IEEE Transaction on Deep Learning,Volume:10,Issue Date:20.July.2022

A Hybrid Deep Learning Approach for Bottleneck Detection in IoT

FRAIDOON SATTARI[1], ASHFAQ HUSSAIN FAROOQI[2], ZAKRIA QADIR[3], BASIT RAZA[1], HADI NAZARI[1], AND MUHANNAD ALMUTIRY[4], (Member, IEEE)

[1]Department of Computer Science, COMSATS University Islamabad, Islamabad 45550, Pakistan
[2]Department of Computer Science, Air University, Islamabad 44000, Pakistan
[3]School of Computing Engineering and Mathematics, Western Sydney University, Penrith, NSW 2751, Australia
[4]Department of Electrical Engineering, Northern Border University, Arar 91431, Saudi Arabia

Corresponding author: Basit Raza (basit.raza@comsats.edu.pk)

The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number IF_2020_NBU_431.

ABSTRACT Cloud computing is perhaps the most enticing innovation in the present figuring situation. It gives an expense-effective arrangement by diminishing the enormous forthright expense of purchasing equipment foundations and processing power. Fog computing is an additional help to cloud infrastructure by utilizing a portion of the less-registered undertaking at the edge devices, reducing the end client's reaction time, such as IoT. However, most of the IoT devices are resource-constrained, and there are many devices that cyber attacks could target. Cyber-attacks such as bottleneck, Dos, DDoS, and botnets are still significant threats in the IoT environment. Botnets are currently the most significant threat on the internet. A set of infected systems connected online and directed by an adversary to carry out malicious actions without authorization or authentication is known as a botnet. A botnet can compromise the system and steal the data. It can also perform attacks, like Phishing, spamming, and more. To overcome the critical issue, we exhibit a novel botnet attack detection approach that could be utilized in fog computing situations to dispense with the attack using the programmable nature of the software-defined network (SDN) environment. We carefully tested the most recent dataset for our proposed technique, standard and extended performance evaluation measures, and current DL models. To further illustrate overall performance, our findings are cross-validated. The proposed method performs better than previous ones in correctly identifying 99.98% of multi-variant sophisticated bot attacks. Additionally, the time of our suggested method is 0.022(ms), indicating good speed efficiency results.

INDEX TERMS Fog security, software defined networks, deep learning, Internet of Things, botnet, intrusion detection.

I. INTRODUCTION

One of the most significant issues for the network system to be efficient and reliable while doing transactions over the IoT is security [1]. The tremendous growth of IoT in different fields, i.e., surveillance, healthcare, transportation, manufacturing industry, education, and others, encourages securing IoT infrastructure to improve its performance. Earlier IoT devices generate data through various types of sensors, and it becomes tidy for the cloud servers to handle or process these transactions efficiently. Fog computing is among the newly proposed schemes that could be utilized to add preferred features to the IoT infrastructure [2]. Fog computing is competent in doing some regional analysis of information [3] before communicating the aggregated data to the cloud server. It helps in keeping the latency constraints in some time compelled real-time issues, making them appropriate for IoT-based applications such as vehicular ad-hoc networks (VANETs) [4]–[11]. These advancements towards using fog servers in IoT infrastructure motivate the adversaries to target

The associate editor coordinating the review of this manuscript and approving it for publication was Ahmed M. Elmisery.

TABLE 1. List of acronyms in manuscript.

Notations	Explanation	Notations	Explanation
ML	Machine Learning	DL	Deep Learning
ROC	Receiver Operating Characteristic	TNR	True Negative Rate
DoS	Denial of Service	FNR	False Negative Rate
IoT	Internet of Things	RNN	Recurrent Neural Network
CNN	Convolutional Neural Network	FOR	False Omission Rate
DDoS	Distributed Denial of Service	IIoT	Industrial Internet of Things
DNN	Deep Neural Network	FPR	False Positive Rate
SDN	Software Defined Network	FDR	False Discovery Rate
NPV	Negative Predictive Value	MCC	Matthews Correlation Coefficient
LSTM	Long short-term Memory	TS	Threat Score

