	-	-	-	-	77756	77756
17	Malformed	-	-	-	-	3535	3535
18	SlowITe	-	-	-	-	3044	3044
Total							6554864

Table 7 displays the details of the IoT-DS2 dataset instances. The IoT-DS2 dataset consists of eighteen attack classes and a normal class. The category column represents the name network traffic class which can be normal, or any of the eighteen attack classes. The next five columns represent the dataset used to develop the IoT-DS2 dataset. The final column represents the number of instances extracted from the BoT-IoT, IoT Network Intrusion, IoT-23, MQTT, and MQTTset datasets to generate the IoT-DS2 dataset. These datasets, available at [96], may be used by researchers to develop and test IoT anomaly detection systems. The extracted datasets were split 80% for training and 20% for testing in the first phase, and then the training dataset was split into 80% for training and 20% for validation using a stratified way. Choosing features is a key stage in deep learning model building. Feature selection is an approach that involves the detection and selection of only those features necessary for enhanced prediction. The feature selection strategy not only reduces overfitting but also speeds up model training and makes the model less susceptible to test errors. The recursive feature elimination technique selects 64 features using a random forest algorithm and IoT-DS2 dataset [83]. The feature selection technique was not used on the NSLKDD dataset. The same set of features was utilized in all models and across all datasets.

V. EVALUATION OF RESULTS

The proposed LSTM, BiLSTM, GRU models, and convolutional neural network based LSTM, BiLSTM, and GRU models were validated using accuracy, precision, recall, F1 score, TNR (True Negative Rate), FPR (False Positive Rate), FNR (False Negative Rate), PPV (Positive Predictive Value), NPV (Negative Predictive Value). The formulas for these metrics are presented in Eq. 11 to Eq. 19 for completeness.

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \tag{11}$$

$$Precision = \frac{TP}{(TP + FP)} \tag{12}$$

$$Recall = Sensitivity = \frac{TP}{(TP + FN)} \tag{13}$$

$$F1\ score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \tag{14}$$

$$TNR = Specificity = \frac{TN}{(TN + FP)} \tag{15}$$

$$FPR = \frac{FP}{(FP + TN)} \tag{16}$$

$$FNR = \frac{FN}{(FN + TP)} \tag{17}$$

$$PPV = \frac{TP}{(TP + FP)} \tag{18}$$

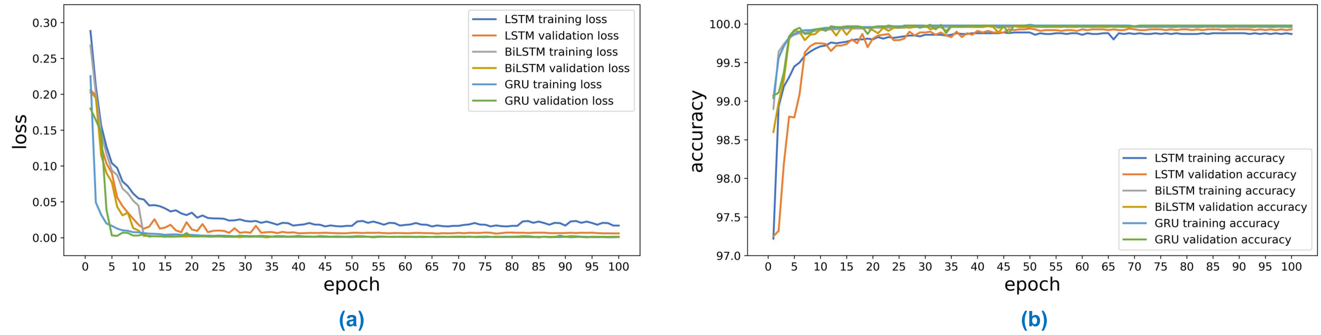
$$NPV = \frac{TN}{(TN + FN)} \tag{19}$$

Seven datasets were used to conduct multiclass and binary classification experiments using the proposed anomaly detection models. This paper combines the Keras framework with the TensorFlow backend to conduct all the experiments. The experiments were conducted using Google Colab. Three processes comprise a neural network model assessment: training, validation, and testing. The classification method favors the majority class if an unequally distributed dataset is used. Class weights and SMOTE approaches were applied to deal with imbalanced classes in the datasets. Class weights were calculated based on the number of class instances, so the class with a small number of instances will get a high weight.

Adam optimizer was used to train each RNN model for 100 epochs and a batch size of 128. The loss functions of a neural network are certainly the most important factor. Sparse categorical cross entropy loss function was used in this paper. Four strategies were used to reduce model overfitting. First, the kernel, bias, and activity regularizers were used at LSTM, BiLSTM, GRU, and CNN layers. Kernel, bias, and activity regularizers use l1-l2 penalty techniques for regularization. Second, the activity regularization layer was used, and third, the dropout layer was used. Finally, an early stopping approach was used to terminate the model if the training loss did not decrease during the training phase. The early stopping strategy also minimizes the likelihood of overfitting, which occurs when a model is trained over a large number of epochs. These strategies eliminate the possibility of the model overfitting. All RNN models were trained with 100 epochs, a batch size of 128, and 5 iterations of patience. The batch size and the number of epochs were increased and decreased to check for improvement in the model's accuracy. The accuracy of the model was not improved. At each epoch value, the accuracy and loss of each model were computed for training and validation sets.

Fig. 10(a) illustrates the loss and Fig. 10(b) accuracy of LSTM, BiLSTM, and GRU models during training and validation using the BoT-IoT dataset. The loss and accuracy of the CNN based LSTM, BiLSTM, and GRU models using the BoT-IoT dataset during training and validation are presented in Fig. 11(a) and 11(b). The loss function calculates the overall deviation across all tests in the training and validation. The early stopping strategy will stop the training process if the training loss does not decrease after a specified number of iterations, reducing the overfitting problem. As illustrated in Fig. 10 and 11, the loss function and the accuracy plot are inversely associated. With 200 and 500 epochs and 10 iterations of patience, the accuracy did not increase. Overfitting of training data occurs when a model is run over a large number of iterations.

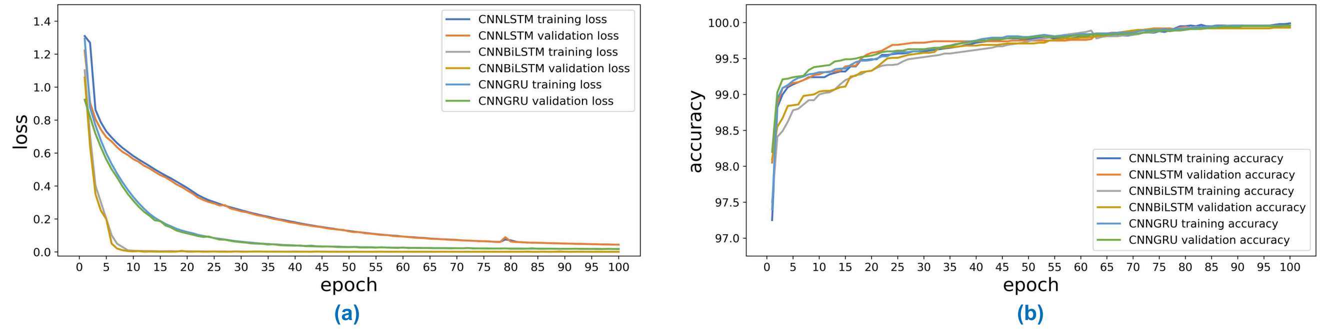
This paper performed multiclass and binary classification for seven datasets. Tables 8, 9, and 10 show the multiclass classification of LSTM, BiLSTM, and GRU models using NSLKDD, BoT-IoT, IoT-NI, IoT-23, MQTT, MQTTset, and IoT-DS2 datasets. The outcome of LSTM, BiLSTM, and GRU models using the NSLKDD dataset is presented in Table 8(a). The accuracy of the NSLKDD dataset was measured at 99.67% for LSTM, 99.82% for BiLSTM, and 99.78% for GRU models. Normal, DoS, and Probe classes achieved a high detection rate while the R2L and U2R detection rates were low. The R2L and U2R attack categories are very rare in the dataset, which is the main reason for the low detection rate of these attack categories. The BiLSTM



(a)

(b)

FIGURE 10. **(a) Loss. (b) Accuracy of multiclass classification using LSTM, BiLSTM, and GRU models utilizing BoT-IoT dataset.**



(a)

(b)

FIGURE 11. **(a) Loss. (b) Accuracy of multiclass classification using CNN based LSTM, BiLSTM, and GRU models utilizing BoT-IoT dataset.**

model achieved a high detection rate among the three models using the NSLKDD dataset. The weighted average of FPR was 0.12%, and FNR was 0.18% for the BiLSTM model.

The accuracy of the BoT-IoT dataset was better than the NSLKDD dataset. Table 8(b) shows the results of LSTM, BiLSTM, and GRU models using the BoT-IoT dataset. Normal, DoS, Scan, and Theft classes of the BoT-IoT dataset achieved at least a 99.50% detection rate. FPR was 0.04%, and FNR was 0.11% for the BiLSTM model. Scan and Theft categories have a high rate of misclassification. The IoT-NI dataset multiclass classification outcomes are presented in Table 8(c). The accuracy of LSTM, BiLSTM, and GRU models using the IoT-NI dataset is 98.14%, 98.89%, and 98.42%, respectively. The precision value for the Normal, DoS, Mirai, and scan classes was relatively high compared to the MITM class using the BiLSTM model. The BiLSTM model's FPR was 0.61%, and the FNR was 1.11% for the IoT-NI dataset.

Table 8(d) shows the evaluation results for the IoT-23 dataset's ten classes using LSTM, BiLSTM, and GRU models. The accuracy was from 99.71% to 99.83% for the proposed models. The BiLSTM model FPR was 0.04%, while the FNR was 0.19%. The IoT-23 dataset has an average precision of 99.81% and a recall of 99.82%. The MQTT and MQTTset datasets had the highest precision and recall scores of any dataset evaluated in this paper for the LSTM, BiLSTM, and GRU models. The results of LSTM, BiLSTM, and GRU models for MQTT and MQTTset datasets are presented in Tables 8(e) and 8(f). The accuracy of MQTT and MQTTset datasets was 99.90% to 99.99%. The BiLSTM model achieved high precision and recall for MQTT and MQTTset datasets. The FPR was 0.0042%, and FNR was 0.06% for the MQTT dataset via the BiLSTM model. The FPR was 0.062%, and the FNR was 0.06% using the BiLSTM model on the MQTTset dataset.

The outcome of LSTM, BiLSTM, and GRU models using the IoT-DS2 dataset are presented in Table 9. There are 19 classes in the IoT-DS2 dataset. The accuracy of LSTM, BiLSTM, and GRU models using the IoT-DS2 dataset were 99.31%, 99.48%, and 99.32%, respectively. The BiLSTM model outperformed the LSTM and GRU models with precision and recall of 99.48% and 99.46%, respectively. The FPR for the IoT-DS2 dataset using the BiLSTM model was 0.06%, while the FNR was 0.51%. The precision rate for the Normal class was at least 99% in LSTM, BiLSTM, and GRU models. MITM, Heartbeat, Malformed Data, and C&C are all attack types with a precision rate of less than 98%. Sensitivity, specificity, PPV, and NPV results for multiclass classification of the NSLKDD, BoT-IoT, IoT-NI, IoT-23, MQTT, MQTTset, and IoT-DS2 datasets LSTM, BiLSTM, and GRU models are presented in Table 10.

A convolutional neural network requires significantly less preprocessing than other deep learning classification techniques, making it a more efficient classification approach. The article [83] discusses the performance metrics of the convolutional neural network anomaly detection model for IoT networks. It also compares them to other deep learning based systems to assess the CNN model's ability to recognize various IoT network attacks. Convolutional neural network benefits in anomaly detection are transferred to image recognition applications. Using appropriate filters, a convolutional neural network can effectively capture spatial and temporal connectivity in anomaly detection problems. Convolutional neural networks are excellent for prediction because they learn input features without losing important information. The first hidden layer was created using a 1D convolutional neural network to learn the network features. The 1D convolutional neural network layer was followed by two hidden LSTM, BiLSTM, or GRU layers. The proposed CNN based LSTM, BiLSTM, and GRU models were evaluated using NSLKDD, BoT-IoT, IoT-NI, IoT-23, MQTT, MQTTset, and IoT-DS2 datasets. CNN based LSTM, BiLSTM, and GRU models evaluation results are presented in Tables 11, 12 and 13.