the fog server with malicious intent to lower its performance. Hence, security and protection of the system are among the major issues that can affect the performance of fog computing [12]. In this regard, availability is among the core security requirements for offering services to the actual customer applications according to their interest. However, this is constantly tested by the adversaries by launching different types of attacks, such as DoS or DDoS attacks [13]. An individual or a group can perform these attacks. If a group performs it, it is named "botnet," while if an individual launches it, it is known as "bot-master." [14]. The bot-master is the attacker node that can launch several types of attacks on the server, such as Phishing, spam, Click fraud, and others. A command-and-control channel remotely controls a botnet. The command-and-control channel is a system the adversary uses to control by sending messages and commands to a compromised system. The adversary can steal the data through these commands and manipulate the infected network [8]. In a botnet attack, some 'n' number of compromised nodes are controlled by a bot-master, and they launch an attack on the server from different compromised systems.

In the fog computing paradigm security is still challenging task, and various security schemes are proposed to make it resilient against vulnerabilities. However, most of the schemes focus on flexibility and continuous monitoring of the fog server. Software-defined networking (SDN) is used at fog servers to address flexibility, and continuous monitoring issues [15]. SDN is an emerging networking paradigm that assists in making the network more flexible that can help in managing the network, analyzing the traffic, and assisting in the routing control architectures [16], [17] as there is a separate control plan that provides a flexible device management policy. Hence, an SDN-based fog computing environment provides centralized control to the fog computing system. The characteristics of the SDN based fog computing system are discussed below:

- SDN can manage the secure connection for thousands of devices connected over the fog for data transmission.
- SDN can provide real-time monitoring and awareness with low latency.
- SDN can dynamically balance the load with its flexible architecture.
- SDN can customize the policies and applications dues to its programmable nature. [18].

The software-defined network plays a vital role as its network control architecture can be directly programmable through the command requests. SDN-based fog computing architecture can assist in analyzing and managing IoT devices. The motivation behind SDN is to give consistency to network management through partitioning the network into the data plane and the control plane. SDN can add programmability, adaptability, and versatility to the fog computing system. In high-speed networks, discovering the botnet attack is a significant concern [19]. The proposed work shows the methodology through which the botnet attack is identified with a high detection rate which can be used in SDN to enhance the security of fog computing. Deep learning (DL) based detection approach in the SDN-based fog computing application can be a better counterattack to improve the overall performance of the system [20]. DL strategy is adaptable to conditions to recognize the abnormal behavior of the network. We proposed a hybrid deep learning detection policy to improve the efficiency and effectiveness of the SDN-based fog computing architecture. Results show that the proposed scheme works better and provides a better detection rate.

A. RESEARCH CONTRIBUTIONS

The research contribution includes the comprehensive evaluation of botnet attacks for different IoT devices and evolving cyber threats in IoT using the dataset N_BaIoT 2018. Our proposed hybrid technique comprises two DL algorithms: DNN and LSTM. We rigorously evaluate the proposed mechanism with standard performance metrics (i.e., Recall, Accuracy, F1-Score, Precision, AU-ROC, etc.). The presented hybrid scheme results show better detection accuracy with low computational complexity. The contributions of this research work are as given below:

- We suggest an efficient deep learning framework for detecting Botnet attacks in an SDN-based fog computing environment.
- The practical experiment is performed on N_BaIoT Dataset, which comprises both Botnet attack and benign samples.

TABLE 2. Summary of existing work.

Ref	Year	Threats	Evaluation Metric	Dataset	Deep Learning	Machine Learning	Achievement
[21]	2017	Botnet Attack	TPR, FPR, TNR	CTU-13	—	Random Forest	93.6% Accuracy
[22]	2021	DNS-Based Botnet Detection	FPR, Precision, F1-Score	CTU-13	—	Hybrid Rule-based Model	99.96% Accuracy and 1.6% false positive rate
[23]	2018	IoT Botnets	Precision, Recall, and F1-Measure	—	—	Logistic Regression Model	97.30% Accuracy
[24]	2022	IoT Botnets	Recall, Precision, Accuracy	N_BaIoT 2018	—	ANN, K-NN, Ensemble tree, Fuzzy classifier	Tree-based algorithm achieved 99% accuracy
[25]	2022	IoT Botnets	Accuracy, F1-Score, Sensitivity	N_BaIoT 2018	—	RF and eXtreme Gradient Boosting (XGB-RF)	99.9426% Accuracy with error score of 0.06%
[34]	2021	IoT Botnets	Accuracy, Recall, F1-Score, Sensitivity	N_BaIoT 2018	CNN-LSTM	—	90.88% Accuracy
[35]	2020	DoS Attacks	Precision, Accuracy	BoT-IoT	Multilayer Perceptron, CNN	RF, SVM	CNN model achieved 91.27% Accuracy
[36]	2021	IoT Botnets	Accuracy, FPR, F-Measure, Recall	IoT-23	CNN-LSTM	—	96% Accuracy
[37]	2021	Intrusion Detection	Recall, F1-Score, and Precision	CICIDS2018	Cu-DNNGRU + Cu-BLSTM	—	Hybrid model achieved FPR of 0.0554% and accuracy of 99.87%
[38]	2018	Botnet DDoS Attacks	—	Self Generated	BLSTM-RNN	—	Hybrid model achieved 99% Accuracy
[39]	2018	DoS,R2L,U2R, probing Attacks	Recall, Specificity, Precision, FPR, F1-Score	KDD99	Restricted Boltzmann Machines (RBM)	—	Achieved precision higher than 94%
[40]	2019	Botnet DoS Attack	Precision, Recall, F1-Score	CTU-13 and ISOT	LSTM and CNN	—	99.3% Accuracy and 99.1% F1-Score

- The proposed technique is evaluated against well-known performance evaluation metrics of the machine and deep learning algorithms known as precision, F1-score, recall, accuracy, and so forth.
- For unbiased results, we also applied the technique of 10-fold-cross-validation.

The paper's organization is as follows; section II introduces related literature. Section III shows the security issues in Fog computing. Section IV provides information about Deep Learning and its algorithms. Section V details our proposed system and the methodology used for detection and experimentation, such as Dataset, detection phase, evaluation phase, and experiment. While Section VI comprises the experimental results and our assessment results. Finally, section VII provides the conclusion and defines the future map.

II. LITERATURE REVIEW

Security remained one of the top research areas in networking paradigms whether it is based on cloud computing [12], fog computing [3], IoT [1] or SCADA (Supervisory Control and Data Acquisition) systems [26], [27] or others. Several researchers are focusing on detecting botnet attacks these days [28]–[30]. The main requirement in botnet detection is identifying the infected devices before they can exploit the network by initiating malicious activity. Authors propose numerous methods that claim to secure the network against botnet attacks. These approaches focus on anomaly detection schemes using artificial intelligence, primarily ML and DL algorithms. In various research approaches, authors [21]–[23] used ML and hybrid ML techniques for botnet detection such as BayesNet (BN), Support Vector Machine (SVM), J48, Decision Tree (DT), and Naive Bayes (NB). Furthermore, Machine Learning methods are categorized as the supervised, the unsupervised, or the semi-supervised learning [24], [25], [31], [32]. Parakash *et al.* performed experiments using three well-known machine learning algorithms to detect DDoS packets: K-Nearest Neighbors algorithm (KNN), SVM, and NB. The findings show that the KNN performs better in detecting DDoS attacks having 97% accuracy, while SVM and NB algorithms achieve 82% and 83% accuracy, respectively [33]. In [34], the authors proposed a detection scheme that uses the SVM algorithm with their own proposed idle timeout adjustment algorithm (IA). They demonstrated the way their proposed methodology outperforms and achieves better results. In another work, [35] uses, NB, SVM and neural network. Results show that the neural network and NB models performed outclass and achieved 100% accuracy,