The NSLKDD dataset evaluation results using CNN based LSTM, BiLSTM, and GRU models are presented in Table 11(a). Precision and recall values of proposed models for the NSLKDD dataset have improved compared to the results shown in Table 8(a). CNNBiLSTM model achieved high precision and recall values, while the LSTM model had the lowest detection rate for the NSLKDD dataset. FPR was 0.08%, and FNR was 0.12% for the BiLSTM model. The BoT-IoT dataset evaluation outcome is presented in Table 11(b). All proposed models designed for the BoT-IoT dataset show improved detection rates and reduced FPR and FNR compared to the findings in Table 8(b). The FPR was 0.02%, and FNR was 0.06% for the BiLSTM model. The IoT-NI dataset evaluation results are presented in Table 11(c). The average precision of LSTM, BiLSTM, and GRU models based on CNN is 98.35%, 99.38%, and 98.77%, respectively. The precision value for IoT-NI was measured at 98.14% when only LSTM was used, but when LSTM was combined with CNN, the average precision value was increased to 98.35%. The average precision value for CNNBiLSTM and CNNGRU models was increased to 99.38% and 98.77%, respectively. The BiLSTM model's FPR was 0.50%, and the FNR was 0.66% for the IoT-NI dataset.

The IoT-23 dataset evaluation result is presented in Table 11(d). The CNNBiLSTM model performed better than other CNN based models in identifying normal and anomalous categories of the IoT-23 dataset. The precision and recall values for all proposed CNN based models utilizing the IoT-23 dataset have improved. As shown in Table 11(d), the detection rate for CNNBiLSTM and CNNGRU models has been enhanced to 99.87 and 99.86%, respectively. The FPR value was reduced to 0.02%, and FNR was reduced to 0.11% using the CNNBiLSTM model. Table 11(e) shows the multiclass classification of the MQTT dataset, while Table 11(f) shows the multiclass classification of the MQTTset dataset. MQTT and MQTTset datasets achieved high precision and recall values compared to other datasets used in this paper. The FPR for the MQTT dataset was

TABLE 8. Accuracy, Precision, Recall, and F1 Score of NSLKDD, BoT-IoT, IoT-NI, IoT-23, MQTT, and MQTTset datasets multiclass classification using LSTM, BiLSTM, and GRU models.

Model	LSTM				BiLSTM				GRU			
Class	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
(a) NSLKDD dataset multiclass classification using LSTM, BiLSTM, and GRU models.												
Normal	99.67	99.71	99.75	99.73	99.82	99.89	99.88	99.89	99.78	99.88	99.73	99.80
DoS	99.67	99.76	99.94	99.85	99.82	99.86	99.97	99.92	99.78	99.81	99.97	99.89
Probe	99.67	99.16	95.93	97.52	99.82	99.07	99.07	99.07	99.78	99.18	99.29	99.24
R2L	99.67	93.26	89.11	91.14	99.82	95.96	87.56	91.57	99.78	95.74	94.74	95.24
U2R	99.67	94.61	91.47	93.01	99.82	96.67	96.21	96.44	99.78	94.61	93.69	94.15
(b) BoT-IoT dataset multiclass classification using LSTM, BiLSTM, and GRU models.												
Normal	99.80	99.58	99.87	99.73	99.89	99.79	99.87	99.83	99.87	99.58	99.87	99.73
DoS	99.80	99.87	99.97	99.92	99.89	99.90	99.97	99.94	99.87	99.87	99.97	99.92
Scan	99.80	99.74	99.77	99.76	99.89	99.85	99.91	99.88	99.87	99.85	99.88	99.87
Theft	99.80	99.83	99.66	99.75	99.89	99.94	99.81	99.88	99.87	99.94	99.77	99.86
(c) IoT-NI dataset multiclass classification using LSTM, BiLSTM, and GRU models.												
Normal	98.14	99.15	99.05	99.10	98.89	98.89	98.30	98.59	98.42	99.15	99.05	99.10
DoS	98.14	96.45	97.04	96.74	98.89	99.97	99.19	99.58	98.42	97.25	95.62	96.43
Mirai	98.14	98.90	98.69	98.79	98.89	99.57	99.20	99.38	98.42	98.93	98.96	98.94
MITM	98.14	97.13	98.03	97.58	98.89	95.76	98.30	97.01	98.42	97.13	98.03	97.58
Scan	98.14	95.62	95.83	95.72	98.89	96.00	97.62	96.80	98.42	96.96	97.87	97.42
(d) IoT-23 dataset multiclass classification using LSTM, BiLSTM, and GRU models.												
Normal	99.71	99.67	99.78	99.72	99.83	99.80	99.88	99.84	99.81	99.73	99.84	99.78
DDoS	99.71	99.55	99.85	99.70	99.83	99.65	99.90	99.78	99.81	99.64	99.90	99.77
Attack	99.71	94.40	90.08	92.19	99.83	99.99	90.08	94.78	99.81	96.75	90.84	93.70
Mirai	99.71	92.92	88.44	90.62	99.83	96.22	96.10	96.16	99.81	98.01	97.28	97.64
File Download	99.71	94.55	93.89	94.22	99.83	96.55	90.25	93.29	99.81	94.17	91.74	92.94
Heartbeat	99.71	96.96	92.54	94.70	99.83	97.23	97.48	97.36	99.81	98.32	95.43	96.86
C&C	99.71	98.34	99.73	99.03	99.83	99.73	99.86	99.79	99.81	99.71	99.88	99.79
Torii	99.71	99.98	99.97	99.98	99.83	99.99	99.99	99.99	99.81	99.99	99.99	99.99
Port Scan	99.71	99.97	99.99	99.98	99.83	99.99	99.99	99.99	99.81	99.99	99.99	99.99
Okiru	99.71	99.94	99.96	99.95	99.83	99.99	99.99	99.99	99.81	99.99	99.99	99.99
(e) MQTT dataset multiclass classification using LSTM, BiLSTM, and GRU models.												
Normal	99.90	97.86	99.99	98.92	99.92	98.41	99.99	99.20	99.90	98.31	99.99	99.14
MQTT-BF	99.90	99.99	99.85	99.92	99.92	99.99	99.86	99.93	99.90	99.99	99.87	99.93
Scan-A	99.90	99.99	99.99	99.99	99.92	99.99	99.99	99.99	99.90	99.98	99.99	99.99
Scan-U	99.90	99.94	99.24	99.59	99.92	99.99	99.97	99.99	99.90	99.94	99.15	99.54
Sparta	99.90	99.99	99.97	99.99	99.92	99.99	99.99	99.99	99.90	99.99	99.99	99.99
(f) MQTTset dataset multiclass classification using LSTM, BiLSTM, and GRU models.												
Normal	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.92	99.99	99.99	99.99
MQTT Flood	99.99	99.99	99.99	99.99	99.99	99.99	99.98	99.99	99.92	99.99	99.97	99.99
DoS	99.99	99.99	99.78	99.89	99.99	99.96	99.69	99.83	99.92	99.78	99.65	99.72
MQTT-BF	99.99	98.59	99.99	99.29	99.99	99.13	99.89	99.51	99.92	98.16	93.94	96.00
Malformed Data	99.99	99.56	98.69	99.12	99.99	99.71	99.56	99.64	99.92	92.28	97.38	94.76
SlowITe	99.99	99.99	99.84	99.92	99.99	99.99	99.99	99.99	99.92	99.99	99.99	99.99

TABLE 9. Accuracy, Precision, Recall, and F1 Score of IoT-DS2 dataset multiclass classification using LSTM, BiLSTM, and GRU models.

Model	LSTM				BiLSTM				GRU			
Class	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
Normal	99.31	99.03	99.51	99.27	99.48	99.50	99.58	99.54	99.32	99.50	99.30	99.40
DDoS	99.31	99.84	99.86	99.85	99.48	99.65	99.90	99.77	99.32	99.60	99.89	99.75
DoS	99.31	99.63	97.63	98.62	99.48	99.48	95.27	97.33	99.32	99.18	97.77	98.47
MITM	99.31	96.25	87.48	91.66	99.48	94.84	94.06	94.45	99.32	96.77	94.82	95.78
Mirai	99.31	98.24	99.01	98.62	99.48	99.10	97.75	98.42	99.32	99.08	98.64	98.86
MQTT-BF	99.31	99.94	98.20	99.06	99.48	98.98	99.63	99.30	99.32	99.50	98.02	98.76
Sparta	99.31	98.14	99.50	98.81	99.48	99.20	99.83	99.51	99.32	97.98	99.47	98.72
Theft	99.31	98.92	96.68	97.79	99.48	98.44	97.24	97.84	99.32	90.93	95.06	92.95
Attack	99.31	99.70	99.46	99.58	99.48	99.49	99.76	99.62	99.32	99.29	97.87	98.58
C&C	99.31	97.32	89.26	93.12	99.48	97.99	96.49	97.24	99.32	98.32	92.42	95.28
File Download	99.31	98.99	97.88	98.43	99.48	98.73	97.13	97.92	99.32	98.72	91.20	94.81
Heartbeat	99.31	90.39	98.51	94.27	99.48	97.02	98.43	97.72	99.32	91.48	97.11	94.21
Okiru	99.31	99.99	99.99	99.99	99.48	99.99	99.99	99.99	99.32	99.99	99.99	99.99
OS Scan	99.31	98.83	99.08	98.95	99.48	99.17	99.53	99.35	99.32	99.07	99.51	99.29
Port Scan	99.31	99.99	99.99	99.99	99.48	99.99	99.98	99.99	99.32	99.99	99.98	99.99
Torii	99.31	99.99	99.94	99.97	99.48	99.99	99.94	99.97	99.32	99.72	99.92	99.82
MQTT Flood	99.31	99.99	99.99	99.99	99.48	99.99	99.99	99.99	99.32	99.99	99.99	99.99
Malformed Data	99.31	97.13	97.13	97.13	99.48	97.04	94.25	95.63	99.32	92.70	90.29	91.48
SlowITe	98.76	99.83	99.99	99.92	99.48	98.17	99.25	98.71	99.32	96.67	99.47	98.05

TABLE 10. Sensitivity, Specificity, PPV, and NPV of NSLKDD, BoT-IoT, IoT-NI, IoT-23, MQTT, MQTTset and IoT-DS2 datasets multiclass classification using LSTM, BiLSTM, and GRU models.

Model	LSTM				BiLSTM				GRU			
Dataset	Sensitivity	Specificity	PPV	NPV	Sensitivity	Specificity	PPV	NPV	Sensitivity	Specificity	PPV	NPV
NSLKDD	99.72	99.80	99.72	99.87	99.80	99.87	99.80	99.88	99.87	99.93	99.87	99.93
BoT-IoT	99.80	99.92	99.80	99.91	99.89	99.96	99.89	99.95	99.87	99.95	99.87	99.94
IoT-NI	98.14	98.59	98.14	98.36	98.89	99.39	98.90	98.98	98.42	98.66	98.42	98.69
IoT-23	99.71	99.95	99.71	99.96	99.81	99.96	99.81	99.98	99.83	99.97	99.83	99.98
MQTT	99.90	99.99	99.90	99.87	99.94	99.99	99.95	99.93	99.90	99.98	99.90	99.90
MQTTset	99.95	99.92	99.95	99.95	99.97	99.94	99.97	99.95	99.92	99.97	99.92	99.99
IoT-DS2	99.31	99.93	99.32	99.95	99.48	99.94	99.48	99.96	99.32	99.93	99.32	99.95

0.002%, while the FNR was 0.04% using the CNNLSTM model. FPR was 0.0016% for the MQTT dataset, while FNR was 0.03% when the CNNBiLSTM model was used. FPR was 0.0026%, while FNR was 0.046% for the MQTT dataset when the CNNGRU model was used. Only 15, 12, and 12 instances of the MQTTset dataset were misclassified using the CNNLSTM, CNNBiLSTM, and CNNGRU models.

The evaluation results for 18 attack categories in the IoT-DS2 dataset are shown in Table 12. The precision values for CNNLSTM, CNNBiLSTM, and CNNGRU models were 99.45%, 99.57%, and 99.52%, respectively. The FPR for the IoT-DS2 dataset was 0.06%, and the FNR was 0.48% using the CNNLSTM model. The FPR was 0.03% for the IoT-DS2 dataset, whereas the FNR was 0.40% for the CNNBiLSTM model. FPR was 0.04 %, and FNR was 0.47 % when the CNNGRU model was used with the IoT-DS2 dataset. Sensitivity, specificity, PPV, and NPV results for multiclass classification of the NSLKDD, BoT-IoT, IoT-NI, IoT-23, MQTT, and MQTTset datasets using CNNLSTM, CNNBiLSTM, and CNNGRU models are presented in Table 13.

A lightweight binary classification model designed using a single recurrent neural network layer. Three models were created using a single hidden layer from the LSTM, BiLSTM, and GRU networks for binary classification. They were tested using NSLKDD, BoT-IoT, IoT-NI, IoT-23, MQTT, MQTTset, and IoT-DS2 datasets. The entire dataset was converted to a binary label classification scheme. Table 14 summarizes the assessment results for binary classification using the LSTM model. The MQTTset dataset achieved high precision and recall values. IoT-DS2 dataset, which combines all IoT datasets, reached an accuracy of 99.42% using the LSTM model. The number of normal class

TABLE 11. Accuracy, Precision, Recall, and F1 Score of NSLKDD, BoT-IoT, IoT-NI, IoT-23, MQTT, and MQTTset datasets multiclass classification using CNNLSTM, CNNBiLSTM, and CNNGRU models.

Model	CNNLSTM				CNNBiLSTM				CNNGRU			
Class	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
(a) NSLKDD dataset multiclass classification using CNNLSTM, CNNBiLSTM, and CNNGRU models.												
Normal	99.81	99.91	99.91	99.91	99.88	99.90	99.96	99.93	99.85	99.87	99.96	99.92
DoS	99.81	99.83	99.96	99.90	99.88	99.93	99.96	99.95	99.85	99.89	99.95	99.92
Probe	99.81	98.47	96.76	97.61	99.88	99.24	99.75	99.49	99.85	99.75	98.51	99.13
R2L	99.81	95.96	90.91	93.37	99.88	97.99	89.04	93.30	99.85	94.61	92.34	93.46
U2R	99.81	97.18	97.18	97.18	99.88	98.27	97.42	97.84	99.85	99.11	93.31	96.12
(b) BoT-IoT dataset multiclass classification using CNNLSTM, CNNBiLSTM, and CNNGRU models.												
Normal	99.93	99.69	99.58	99.63	99.96	99.79	99.79	99.79	99.95	99.75	99.62	99.68
DoS	99.93	99.93	99.99	99.96	99.96	99.97	99.98	99.97	99.95	99.96	99.99	99.97
Scan	99.93	99.95	99.93	99.94	99.96	99.97	99.97	99.97	99.95	99.96	99.96	99.96
Theft	99.93	99.95	99.93	99.94	99.96	99.98	99.97	99.97	99.95	99.96	99.96	99.96
(c) IoT-NI dataset multiclass classification using CNNLSTM, CNNBiLSTM, and CNNGRU models.												
Normal	98.35	99.55	94.10	96.75	99.18	99.33	98.72	99.02	98.76	99.76	97.65	98.69
DoS	98.35	99.92	99.19	99.55	99.18	99.98	99.39	99.69	98.76	99.98	99.37	99.68
Mirai	98.35	98.27	99.47	98.87	99.18	99.49	99.59	99.54	98.76	99.18	99.10	99.14
MITM	98.35	96.01	95.74	95.87	99.18	96.50	96.03	96.26	98.76	92.22	94.81	93.50
Scan	98.35	97.88	95.38	96.61	99.18	98.08	98.57	98.32	98.76	98.11	98.90	98.51
(d) IoT-23 dataset multiclass classification using CNNLSTM, CNNBiLSTM, and CNNGRU models.												
Normal	99.83	99.68	99.89	99.79	99.87	99.82	99.93	99.87	99.86	99.84	99.86	99.85
DDoS	99.83	99.71	99.39	99.55	99.87	99.73	99.94	99.84	99.86	99.73	99.96	99.84
Attack	99.83	98.50	99.99	99.24	99.87	99.17	90.84	94.82	99.86	99.99	99.99	99.99
Mirai	99.83	98.63	97.84	98.23	99.87	98.57	98.02	98.29	99.86	98.21	98.21	98.21
File Download	99.83	95.73	94.70	95.21	99.87	97.66	93.09	95.32	99.86	96.52	95.85	96.18
Heartbeat	99.83	99.07	97.60	98.33	99.87	98.17	97.11	97.63	99.86	97.75	96.38	97.06
C&C	99.83	99.84	99.86	99.85	99.87	99.90	99.86	99.88	99.86	99.86	99.88	99.87
Torii	99.83	99.99	99.99	99.99	99.87	99.99	99.99	99.99	99.86	99.99	99.99	99.99
Port Scan	99.83	99.99	99.99	99.99	99.87	99.99	99.99	99.99	99.86	99.99	99.99	99.99
Okiru	99.83	99.99	99.99	99.99	99.87	99.99	99.99	99.99	99.86	99.99	99.99	99.99
(e) MQTT dataset multiclass classification using CNNLSTM, CNNBiLSTM, and CNNGRU models.												
Normal	99.96	99.18	99.99	99.59	99.97	99.40	99.99	99.70	99.95	99.08	99.99	99.53
MQTT-BF	99.96	99.99	99.94	99.97	99.97	99.99	99.95	99.98	99.95	99.99	99.92	99.96
Scan-A	99.96	99.98	99.99	99.99	99.97	99.99	99.99	99.99	99.95	99.94	99.99	99.97
Scan-U	99.96	99.94	99.50	99.72	99.97	99.94	99.96	99.95	99.95	99.99	99.94	99.97
Sparta	99.96	99.99	99.99	99.99	99.97	99.99	99.99	99.99	99.95	99.99	99.99	99.99
(f) MQTTset dataset multiclass classification using CNNLSTM, CNNBiLSTM, and CNNGRU models.												
Normal	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99
MQTT Flood	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99
DoS	99.99	99.96	99.69	99.83	99.99	99.91	99.87	99.89	99.99	99.83	99.83	99.83
MQTT-BF	99.99	99.45	99.89	99.67	99.99	99.89	98.90	99.39	99.99	99.78	99.56	99.67
Malformed Data	99.99	99.71	99.99	99.85	99.99	98.57	99.99	99.28	99.99	99.56	99.56	99.56
SlowITe	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99

TABLE 12. Accuracy, Precision, Recall, and F1 Score of IoT-DS2 dataset multiclass classification using CNNLSTM, CNNBiLSTM, and CNNGRU models.

Model	CNNLSTM				CNNBiLSTM				CNNGRU			
Class	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
Normal	99.45	99.53	99.63	99.58	99.57	99.51	99.67	99.59	99.52	99.39	99.68	99.53
DDoS	99.45	99.87	99.89	99.88	99.57	99.93	99.90	99.92	99.52	99.97	99.90	99.93
DoS	99.45	99.75	97.58	98.65	99.57	99.50	97.55	98.52	99.52	99.13	97.31	98.21
MITM	99.45	95.50	92.54	93.99	99.57	95.54	91.87	93.67	99.52	92.22	91.28	91.75
Mirai	99.45	98.92	98.96	98.94	99.57	98.94	99.16	99.05	99.52	98.99	98.66	98.83
MQTT-BF	99.45	99.80	96.98	98.37	99.57	99.76	98.07	98.91	99.52	99.70	99.00	99.35
Sparta	99.45	98.73	98.73	98.73	99.57	98.07	99.41	98.73	99.52	98.73	99.64	99.18
Theft	99.45	99.17	96.43	97.78	99.57	99.09	96.68	97.87	99.52	99.16	96.19	97.66
Attack	99.45	99.52	99.89	99.70	99.57	99.44	99.84	99.64	99.52	99.17	99.68	99.42
C&C	99.45	98.66	98.29	98.48	99.57	99.43	98.00	98.71	99.52	98.69	97.17	97.92
File Download	99.45	99.05	97.44	98.24	99.57	98.87	97.94	98.40	99.52	99.05	97.32	98.17
Heartbeat	99.45	97.62	98.04	97.83	99.57	97.50	97.88	97.69	99.52	97.00	95.17	96.08
Okiru	99.45	99.28	99.99	99.64	99.57	99.99	99.99	99.99	99.52	99.99	99.99	99.99
OS Scan	99.45	98.97	99.27	99.12	99.57	99.21	99.55	99.38	99.52	98.98	99.59	99.29
Port Scan	99.45	99.61	99.98	99.80	99.57	99.99	99.99	99.99	99.52	99.99	99.98	99.99
Torii	99.45	99.99	99.92	99.96	99.57	99.99	99.96	99.98	99.52	99.92	99.96	99.94
MQTT Flood	99.45	99.98	99.99	99.99	99.57	99.98	99.99	99.99	99.52	99.98	99.99	99.99
Malformed Data	99.45	97.84	99.32	98.57	99.57	97.83	98.77	98.30	99.52	96.72	92.89	94.77
SlowITe	99.45	99.69	99.60	99.65	99.57	99.69	99.82	99.76	99.52	98.26	99.47	98.86

TABLE 13. Sensitivity, Specificity, PPV, and NPV of NSLKDD, BoT-IoT, IoT-NI, IoT-23, MQTT, MQTTset and IoT-DS2 datasets multiclass classification using CNNLSTM, CNNBiLSTM, and CNNGRU models.

Model	CNNLSTM				CNNBiLSTM				CNNGRU			
Dataset	Sensitivity	Specificity	PPV	NPV	Sensitivity	Specificity	PPV	NPV	Sensitivity	Specificity	PPV	NPV
NSLKDD	99.90	99.95	99.90	99.94	99.92	99.96	99.92	99.97	99.88	99.93	99.88	99.96
BoT-IoT	99.93	99.98	99.93	99.98	99.96	99.99	99.96	99.99	99.95	99.98	99.95	99.99
IoT-NI	98.35	97.88	98.35	99.22	99.18	99.34	99.18	99.46	98.76	98.97	98.77	98.88
IoT-23	99.83	99.96	99.83	99.98	99.87	99.97	99.87	99.99	99.86	99.98	99.86	99.98
MQTT	99.96	99.99	99.96	99.95	99.97	99.99	99.97	99.96	99.95	99.99	99.95	99.94
MQTTset	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99
IoT-DS2	99.45	99.94	99.45	99.96	99.57	99.96	99.57	99.97	99.52	99.96	99.52	99.97

instances was small in the IoT-NI dataset compared to the anomaly class; as a result, the normal class achieved a low detection rate compared to the anomaly class. The detection rate was 99.98%, FPR was 0.04%, and FNR was 0.02% for MQTTset using the LSTM model.

The findings of the binary classification evaluation using the BiLSTM model are summarized in Table 15. The MQTTset dataset demonstrated a high level of precision and recall. The detection rate for MQTTset using the BiLSTM model was 99.98%, FPR was 0.03%, and FNR was 0.02%. IoT-DS2 has an average detection rate of 99.81%, considerably higher than the LSTM model detection rate. The results of the binary classification evaluation using the GRU model are summarized in Table 16. The GRU model performs better compared to the LSTM model. The BiLSTM model performed better than the GRU model. The GRU model attained an accuracy of 99.96% when used with the MQTTset dataset. The detection rate for MQTTset using the GRU model was 99.97%, FPR was 0.06%, and FNR was 0.03%. IoT-DS2 has an accuracy of 99.45%, which is high than the accuracy of the LSTM model but lower than the accuracy of the BiLSTM model.

The NSLKDD, BoT-IoT, IoT-NI, IoT-23, MQTT, MQTTset, and IoT-DS2 datasets were accurately classified using a lightweight binary classification model based on a single recurrent neural network hidden layer. In addition, binary classification performance was evaluated using the receiver operating characteristic area under the curve (ROC AUC). The validation set and testing set of the NSLKDD, BoT-IoT, IoT-NI, IoT-23, MQTT, and MQTTset datasets were plotted using ROC curves. Figure 12 shows the ROC curve for the validation set of NSLKDD, BoT-IoT, IoT-NI, IoT-23, MQTT, and MQTTset datasets binary classification using the BiLSTM model, whereas Fig. 13 shows the ROC curve for the testing set.