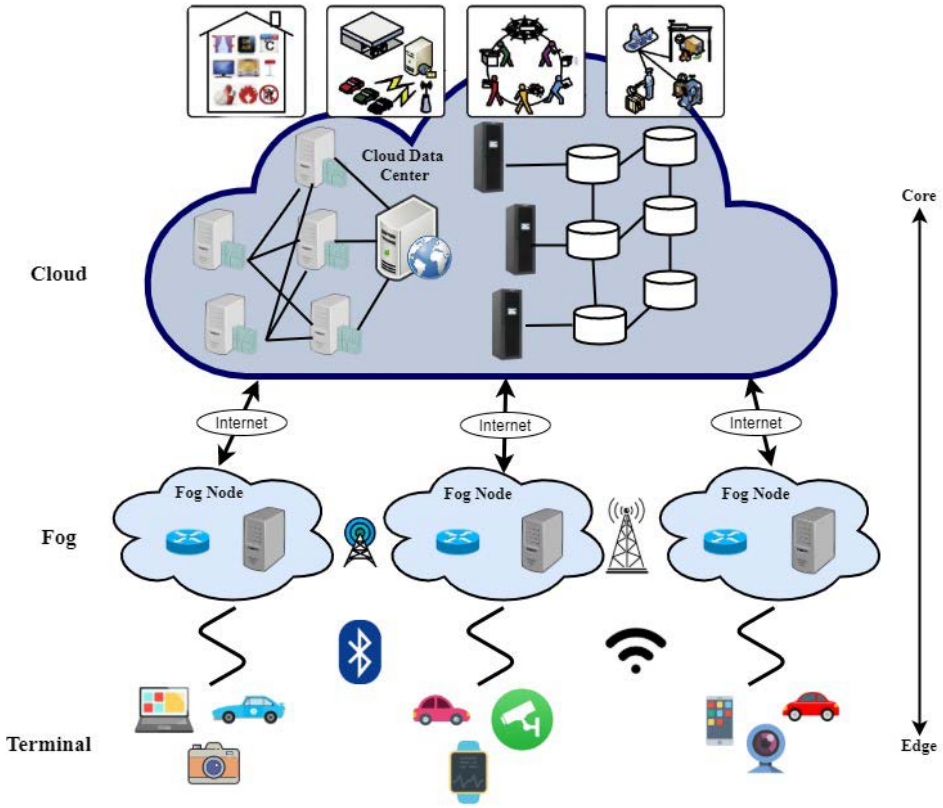

FIGURE 1. Fog computing and communications.

while the SVM model was at 95% accuracy. Ye *et al.* [36] also used the SVM algorithm and achieved an average accuracy of 95.24%. In [37], authors performed experiments using various algorithms such as Naive Bayesian and decision tree classifier algorithms. They achieved a 99.6% detection accuracy rate.

ML algorithms face challenges like scalability, learning from massive data, and low-value density data. To convert big data into usable intelligence in the face of an ever-growing big data universe, ML must develop and improve. Massive data is developing exponentially, so ML must develop and evolve to turn big data into valuable insight.

DL algorithms are the subset of ML. That can deal with large datasets and unstructured data. ML algorithms do not provide better results for extensive data produced by IoT devices and unstructured data [38]. Hence DL algorithms are preferable for IoT compared to traditional ML algorithms such as KNN, SVM, NB, and others. Different DL and hybrid DL approaches are applied for detecting various kinds of malware in IoT devices [39]–[41]. In [42], the authors described a technique for defending the IoT environment against malware and cyber attacks, such as DDoS, brute force, bot, and infiltration. This strategy makes use of DL in SDN. They used the CICIDS2018 dataset for the evaluation of the presented scheme. The proposed model achieved 99.87% accuracy, 0.0554% FPR, with a testing time of only 18.9ms. Likewise, in [43], using two-way LSTM to implement DL for evaluation demonstrates a new method for packet-level inspection on the IoT and networks. The authors utilized Mirai and normal IoT traffic generated in this paper for experiments.