TABLE 14. NSLKDD, BoT-IoT, IoT-NI, IoT-23, MQTT, MQTTset, and IoT-DS2 datasets binary classification using LSTM model.

Dataset	Class	Accuracy	Precision	Recall	F1 Score	TNR	FPR	FNR
NSLKDD	Normal	99.91	99.89	99.89	99.89	99.97	0.03	0.11
	Anomaly		99.97	99.97	99.97	99.89	0.11	0.03
BoT-IoT	Normal	99.90	99.52	99.45	99.49	99.97	0.03	0.55
	Anomaly		99.97	99.97	99.97	99.45	0.55	0.03
IoT-NI	Normal	99.10	96.36	97.02	96.69	99.72	0.28	2.98
	Anomaly		99.77	99.72	99.75	97.02	2.98	0.28
IoT-23	Normal	99.80	99.74	99.93	99.84	99.89	0.11	0.07
	Anomaly		99.97	99.89	99.93	99.93	0.07	0.11
MQTT	Normal	99.91	99.21	99.81	99.51	99.96	0.04	0.19
	Anomaly		99.99	99.96	99.98	99.81	0.19	0.04
MQTTset	Normal	99.96	99.99	99.99	99.99	99.95	0.05	0.01
	Anomaly		99.94	99.95	99.94	99.99	0.01	0.05
IoT-DS2	Normal	99.42	98.96	99.11	99.03	99.82	0.18	0.89
	Anomaly		99.84	99.82	99.83	99.11	0.89	0.18

TABLE 15. NSLKDD, BoT-IoT, IoT-NI, IoT-23, MQTT, MQTTset, and IoT-DS2 datasets binary classification using BiLSTM model.

Dataset	Class	Accuracy	Precision	Recall	F1 Score	TNR	FPR	FNR
NSLKDD	Normal	99.92	99.88	99.92	99.90	99.97	0.03	0.08
	Anomaly		99.98	99.97	99.98	99.92	0.08	0.03
BoT-IoT	Normal	99.96	99.76	99.83	99.80	99.99	0.01	0.17
	Anomaly		99.99	99.99	99.99	99.83	0.17	0.01
IoT-NI	Normal	99.25	96.84	97.80	97.32	99.76	0.24	2.20
	Anomaly		99.83	99.76	99.80	97.80	2.20	0.24
IoT-23	Normal	99.70	99.63	99.71	99.67	99.84	0.16	0.29
	Anomaly		99.88	99.84	99.86	99.71	0.29	0.16
MQTT	Normal	99.88	98.74	99.84	99.29	99.94	0.06	0.16
	Anomaly		99.99	99.94	99.96	99.84	0.16	0.06
MQTTset	Normal	99.96	99.99	99.99	99.99	99.95	0.05	0.01
	Anomaly		99.96	99.95	99.95	99.99	0.01	0.05
IoT-DS2	Normal	99.65	99.31	99.40	99.36	99.88	0.12	0.60
	Anomaly		99.89	99.88	99.89	99.40	0.60	0.12

TABLE 16. NSLKDD, BoT-IoT, IoT-NI, IoT-23, MQTT, MQTTset, and IoT-DS2 datasets binary classification using GRU model.

Dataset	Class	Accuracy	Precision	Recall	F1 Score	TNR	FPR	FNR
NSLKDD	Normal	99.91	99.89	99.89	99.89	99.97	0.03	0.11
	Anomaly		99.97	99.97	99.97	99.89	0.11	0.03
BoT-IoT	Normal	99.93	99.71	99.61	99.66	99.98	0.02	0.39
	Anomaly		99.98	99.98	99.98	99.61	0.39	0.02
IoT-NI	Normal	99.25	96.84	97.80	97.32	99.76	0.24	2.20
	Anomaly		99.83	99.76	99.80	97.80	2.20	0.24
IoT-23	Normal	99.23	99.36	99.36	99.36	99.73	0.27	0.64
	Anomaly		99.72	99.73	99.72	99.36	0.64	0.27
MQTT	Normal	99.88	98.74	99.84	99.29	99.94	0.06	0.16
	Anomaly		99.99	99.94	99.96	99.84	0.16	0.06
MQTTset	Normal	99.96	99.99	99.99	99.99	99.95	0.05	0.01
	Anomaly		99.96	99.95	99.95	99.99	0.01	0.05
IoT-DS2	Normal	99.45	99.15	98.99	99.07	99.85	0.15	1.01
	Anomaly		99.82	99.85	99.84	98.99	1.01	0.15

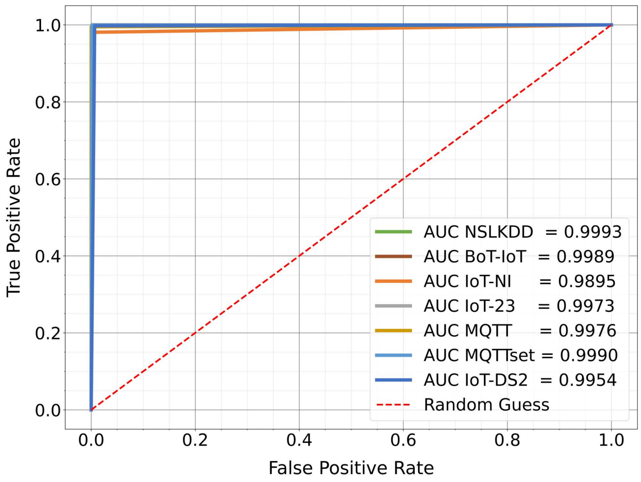


FIGURE 12. ROC curve using the BiLSTM model for the validation set of NSLKDD, BoT-IoT, IoT-NI, IoT-23, MQTT, MQTTset, and IoT-DS2 datasets.

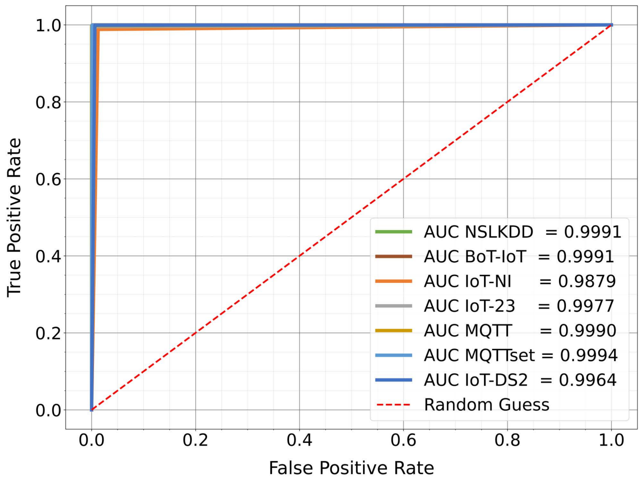


FIGURE 13. ROC curve using the BiLSTM model for the testing set of NSLKDD, BoT-IoT, IoT-NI, IoT-23, MQTT, MQTTset, and IoT-DS2 datasets.

Next, we expand the binary classification model to classify normal and individual anomalous classes of the IoT-DS2 dataset. The IoT-DS2 dataset was divided into eighteen subsets, with each subset consisting of a normal data class and an anomalous data class. Each subset was used to evaluate LSTM, BiLSTM, and GRU models. The accuracy, precision, recall, and F1 score of the IoT-DS2 dataset normal and individual anomalous classes using LSTM, BiLSTM, and GRU models are presented in Fig. 14. The BiLSTM model performs better compared to LSTM and GRU models. A single hidden layer recurrent neural network correctly detected normal and anomalous occurrences in the IoT-DS2 dataset. These evaluation results indicate that a single layer recurrent neural network model can be used to detect anomalies in various IoT networks.

Class weights were used in the training phase to address imbalanced classes in the datasets. The evaluation results presented in the preceding sections are based on class weights. Class weights were calculated based on the number of instances of each class; therefore, a minority class with a small number of instances will receive high importance. Next, SMOTE was implemented to resolve the imbalances between the classes. SMOTE was implemented for NSLKDD, BoT-IoT, IoT-NI, IoT-23, MQTT, and MQTTset datasets. The borderline SMOTE algorithm was used to generate new synthetic samples. Random state was used to control the algorithm's randomization. K is the number of neighbors used in the borderline SMOTE algorithm to calculate the average distance to minority samples. The optimal number of neighbors for estimating the average distance to minority samples was determined to be K=6 to 10 after testing a variety of K values. New synthetic samples were created using the borderline algorithm. The performance of the borderline SMOTE algorithm was evaluated using CNN based LSTM, BiLSTM, and GRU models. The borderline SMOTE algorithm ensures that the training set is balanced appropriately. NSLKDD, BoT-IoT, IoT-NI, IoT-23, MQTT, and MQTTset datasets class balancing using SMOTE and multiclass classification using CNN based LSTM, BiLSTM, and GRU models are presented in Table 17.

The NSLKDD dataset Prob, U2L, and R2L classes detection rates have been improved, as shown in Table 17(a). One million instances were selected randomly from the BoT-IoT dataset's DoS and Scan classes, and then borderline SMOTE was used to balance the remaining classes. Table 17(b) shows the detection rate for the Normal and Theft classes has been enhanced. Four classes were balanced in the IoT-NI dataset. As presented in Table 17(c), the detection rates for the Normal, DoS, MITM, and Scan classes have been improved for the IoT-NI dataset. One million instances were chosen randomly from the IoT-23 dataset's Normal, Attack, Port Scan, DDoS, and Okiru classes, and then borderline SMOTE was used to balance the IoT-23 dataset. The IoT-23 dataset has improved the detection rate of minority classes, as seen in Table 17(d).

One million instances were selected randomly from the MQTT dataset's MQTT-BF and Sparta classes, and then borderline SMOTE was used to balance the MQTT dataset. The MQTT dataset achieved a high detection rate when class weight was used to handle imbalance classes in the dataset. Each model enhanced the overall detection rate in the MQTT dataset, as shown in Table 17(e). All anomalous classes were in the minority in the MQTTset dataset. Class weight was used to address imbalance classes in the MQTTset dataset, resulting in a high detection rate, but when borderline SMOTE was used on the MQTTset dataset, each model improved the overall detection rate, as shown in Table 17(f). The borderline SMOTE technique performs better than class weight, but the borderline SMOTE technique requires more computing resources than class weight. The overfitting of the LSTM, BiLSTM, and GRU models was investigated using the dropout layer and early stopping strategies. The overfitting of LSTM, BiLSTM,

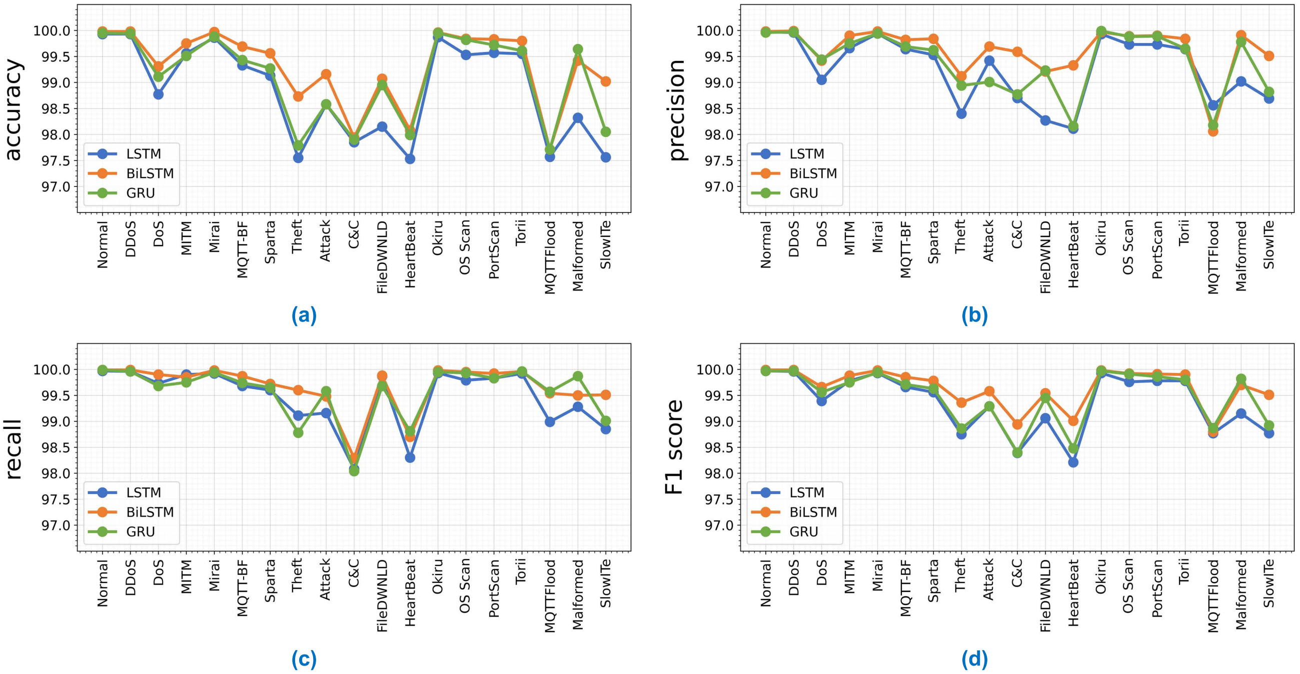


FIGURE 14. (a) Accuracy. (b) Precision. (c) Recall. (d) F1 Score of binary classification using LSTM, BiLSTM, and GRU models for normal and individual anomalous classes using IoT-DS2 dataset.

and GRU models was further evaluated using a 5-fold cross-validation test. Identical results were achieved using 5-fold cross-validation, demonstrating the proposed model's consistency.

VI. DISCUSSION AND COMPARISON OF RESULTS

The proposed models were tested using a variety of IoT network intrusion datasets. A comparison was made between the proposed anomaly detection models and the previously published benchmark methodologies in this field based on the outcomes of their evaluations. LSTM, BiLSTM, GRU, CNNLSTM, CNNBiLSTM, and CNNGRU were used to design multiclass classification models. These models were evaluated using NSLKDD, BoT-IoT, IoT-NI, IoT-23, MQTT, MQTTset, and IoT-DS2 datasets. LSTM, BiLSTM, and GRU models with four hidden layers were initially used. The proposed models performed significantly better in multiclass and binary classification while detecting anomalies in IoT networks. The prospect of using a recurrent neural network and merging it with a convolutional neural network to solve anomaly detection in IoT networks was investigated.

The proposed model also uses kernel, bias, and activity regularizers at the RNN and CNN layers. These regularizers apply penalties on the kernel, bias, activity, and regularization layer output. The activation layer used LeakyReLU activation function. The utilization of the LeakyReLU activation function resulted in a significant acceleration of the learning process. A model normalization layer can often aid in accelerating and stabilizing the learning process by decreasing the error rate in the model being learned. Layer normalization can stabilize the hidden state dynamics of the recurrent network. The activity regularization layer and the dropout layer were used to decrease the possibility of overfitting. A binary classification model was designed utilizing a single hidden layer recurrent neural network. Table 18 compares binary classification techniques used in deep learning with proposed anomaly detection models. The NSLKDD dataset was used to compare the evaluation results of the proposed binary classification model for anomaly detection in a generic network setting. The detection rate was very high for IoT-23, MQTT, and MQTTset datasets. Given that these datasets are recently released datasets for the IoT network, there are few research papers in which authors constructed a deep learning model. Since BoT-IoT is the most cited IoT dataset, it is also used to compare proposed models to previously published IoT models.

Anomaly detection models based on RNN were evaluated using the NSLKDD dataset by Yin *et al.* [19]. The model achieved a higher level of accuracy in their experiment when they utilize 80 hidden nodes and a learning rate of 0.1. RNN-IDS had a high detection rate of 83.28% when 100 epochs were assigned. Liu *et al.* [37] proposed long and short session features and developed a neural network based on CNN and LSTM models to extract the variations between normal and abnormal models. The results demonstrate that the proposed quantitative model can effectively prevent hiding identity information. However, their model also increases computing efficiency and anomaly detection accuracy for small subsets of features. Their model got a 98.90% accuracy rate.

Li *et al.* [39] use LSTM and GRU with a variable number of hidden layers to evaluate the recently suggested Broad Learning System (BLS). Pseudoinverse weight adjustment replaces backpropagation in BLS models with fewer hidden layers. In the case of incremental learning, weights can be dynamically updated. The accuracy and F1 score were very low for LSTM and GRU models. Imrana *et al.* [9]

TABLE 17. NSLKDD, BoT-IoT, IoT-NI, IoT-23, MQTT, and MQTTset datasets class balancing using SMOTE and multiclass classification utilizing CNNLSTM, CNNBiLSTM, and CNNGRU models.

Model	CNNLSTM				CNNBiLSTM				CNNGRU			
Class	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
(a) NSLKDD dataset class balancing using SMOTE and multiclass classification utilizing CNNLSTM, CNNBiLSTM, and CNNGRU models.												
Normal	99.91	99.94	99.95	99.95	99.94	99.95	99.96	99.96	99.92	99.95	99.96	99.96
DoS	99.91	99.93	99.98	99.96	99.94	99.90	99.97	99.94	99.92	99.91	99.96	99.94
Probe	99.91	99.64	99.41	99.52	99.94	99.49	99.37	99.43	99.92	99.37	99.75	99.56
R2L	99.91	97.66	94.14	95.87	99.94	97.54	91.24	94.29	99.92	97.06	90.83	93.84
U2R	99.91	99.04	98.10	98.57	99.94	98.70	98.27	98.48	99.92	99.11	98.67	98.89
(b) BoT-IoT dataset class balancing using SMOTE and multiclass classification utilizing CNNLSTM, CNNBiLSTM, and CNNGRU models.												
Normal	99.94	99.96	99.87	99.91	99.97	99.98	99.90	99.94	99.96	99.96	99.85	99.91
DoS	99.94	99.93	99.93	99.93	99.97	99.96	99.99	99.98	99.96	99.95	99.97	99.96
Scan	99.94	99.93	99.93	99.93	99.97	99.97	99.95	99.96	99.96	99.95	99.96	99.95
Theft	99.94	99.94	99.96	99.95	99.97	99.97	99.96	99.96	99.96	99.97	99.95	99.96
(c) IoT-NI dataset class balancing using SMOTE and multiclass classification utilizing CNNLSTM, CNNBiLSTM, and CNNGRU models.												
Normal	98.88	99.63	97.66	98.64	99.40	99.38	99.22	99.30	98.90	99.59	97.69	98.63
DoS	98.88	99.91	99.39	99.65	99.40	99.92	99.61	99.77	98.90	99.90	99.36	99.63
Mirai	98.88	98.84	99.67	99.25	99.40	99.70	99.64	99.67	98.90	98.86	99.68	99.27
MITM	98.88	97.39	97.24	97.32	99.40	97.54	97.60	97.57	98.90	97.50	97.17	97.33
Scan	98.88	98.54	95.80	97.15	99.40	98.25	98.91	98.58	98.90	98.56	95.95	97.23
(d) IoT-23 dataset class balancing using SMOTE and multiclass classification utilizing CNNLSTM, CNNBiLSTM, and CNNGRU models.												
Normal	99.86	99.74	99.89	99.81	99.89	99.86	99.92	99.89	99.88	99.74	99.95	99.85
DDoS	99.86	99.80	99.39	99.59	99.89	99.73	99.94	99.84	99.88	99.83	99.39	99.61
Attack	99.86	98.50	99.99	99.24	99.89	99.17	90.84	94.82	99.88	98.50	99.24	98.87
Mirai	99.86	98.63	97.84	98.23	99.89	98.58	98.58	98.58	99.88	99.68	97.90	98.78
FileDownload	99.86	96.48	96.93	96.71	99.89	98.50	93.09	95.72	99.88	98.64	97.39	98.01
Heartbeat	99.86	99.07	98.63	98.85	99.89	99.08	98.53	98.81	99.88	99.64	98.73	99.18
C&C	99.86	99.84	99.86	99.85	99.89	99.90	99.86	99.88	99.88	99.84	99.84	99.84
Torii	99.86	99.99	99.99	99.99	99.89	99.99	99.99	99.99	99.88	99.99	99.99	99.99
Port Scan	99.86	99.99	99.99	99.99	99.89	99.99	99.99	99.99	99.88	99.99	99.99	99.99
Okiru	99.86	99.99	99.99	99.99	99.89	99.93	99.99	99.97	99.88	99.95	99.99	99.97
(e) MQTT dataset class balancing using SMOTE and multiclass classification utilizing CNNLSTM, CNNBiLSTM, and CNNGRU models.												
Normal	99.97	99.49	99.99	99.74	99.97	99.52	99.99	99.75	99.96	99.35	99.97	99.66
MQTT-BF	99.97	99.99	99.96	99.98	99.97	99.99	99.96	99.98	99.96	99.99	99.95	99.97
Scan-A	99.97	99.94	99.92	99.93	99.97	99.97	99.90	99.94	99.96	99.92	99.81	99.86
Scan-U	99.97	99.94	99.62	99.78	99.97	99.97	99.72	99.84	99.96	99.84	99.87	99.85
Sparta	99.97	99.99	99.99	99.99	99.97	99.99	99.99	99.99	99.96	99.99	99.99	99.99
(f) MQTTset dataset class balancing using SMOTE and multiclass classification utilizing CNNLSTM, CNNBiLSTM, and CNNGRU models.												
Normal	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99
MQTT Flood	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99
DoS	99.99	99.96	99.69	99.83	99.99	99.91	99.87	99.89	99.99	99.83	99.83	99.83
MQTT-BF	99.99	99.45	99.89	99.67	99.99	99.89	98.90	99.39	99.99	99.78	99.56	99.67
Malformed Data	99.99	99.71	99.99	99.85	99.99	98.57	99.99	99.28	99.99	99.56	99.56	99.56
SlowITe	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99

TABLE 18. **Comparison of binary classification models using deep learning for anomaly detection.**

Article	Model	Dataset	Accuracy	Precision	Recall	F1 Score	FPR
Yin *et al.* [19]	RNN	NSLKDD	71.35	86.64	83.28	83.28	13.40
Arivud *et al.* [33]	CNN	NSLKDD	99.67	99.69	-	-	0.57
Liu *et al.* [37]	CNN-LSTM	NSLKDD	98.90	-	-	-	-
Li *et al.* [39]	LSTM	NSLKDD	82.78	-	-	83.34	-
Li *et al.* [39]	GRU	NSLKDD	82.87	-	-	83.05	-
Imrana *et al.* [9]	LSTM	NSLKDD	89.81	97.75	84.03	90.38	2.55
Imrana *et al.* [9]	BiLSTM	NSLKDD	94.26	99.05	90.79	94.75	1.15
Biswas *et al.* [57]	ANN	NSLKDD	98.39	-	-	-	-
Biswas *et al.* [57]	GRU	NSLKDD	99.14	-	-	-	-
Liu *et al.* [98]	DNN	NSLKDD	93.20	-	-	91.10	0.09
Liu *et al.* [98]	CNN	NSLKDD	92.70	-	-	89.20	0.10
Liu *et al.* [98]	RNN	NSLKDD	91.90	-	-	88.10	0.12
Liu *et al.* [98]	MTDL	NSLKDD	95.50	-	-	94.10	0.079
Moizuddin *et al.* [99]	GMGWO	NSLKDD	99.84	99.68	99.94	99.81	-
Ferrag *et al.* [32]	RNN -BPTT	BoT-IoT	98.20	-	-	-	1.28
Susilo *et al.* [46]	MLP	BoT-IoT	79.01	-	-	-	
Susilo *et al.* [46]	CNN	BoT-IoT	91.27	-	-	-	-
Ferrag *et al.* [48]	RNN	BoT-IoT	98.31	97.50	-	-	1.1
Biswas *et al.* [57]	ANN	BoT-IoT	97.23	-	-	-	-
Biswas *et al.* [57]	GRU	BoT-IoT	99.76	-	-		
Liu *et al.* [98]	DNN	BoT-IoT	99.80	-	-	97.80	0.33
Liu *et al.* [98]	CNN	BoT-IoT	99.10	-	-	98.90	0.30
Liu *et al.* [98]	RNN	BoT-IoT	99.90	-	-	99.70	0.30
Liu *et al.* [98]	MTDL	BoT-IoT	99.90	-	-	99.80	0.21
Ge et al. [47]	DNN	BoT-IoT	99.90	-	-	-	-
Proposed Models	LSTM	NSLKDD	99.91	99.94	99.92	99.93	0.09
	GRU	NSLKDD	99.91	99.95	99.96	99.96	0.09
	BiLSTM	NSLKDD	99.92	99.96	99.95	99.96	0.07
	LSTM	BoT-IoT	99.90	99.95	99.95	99.95	0.52
	GRU	BoT-IoT	99.93	99.97	99.97	99.97	0.37
	BiLSTM	BoT-IoT	99.96	99.98	99.98	99.98	0.16

proposed a BiLSTM based intrusion detection system. BiLSTM and LSTM model's performance were compared using the NSLKDD dataset. The LSTM model has high false alarms than the BiLSTM model. Biswas *et al.* [57] proposed ANN and GRU models for distinguishing malicious botnet traffic from legitimate network traffic. The GRU model has a higher detection rate than the ANN model. Liu *et al.* [98] propose leveraging different aspects of each type of communication from three perspectives: anomaly discovery, clustering, and classification. Additionally, they offer a tailored loss function to adjust for traffic data that is distributed irregularly. DNN, CNN, and RNN models achieved a low detection rate compared to their proposed MTDL model.

The detection rate was high for IoT-23, MQTT, and MQTTset datasets. These datasets are recently released datasets for the IoT network, and there are few published research papers in which researchers developed a deep learning model using these datasets. Since the BoT-IoT dataset is the most referenced IoT dataset, the proposed binary classification models were also compared against previously published models using the BoT-IoT dataset. Ferrag *et al.* [32] came up with a new way to protect smart grids from hackers. They used deep learning and blockchain technology to build a new framework for smart grids. They used a recurrent neural network to monitor the energy network for network threats and fraudulent transactions. Their model achieved a high accuracy of 98.20% when applied to the BoT-IoT dataset.

Ferrag *et al.* [48] use a binary and multiclass classification approach to study seven deep learning models. RNN models with different learning rates and hidden nodes were tested for accuracy and training time using the BoT-IoT dataset. The accuracy and training time of unsupervised models with different learning rates and hidden nodes were also evaluated using the BoT-IoT dataset. The RNN model achieved a high accuracy of 98.31%. Susilo *et al.* [46] created a deep learning method for identifying DoS attacks in IoT networks. They

TABLE 19. Comparison of multiclass classification models using deep learning for anomaly detection.

Article	Model	Dataset	Accuracy	Precision	Recall	F1 Score	FPR
Yin *et al.* [19]	RNN	NSLKDD	81.29	83.06	81.29	79.25	13.00
Wu *et al.* [25]	CNN	NSLKDD	79.48	80.91	79.48	77.24	14.11
Naseer *et al.* [26]	LSTM	NSLKDD	89.00	-	-	-	-
Naseer *et al.* [26]	CNN	NSLKDD	85.00	-	-	-	-
Ding *et al.* [27]	LSTM	NSLKDD	73.18	-	-	-	-
Ding *et al.* [27]	CNN	NSLKDD	80.13	-	-	-	-
Xu *et al.* [24]	BGRU	NSLKDD	99.24	99.31	-	-	0.84
Chouhan *et al.* [34]	CNN	NSLKDD	89.41	-	-	-	-
Imrana *et al.* [9]	BiLSTM	NSLKDD	91.36	92.81	91.36	91.67	4.03
Liu *et al.* [51]	DSSTE+LSTM	NSLKDD	81.78	82.71	81.77	80.98	
Sethi *et al.* [61]	Reinforcement	NSLKDD	-	96.50	99.10	98.80	0.0124
Vinayakumar *et al.* [35]	DNN	NSLKDD	78.50	81.00	78.50	76.50	-
Moizuddin *et al.* [99]	GMGWO	NSLKDD	99.90	99.06	99.79	99.41	-
Sahu *et al.* [100]	LSTM	NSLKDD	98.93	98.90	99.10	99.00	-
Ge *et al.* [41]	DNN	BoT-IoT	99.03	99.06	99.03	99.04	0.73
Ge *et al.* [47]	DNN	BoT-IoT	99.79	99.80	99.78	99.79	0.11
Proposed Models	LSTM	NSLKDD	99.91	99.94	99.92	99.93	0.06
	GRU	NSLKDD	99.92	99.93	99.92	99.92	0.06
	BiLSTM	NSLKDD	99.94	99.96	99.94	99.95	0.05
	LSTM	BoT-IoT	99.94	99.97	99.94	99.95	0.03
	GRU	BoT-IoT	99.96	99.96	99.93	99.94	0.02
	BiLSTM	BoT-IoT	99.97	99.97	99.95	99.96	0.02

concluded that a deep learning model could enhance accuracy significantly, allowing for the most effective threat prevention in IoT networks. The accuracy of their proposed CNN model was very low, as seen in Table 18.

Advances in communication and information technology have enabled an increased amount of data to be exchanged via the Internet, resulting in new applications for Internet services. Biswas *et al.* [57] suggested a method for distinguishing malicious botnet traffic from legitimate traffic by using novel deep learning techniques such as ANN, GRU, and LSTM models. Testing reveals that the classification accuracy is 99.76%, which is better than all prior research, as indicated by the author. The proposed LSTM, BiLSTM, and GRU models outperformed previously published anomaly detection models in terms of accuracy, precision, recall, and F1 score using the BoT-IoT dataset.

Two anomaly detection models for multiclass classification in IoT networks were designed and implemented using LSTM, BiLSTM, and GRU networks. The LSTM, BiLSTM, and GRU models were built using the same structure. Figure 7 represents a layered view of the proposed LSTM, BiLSTM, or GRU models. There are four hidden layers in this implementation. The hidden layers may be LSTM, BiLSTM, or GRU. Convolutional neural network advantages become more apparent when anomaly detection problems are turned into image recognition problems. A convolutional neural network is excellent at capturing spatial and temporal correlations, which are particularly important for intrusion detection. A hybrid deep learning model for IoT networks was built using convolutional and recurrent neural networks. Three hidden layers were used in this technique to achieve the desired result. Figure 8 represents a layered view of the proposed CNN based LSTM, BiLSTM, or GRU models. There are three hidden layers in this implementation. The first hidden layer uses a convolutional neural network to extract feature information from the input features. An average pooling layer was utilized to decrease the number of features by half. The CNN layer is followed by two hidden layers of the recurrent neural network. These two hidden layers may be LSTM, BiLSTM, or GRU. Yin *et al.* [19], Wu *et al.* [25], Naseer *et al.* [26], Ding *et al.* [27], Chouhan *et al.* [34], Imrana *et al.* [9], Liu *et al.* [51], Sethi *et al.* [61], and Vinayakumar *et al.* [35] used the NSLKDD dataset to evaluate their model, but these techniques achieved very low accuracy, as shown in Table 19. The model proposed by Moizuddin *et al.* [99] and Sahu *et al.* [100] achieved reasonably high accuracy and detection rate compared to other model models.

Xu *et al.* [24] investigate a multilayer perception and gated recurrent units intrusion detection model using the KDD99 and NSLKDD datasets. Their model achieved a maximum accuracy of 99.24% for the NSLKDD dataset. Numerous issues emerge because malicious cyberattacks constantly evolve and happen in extremely high numbers, demanding a scalable solution. Due to malware's dynamic nature and ever changing attack methodologies, publicly accessible malware datasets must be regularly updated and benchmarked. DNNs and other conventional machine learning classifiers have been tested on various publicly accessible malware datasets by Vinayakumar *et al.* [35]. Their proposed model achieved a maximum accuracy of 92.90% for the KDD99 dataset using four hidden layers and 78.50% for the NSLKDD dataset using five hidden layers.

The security of an IoT network is essentially dependent on protecting the supporting communication infrastructure. Ge *et al.* [41] propose an intrusion detection technique for IoT networks, classifying IoT traffic flow using a deep learning technique. They use the BoT-IoT dataset to obtain generic features from packet data. Their model achieved 99.03% accuracy. Recently, Ge *et al.* [47] also proposed a model based on a feed forward neural network with embedding layers for encoding high dimensional categorical features for multiclass classification. They also used transfer learning to encode high dimensional category features to construct a binary classifier based on another feed forward neural network model. Their model reached a 99.79% accuracy rate. They used the source port in the feature set. All the attacks were generated from specific source ports. Due to this fact, there is a possibility that their model overfits the training data, resulting in high accuracy and detection rate. In terms of accuracy, precision, recall, and F1 score on the BoT-IoT dataset, the proposed LSTM, BiLSTM, and GRU models outperformed previously published anomaly detection techniques.

If an unbalanced dataset is used, the classification algorithm strongly leans toward one of the classes containing the majority of the data. When it comes to classification, if certain classes dominate, it can lead to biased results. As a result, it is suggested to balance the dataset. Oversampling technique were used to balance the datasets to solve this issue. When using SMOTE, synthetic samples are generated for the minority class; however, the methodologies used in SMOTE are based on local knowledge rather than generalized information about the minority class. The proposed models effectively capture anomaly detection problems spatial and temporal connectivity. IoT networks are comprised of a diverse range of applications and data types. The proposed methodology can be applied to a wide range of IoT applications and data to detect and investigate anomalies. Some IoT networks generate a large quantity of data due to their continued operation. The proposed model can deal with a large amount of data. Moreover, the proposed model achieves better performance when dealing with large volumes of data. A limitation of the proposed model is that it requires a significant volume of data to outperform other techniques.

VII. CONCLUSION AND FUTURE WORK

RNNs are suited to evaluate sequential data that is periodic in nature. RNN model can recognize and utilize the temporal context for sequential data, including repeating patterns. A novel deep learning model based on recurrent neural networks has been designed for detecting anomalies in IoT networks. The proposed model incorporates LSTM, BiLSTM, and GRU approaches to build a structure for anomalous activity analysis for intrusion detection in IoT networks. Convolutional neural networks are specifically well suited for feature learning because they can examine input features without losing essential information. A hybrid deep learning model has been designed to combine convolutional and recurrent neural networks. Finally, a lightweight deep learning model for binary classification that incorporates LSTM, BiLSTM, and GRU approaches has been designed. Seven datasets were used to evaluate the proposed models. In comparison to previous deep learning implementations, proposed multiclass and binary classification models achieved high accuracy, precision, recall, and F1 score. The proposed model enhanced the learning of weak features by utilizing an activity regularization layer; as a result, the model produced more balanced learning. The proposed hybrid model used a convolutional layer before recurrent layers to improve feature learning. The reliability of the proposed architecture for anomaly detection in IoT networks is demonstrated by the consistent performance of multiclass and binary classification models across several datasets.

In future work, we plan to investigate more deep learning approaches for anomaly detection in IoT networks, adopting various optimization techniques to boost the detection capability of these models on small datasets. We also plan to develop and evaluate ensemble techniques for LSTM, BiLSTM, and GRU models.

REFERENCES

[1] D. Li, L. Deng, M. Lee, and H. Wang, "IoT data feature extraction and intrusion detection system for smart cities based on deep migration learning," *Int. J. Inf. Manage.*, vol. 49, pp. 533–545, Oct. 2019, doi: 10.1016/j.ijinfomgt.2019.04.006.

[2] Shekharsaxena. *5 Layer Architecture of Internet of Things*. Accessed: Nov. 12, 2021. [Online]. Available: https://www.geeksforgeeks.org/5-layer-architecture-of-internet-of-things

[3] P. Sethi and S. R. Sarangi, "Internet of Things: Architectures, protocols, and applications," *J. Electr. Comput. Eng.*, vol. 2017, pp. 1–25, Jan. 2017, doi: 10.1155/2017/9324035.

[4] A. Sarangam. *7 IoT Layers That You Should Know in 2021*. Accessed: Nov. 10, 2021. [Online]. Available: https://www.jigsawacademy.com/4-layers-of-the-internet-of-things

[5] T. Vaiyapuri, Z. Sbai, H. Alaskar, and N. Ali, "Deep learning approaches for intrusion detection in IIoT networks—Opportunities and future directions," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 4, pp. 1–7, 2021, doi: 10.14569/IJACSA.2021.0120411.

[6] H. Hou, Y. Xu, M. Chen, Z. Liu, W. Guo, M. Gao, Y. Xin, and L. Cui, "Hierarchical long short-term memory network for cyberattack detection," *IEEE Access*, vol. 8, pp. 90907–90913, 2020, doi: 10.1109/ACCESS.2020.2983953.

[7] M. Al-Hawawreh, N. Moustafa, S. Garg, and M. S. Hossain, "Deep learning-enabled threat intelligence scheme in the Internet of Things networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 4, pp. 2968–2981, Oct. 2021, doi: 10.1109/TNSE.2020.3032415.

[8] S. Tsimenidis, T. Lagkas, and K. Rantos, "Deep learning in IoT intrusion detection," *J. Netw. Syst. Manage.*, vol. 30, no. 1, pp. 1–40, Jan. 2022, doi: 10.1007/s10922-021-09621-9.

[9] Y. Imrana, Y. Xiang, L. Ali, and Z. Abdul-Rauf, "A bidirectional LSTM deep learning approach for intrusion detection," *Expert Syst. Appl.*, vol. 185, Dec. 2021, Art. no. 115524, doi: 10.1016/j.eswa.2021.115524.

[10] M. M. Salim, S. K. Singh, and J. H. Park, "Securing smart cities using LSTM algorithm and lightweight containers against botnet attacks," *Appl. Soft Comput.*, vol. 113, Dec. 2021, Art. no. 107859, doi: 10.1016/j.asoc.2021.107859.

[11] H. Alkahtani and T. H. H. Aldhyani, "Botnet attack detection by using CNN-LSTM model for Internet of Things applications," *Secur. Commun. Netw.*, vol. 2021, pp. 1–23, Sep. 2021, doi: 10.1155/2021/3806459.

[12] 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.*, early access, Aug. 20, 2021, doi: 10.1109/TITS.2021.3105834.

[13] S. I. Popoola, B. Adebisi, M. Hammoudeh, H. Gacanin, and G. Gui, "Stacked recurrent neural network for botnet detection in smart Homes," *Comput. Electr. Eng.*, vol. 92, Jun. 2021, Art. no. 107039, doi: 10.1016/j.compeleceng.2021.107039.

[14] A. K. Desta, S. Ohira, I. Arai, and K. Fujikawa, "ID sequence analysis for intrusion detection in the CAN bus using long short term memory networks," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2020, pp. 1–6, doi: 10.1109/PerComWorkshops48775.2020.9156250.

[15] B. Wang, Y. Su, M. Zhang, and J. Nie, "A deep hierarchical network for packet-level malicious traffic detection," *IEEE Access*, vol. 8, pp. 201728–201740, 2020, doi: 10.1109/ACCESS.2020.3035967.

[16] Y. Hao, Y. Sheng, and J. Wang, "Variant gated recurrent units with encoders to preprocess packets for payload-aware intrusion detection," *IEEE Access*, vol. 7, pp. 49985–49998, 2019, doi: 10.1109/ACCESS.2019.2910860.

[17] Q. Wang, W. Zhao, and J. Ren, "Intrusion detection algorithm based on image enhanced convolutional neural network," *J. Intell. Fuzzy Syst.*, vol. 41, no. 1, pp. 2183–2194, Aug. 2021, doi: 10.3233/JIFS-210863.

[18] W. Wang, Y. Sheng, J. Wang, X. Zeng, X Ye, Y. Huang, and M. Zhu, "HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection," *IEEE Access*, vol. 6, pp. 1792–1806, 2018, doi: 10.1109/ACCESS.2017.2780250.

[19] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017, doi: 10.1109/ACCESS.2017.2762418.

[20] F. Meng, Y. Fu, F. Lou, and Z. Chen, "An effective network attack detection method based on kernel PCA and LSTM-RNN," in *Proc. Int. Conf. Comput. Syst., Electron. Control (ICCSEC)*, Dec. 2017, pp. 568–572, doi: 10.1109/ICCSEC.2017.8447022.

[21] A. Diro and N. Chilamkurti, "Leveraging LSTM networks for attack detection in fog-to-things communications," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 124–130, Sep. 2018, doi: 10.1109/MCOM.2018.1701270.

[22] B. Roy and H. Cheung, "A deep learning approach for intrusion detection in Internet of Things using bi-directional long short-term memory recurrent neural network," in *Proc. 28th Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, Nov. 2018, pp. 1–6, doi: 10.1109/ATNAC.2018.8615294.

[23] Y. Yang, K. Zheng, C. Wu, X. Niu, and Y. Yang, "Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks," *Appl. Sci.*, vol. 9, no. 2, p. 238, Jan. 2019, doi: 10.3390/app9020238.

[24] C. Xu, J. Shen, X. Du, and F. Zhang, "An intrusion detection system using a deep neural network with gated recurrent units," *IEEE Access*, vol. 6, pp. 48697–48707, 2018, doi: 10.1109/ACCESS.2018.2867564.

[25] K. Wu, Z. Chen, and W. Li, "A novel intrusion detection model for a massive network using convolutional neural networks," *IEEE Access*, vol. 6, pp. 50850–50859, 2018, doi: 10.1109/ACCESS.2018.2868993.

[26] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, and K. Han, "Enhanced network anomaly detection based on deep neural networks," *IEEE Access*, vol. 6, pp. 48231–48246, 2018, doi: 10.1109/ACCESS.2018.2863036.

[27] Y. Ding and Y. Zhai, "Intrusion detection system for NSL-KDD dataset using convolutional neural networks," in *Proc. 2nd Int. Conf. Comput. Sci. Artif. Intell. (CSAI)*, 2018, pp. 81–85, doi: 10.1145/3297156.3297230.

[28] R.-H. Hwang, M.-C. Peng, V.-L. Nguyen, and Y.-L. Chang, "An LSTM-based deep learning approach for classifying malicious traffic at the packet level," *Appl. Sci.*, vol. 9, no. 16, p. 3414, Aug. 2019, doi: 10.3390/app9163414.

[29] H. Alaiz-Moreton, J. Aveleira-Mata, J. Ondicol-Garcia, A. L. Muñoz-Castañeda, I. García, and C. Benavides, "Multiclass classification procedure for detecting attacks on MQTT-IoT protocol," *Complexity*, vol. 2019, pp. 1–11, Apr. 2019, doi: 10.1155/2019/6516253.

[30] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "DÏoT: A federated self-learning anomaly detection system for IoT," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 756–767, doi: 10.1109/ICDCS.2019.00080.

[31] F. Li, A. Shinde, Y. Shi, J. Ye, X.-Y. Li, and W.-Z. Song, "System statistics learning-based iot security: Feasibility and suitability," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6396–6403, Aug. 2019, doi: 10.1109/JIOT.2019.2897063.

[32] M. A. Ferrag and L. Maglaras, "DeepCoin: A novel deep learning and blockchain-based energy exchange framework for smart grids," *IEEE Trans. Eng. Manage.*, vol. 67, no. 4, pp. 1285–1297, Nov. 2020, doi: 10.1109/TEM.2019.2922936.

[33] D. Arivudainambi, V. K. K. A, and S. S. Chakkaravarthy, "LION IDS: A meta-heuristics approach to detect DDoS attacks against software-defined networks," *Neural Comput. Appl.*, vol. 31, no. 5, pp. 1491–1501, May 2019, doi: 10.1007/s00521-018-3383-7.

[34] N. Chouhan, A. Khan, and H.-U.-R. Khan, "Network anomaly detection using channel boosted and residual learning based deep convolutional neural network," *Appl. Soft Comput.*, vol. 83, Oct. 2019, Art. no. 105612, doi: 10.1016/j.asoc.2019.105612.

[35] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.

[36] O. Faker and E. Dogdu, "Intrusion detection using big data and deep learning techniques," in *Proc. ACM Southeast Conf.*, Apr. 2019, pp. 86–93, doi: 10.1145/3299815.3314439.

[37] A. Liu and B. Sun, "An intrusion detection system based on a quantitative model of interaction mode between ports," *IEEE Access*, vol. 7, pp. 161725–161740, 2019, doi: 10.1109/ACCESS.2019.2951839.

[38] W. Anani and J. Samarabandu, "Comparison of recurrent neural network algorithms for intrusion detection based on predicting packet sequences," in *Proc. IEEE Can. Conf. Electr. Comput. Eng. (CCECE)*, May 2018, pp. 1–4, doi: 10.1109/CCECE.2018.8447793.

[39] Z. Li, A. L. G. Rios, G. Xu, and L. Trajkovic, "Machine learning techniques for classifying network anomalies and intrusions," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2019, pp. 1–5, doi: 10.1109/ISCAS.2019.8702583.

[40] A. N. Sokolov, S. K. Alabugin, and I. A. Pyatnitsky, "Traffic modeling by recurrent neural networks for intrusion detection in industrial control systems," in *Proc. Int. Conf. Ind. Eng., Appl. Manuf. (ICIEAM)*, Mar. 2019, pp. 1–5, doi: 10.1109/ICIEAM.2019.8742961.

[41] M. Ge, X. Fu, N. Syed, Z. Baig, G. Teo, and A. Robles-Kelly, "Deep learning-based intrusion detection for IoT networks," in *Proc. IEEE 24th Pacific Rim Int. Symp. Dependable Comput. (PRDC)*, Dec. 2019, pp. 256–25609, doi: 10.1109/PRDC47002.2019.00056.

[42] A. Kim, M. Park, and D. H. Lee, "AI-IDS: Application of deep learning to real-time web intrusion detection," *IEEE Access*, vol. 8, pp. 70245–70261, 2020, doi: 10.1109/ACCESS.2020.2986882.

[43] M. Roopak, G. Y. Tian, and J. Chambers, "An intrusion detection system against DDoS attacks in IoT networks," in *Proc. 10th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2020, pp. 0562–0567, doi: 10.1109/CCWC47524.2020.9031206.

[44] F. Jiang, Y. Fu, B. B. Gupta, Y. Liang, S. Rho, F. Lou, F. Meng, and Z. Tian, "Deep learning based multi-channel intelligent attack detection for data security," *IEEE Trans. Sustain. Comput.*, vol. 5, no. 2, pp. 204–212, Apr. 2020, doi: 10.1109/TSUSC.2018.2793284.

[45] J. Malik, A. Akhunzada, I. Bibi, M. Imran, A. Musaddiq, and S. W. Kim, "Hybrid deep learning: An efficient reconnaissance and surveillance detection mechanism in SDN," *IEEE Access*, vol. 8, pp. 134695–134706, 2020, doi: 10.1109/ACCESS.2020.3009849.

[46] B. Susilo and R. F. Sari, "Intrusion detection in IoT networks using deep learning algorithm," *Information*, vol. 11, no. 5, p. 279, May 2020, doi: 10.3390/info11050279.

[47] M. Ge, N. F. Syed, X. Fu, Z. Baig, and A. Robles-Kelly, "Towards a deep learning-driven intrusion detection approach for Internet of Things," *Comput. Netw.*, vol. 186, Feb. 2021, Art. no. 107784, doi: 10.1016/j.comnet.2020.107784.

[48] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *J. Inf. Secur. Appl.*, vol. 50, Feb. 2020, Art. no. 102419, doi: 10.1016/j.jisa.2019.102419.

[49] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, and G. Fortino, "A hybrid deep learning model for efficient intrusion detection in big data environment," *Inf. Sci.*, vol. 513, pp. 386–396, Mar. 2020, doi: 10.1016/j.ins.2019.10.069.

[50] S. Aldhaheri, D. Alghazzawi, L. Cheng, B. Alzahrani, and A. Al-Barakati, "DeepDCA: Novel network-based detection of IoT attacks using artificial immune system," *Appl. Sci.*, vol. 10, no. 6, p. 1909, Mar. 2020, doi: 10.3390/app10061909.

[51] L. Liu, P. Wang, J. Lin, and L. Liu, "Intrusion detection of imbalanced network traffic based on machine learning and deep learning," *IEEE Access*, vol. 9, pp. 7550–7563, 2021, doi: 10.1109/ACCESS.2020.3048198.

[52] J. Ashraf, A. D. Bakhshi, N. Moustafa, H. Khurshid, A. Javed, and A. Beheshti, "Novel deep learning-enabled LSTM autoencoder architecture for discovering anomalous events from intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4507–4518, Jul. 2021, doi: 10.1109/TITS.2020.3017882.

[53] B. B. Borisenko, S. D. Erokhin, A. S. Fadeev, and I. D. Martishin, "Intrusion detection using multilayer perceptron and neural networks with long short-term memory," in *Proc. Syst. Signal Synchronization, Generating Process. Telecommun. (SYNCHROINFO*, Jun. 2021, pp. 1–6, doi: 10.1109/SYNCHROINFO51390.2021.9488416.

[54] T. H. Hai and L. H. Nam, "A practical comparison of deep learning methods for network intrusion detection," in *Proc. Int. Conf. Electr., Commun., Comput. Eng. (ICECCE)*, Jun. 2021, pp. 1–6, doi: 10.1109/ICECCE52056.2021.9514161.

[55] T. S. Pooja and P. Shrinivasacharya, "Evaluating neural networks using bi-directional LSTM for network IDS (intrusion detection systems) in cyber security," *Global Transitions Proc.*, vol. 2, no. 2, pp. 448–454, Nov. 2021, doi: 10.1016/j.gltp.2021.08.017.

[56] H. Jia, J. Liu, M. Zhang, X. He, and W. Sun, "Network intrusion detection based on IE-DBN model," *Comput. Commun.*, vol. 178, pp. 131–140, Oct. 2021, doi: 10.1016/j.comcom.2021.07.016.

[57] R. Biswas and S. Roy, "Botnet traffic identification using neural networks," *Multimedia Tools Appl.*, vol. 2021, pp. 24147–24171, Apr. 2021, doi: 10.1007/s11042-021-10765-8.

[58] 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, pp. 1–16, Dec. 2021, doi: 10.1186/s40537-021-00448-4.

[59] M. S. ElSayed, N.-A. Le-Khac, M. A. Albahar, and A. Jurcut, "A novel hybrid model for intrusion detection systems in SDNs based on CNN and a new regularization technique," *J. Netw. Comput. Appl.*, vol. 191, Oct. 2021, Art. no. 103160, doi: 10.1016/j.jnca.2021.103160.

[60] C. Joshi, R. K. Ranjan, and V. Bharti, "A fuzzy logic based feature engineering approach for botnet detection using ANN," *J. King Saud Univ. Comput. Inf. Sci.*, Jul. 2021, doi: 10.1016/j.jksuci.2021.06.018.

[61] K. Sethi, Y. V. Madhav, R. Kumar, and P. Bera, "Attention based multi-agent intrusion detection systems using reinforcement learning," *J. Inf. Secur. Appl.*, vol. 61, Sep. 2021, Art. no. 102923, doi: 10.1016/j.jisa.2021.102923.

[62] H. Alyasiri, J. A. Clark, A. Malik, and R. D. Frein, "Grammatical evolution for detecting cyberattacks in Internet of Things environments," in *Proc. Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2021, pp. 1–6, doi: 10.1109/ICCCN52240.2021.9522283.

[63] F. Hussain, S. G. Abbas, G. A. Shah, I. M. Pires, U. U. Fayyaz, F. Shahzad, N. M. Garcia, and E. Zdravevski, "A framework for malicious traffic detection in IoT healthcare environment," *Sensors*, vol. 21, no. 9, p. 3025, Apr. 2021, doi: 10.3390/s21093025.

[64] I. Vaccari, S. Narteni, M. Aiello, M. Mongelli, and E. Cambiaso, "Exploiting Internet of Things protocols for malicious data exfiltration activities," *IEEE Access*, vol. 9, pp. 104261–104280, 2021, doi: 10.1109/ACCESS.2021.3099642.

[65] T. T. Huong, T. P. Bac, D. M. Long, T. D. Luong, N. M. Dan, L. A. Quang, L. T. Cong, B. D. Thang, and K. P. Tran, "Detecting cyberattacks using anomaly detection in industrial control systems: A federated learning approach," *Comput. Ind.*, vol. 132, Nov. 2021, Art. no. 103509, doi: 10.1016/j.compind.2021.103509.

[66] S. Nayak, N. Ahmed, and S. Misra, "Deep learning-based reliable routing attack detection mechanism for industrial Internet of Things," *Ad Hoc Netw.*, vol. 123, Dec. 2021, Art. no. 102661, doi: 10.1016/j.adhoc.2021.102661.

[67] J. Ashraf, M. Keshk, N. Moustafa, M. Abdel-Basset, H. Khurshid, A. D. Bakhshi, and R. R. Mostafa, "IoTBoT-IDS: A novel statistical learning-enabled botnet detection framework for protecting networks of smart cities," *Sustain. Cities Soc.*, vol. 72, Sep. 2021, Art. no. 103041, doi: 10.1016/j.scs.2021.103041.

[68] M. H. L. Louk and B. A. Tama, "Exploring ensemble-based class imbalance learners for intrusion detection in industrial control networks," *Big Data Cognit. Comput.*, vol. 5, no. 4, p. 72, Dec. 2021, doi: 10.3390/bdcc5040072.

[69] L. Nkenyereye, B. A. Tama, and S. Lim, "A stacking-based deep neural network approach for effective network anomaly detection," *Comput., Mater. Continua*, vol. 66, no. 2, pp. 2217–2227, 2021, doi: 10.32604/cmc.2020.012432.

[70] I. Ullah and Q. H. Mahmoud, "A two-level hybrid model for anomalous activity detection in IoT networks," in *Proc. 16th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2019, pp. 1–6, doi: 10.1109/CCNC.2019.8651782.

[71] I. Ullah and Q. H. Mahmoud, "An intrusion detection framework for the smart grid," in *Proc. IEEE 30th Can. Conf. Electr. Comput. Eng. (CCECE)*, Apr. 2017, pp. 1–5, doi: 10.1109/CCECE.2017.7946654.

[72] I. Ullah and Q. H. Mahmoud, "A filter-based feature selection model for anomaly-based intrusion detection systems," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2017, pp. 2151–2159, doi: 10.1109/BigData.2017.8258163.

[73] I. Ullah and Q. H. Mahmoud, "A hybrid model for anomaly-based intrusion detection in SCADA networks," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2017, pp. 2160–2167, doi: 10.1109/BigData.2017.8258164.

[74] I. Ullah and Q. H. Mahmoud, "A two-level flow-based anomalous activity detection system for IoT networks," *Electronics*, vol. 9, no. 3, p. 530, Mar. 2020, doi: 10.3390/electronics9030530.

[75] I. Ullah and Q. H. Mahmoud, "Network traffic flow based machine learning technique for IoT device identification," in *Proc. IEEE Int. Syst. Conf. (SysCon)*, Apr. 2021, pp. 1–8, doi: 10.1109/SysCon48628.2021.9447099.

[76] I. Ullah and Q. H. Mahmoud, "An anomaly detection model for IoT networks based on flow and flag features using a feed-forward neural network," in *Proc. IEEE 19th Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2022, pp. 363–368, doi: 10.1109/CCNC49033.2022.9700597.

[77] I. Ullah and Q. H. Mahmoud, "A framework for anomaly detection in IoT networks using conditional generative adversarial networks," *IEEE Access*, vol. 9, pp. 165907–165931, 2021, doi: 10.1109/ACCESS.2021.3132127.

[78] Q. A. Al-Haija, A. Al Badawi, and G. R. Bojja, "Boost-defence for resilient IoT networks: A head-to-toe approach," *Expert Syst.*, vol. 2022, p. 12934, Jan. 2022, doi: 10.1111/exsy.12934.

[79] Q. A. Al-Haija and A. Al-Badawi, "Attack-aware IoT network traffic routing leveraging ensemble learning," *Sensors*, vol. 22, no. 1, p. 241, Dec. 2021, doi: 10.3390/s22010241.

[80] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[81] C. Olah. *Understanding LSTM Networks*. Accessed: Nov. 15, 2021. [Online]. Available: http://colah.github.io/posts /2015-08-Understanding-LSTMs/#fn1

[82] 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*.

[83] 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, doi: 10.1109/ACCESS.2021.3094024.

[84] D. Liu. (2017). *A Practical Guide to ReLU*. Accessed: Sep. 20, 2021. [Online]. Available: https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7

[85] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.

[86] S. Saha. (2018). *A Comprehensive Guide to Convolutional Neural Networks*. Accessed: Sep. 24, 2021. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[87] I. Ullah and Q. H. Mahmoud, ''A technique for generating a botnet dataset for anomalous activity detection in IoT networks,'' in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2020, pp. 134–140, doi: 10.1109/SMC42975.2020.9283220.

[88] I. Ullah and Q. H. Mahmoud, ''A scheme for generating a dataset for anomalous activity detection in IoT networks,'' in *Advances in Artificial Intelligence* (Lecture Notes in Computer Science), vol. 12109, C. Goutte and X. Zhu, Eds. Cham, Switzerland: Springer, 2020, pp. 508–520, doi: 10.1007/978-3-030-47358-7_52.

[89] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, ''A detailed analysis of the KDD CUP 99 data set,'' in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Jul. 2009, pp. 1–6, doi: 10.1109/CISDA.2009.5356528.

[90] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, ''Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset,'' *Future Gener. Comput. Syst.*, vol. 100, pp. 779–796, Nov. 2019, doi: 10.1016/j.future.2019.05.041.

[91] H. Kang, D. H. Ahn, G. M. Lee, J. D. Yoo, K. H. Park, and H. K. Kim, ''IoT network intrusion dataset,'' IEEE Dataport, Tech. Rep., doi: 10.21227/q70p-q449.

[92] A. Parmisano, S. Garcia, and M. J. Erquiaga, ''IoT-23: A labeled dataset with malicious and benign IoT network traffic,'' Stratos. Lab., Praha, Czech Republic, Tech. Rep., 2020, doi: 10.5281/zenodo.4743746.

[93] 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, doi: 10.1007/978-3-030-64758-2_6.

[94] I. Vaccari, G. Chiola, M. Aiello, M. Mongelli, and E. Cambiaso, ''MQTTset, a new dataset for machine learning techniques on MQTT,'' *Sensors*, vol. 20, no. 22, p. 6578, Nov. 2020, doi: 10.3390/s20226578.

[95] A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani, ''Characterization of tor traffic using time based features,'' in *Proc. 3rd Int. Conf. Inf. Syst. Secur. Privacy*, Sep. 2017, pp. 253–262, doi: 10.5220/0006105602530262.

[96] I. Ullah and H. Q. Mahmoud. (2021). *IoT Network Intrusion Datasets*. [Online]. Available: https://sites.google.com/view/iotdataset1

[97] S. Ghazanfar, F. Hussain, A. U. Rehman, U. U. Fayyaz, F. Shahzad, and G. A. Shah, ''IoT-flock: An open-source framework for IoT traffic generation,'' in *Proc. Int. Conf. Emerg. Trends Smart Technol. (ICETST)*, Mar. 2020, pp. 1–6, doi: 10.1109/ICETST49965.2020.9080732.

[98] Q. Liu, D. Wang, Y. Jia, S. Luo, and C. Wang, ''A multi-task based deep learning approach for intrusion detection,'' *Knowl.-Based Syst.*, vol. 238, Feb. 2022, Art. no. 107852, doi: 10.1016/j.knosys.2021.107852.

[99] M. Moizuddin and M. V. Jose, ''A bio-inspired hybrid deep learning model for network intrusion detection,'' *Knowl.-Based Syst.*, vol. 238, Feb. 2022, Art. no. 107894, doi: 10.1016/j.knosys.2021.107894.

[100] S. K. Sahu, D. P. Mohapatra, J. K. Rout, K. S. Sahoo, Q. Pham, and N. Dao, ''A LSTM-FCNN based multi-class intrusion detection using scalable,'' *Comput. Electr. Eng.*, vol. 99, Apr. 2022, Art. no. 107720, doi: 10.1016/j.compeleceng.2022.107720.

IMTIAZ ULLAH (Member, IEEE) received the M.Sc. degree in internet, computer, and system security from the Department of Computer Science, University of Bradford, U.K., the M.Sc. degree in computer science from the Department of Computer Science, University of Peshawar, Pakistan, and the Ph.D. degree from the Department of Electrical, Computer, and Software Engineering, Ontario Tech University, Oshawa, ON, Canada. His current research interests include deep learning and anomaly detection models for the Internet of Things.

QUSAY H. MAHMOUD (Senior Member, IEEE) was the Founding Chair at the Department of Electrical, Computer and Software Engineering, Ontario Tech University, Canada. He is currently a Professor of software engineering with the Department of Electrical, Computer and Software Engineering. His research interests include intelligent software systems and cybersecurity.