Consequently, the authors of [44] used SDN to deploy an detection mechanism system to safeguard the IoT and showed a testing success rate with 95% accuracy. They considered the KDD99 dataset for attack detection using the Restricted Boltzmann Machine for DoS, login, and Probe (RBM). Moreover, in [45], authors considered a hybrid model consisting of CNN and RNN. The proposed solution is based on network flow attributes. They applied the proposed model to two datasets, i.e., CTU13 and ISOT. These combined datasets form two classes, i.e., botnets and benign. The author of [16] offered an IoT based work that acknowledges the effectiveness of a DL-based algorithm (LSTM) for botnet attack detection. The study used data from various IoT devices from the N IoT 2018 dataset, which had a 99.90% detection rate.

DL approaches are helpful for intrusion detection in SDN-based architectures [20], [46] [47]. DL methods are applied to identify botnet attacks in non-SDN infrastructures [48] while requiring more research to analyze the feasibility and efficacy of using DL (CNN, RNN, and LSTM) algorithms to detect and mitigate botnet attacks on SDN controllers.The studies show that to effectively defend the system against newly developing threats, a centralised mechanism

TABLE 3. Security issues in fog computing.

Attacks	Description
Spam	An unwanted message was developed and distributed by intruders. Spam is one of the most severe security threats since it can allow malware to propagate and waste resources.
DoS	To make the Fog nodes unavailable to real users, flood them with many bogus requests.
Man-in-the-middle	An attack in which an attacker is placed between two connecting parties to intercept and manipulate data passing between them.
Tampering	The attacker alters, delays or drops the data packets to reduce or disrupt the performance and efficiency of Fog computing.
Eavesdropping	Sniffing and spoofing attacks are other names for these types of attacks. Hackers take data from computers, smartphones, and other devices and send it across the internet without the users' permission.
Jamming	A DOS attack where one node blocks other nodes from interacting on the channel by occupying the channel they are communicating.
Forgery	The attackers use bogus information to fool victims by imitating their identities. Because of the bogus data packets, this attack reduces network performance using energy, storage, and bandwidth.
Sybil	An attack where one node steals the identity of another node to take control of and compromise Fog nodes to generate fraudulent sensing data while exposing the users' personal information.

and intelligence are still needed. The accuracy of botnet malware detection varies for different algorithms applied to different datasets.

In Table 2, several ML or DL based detection schemes are presented. It shows that the selected research area is among the emerging research trends in the field of IoT security. There is ongoing research in this area using different datasets [49]. As per our findings, a thorough study of the DL hybrid combinations is required to explore the possibility of increasing the accuracy and precision in the detection of botnet attacks such that it further achieves lower FPR in consuming less time. Hence, we tested various combinations of DL algorithms in our research work and concluded that the hybrid deep learning algorithm uses DNN [50] and LSTM [51] is effective. It also produces better outcomes compared to other strategies that have been suggested. Additionally, it completely pinpoints sophisticated and devastating multi-attacks in the IoT environment.

III. SECURITY ISSUES IN FOG COMPUTING

Edge computing, IoT, and Industry 4.0 have advanced and developed quickly in recent years. Recently, there have been many cloud-based service providers (Cisco, VMware, IBM, Juniper, Big Switch Networks, Versa Networks, Colt Technology, and Lumina SDN) who have shifted from the traditional network paradigm towards Software-Defined Fog (SD-FoG) [52]. The fog server is the fundamental part of the system as it holds critical information related to IoT devices, accumulating and storing the client's data. The basic architecture of the fog paradigm is depicted in Figure 1, representing edge devices connected with fog servers for communication. Numerous distributed fog and edge servers are deployed to offer services over the network to millions of consumers, whereas; the fog servers are connected to cloud servers. The openness and accessibility of the network assets through these devices make fog computing vulnerable. Fog computing has created a new security conundrum due to its significant distribution properties, heterogeneity, mobility, and restricted resources. Because of its limited computing capability, the Fog would struggle to implement a comprehensive security solution to detect and prevent attacks.

Furthermore, because of its position (i.e., close to IoT devices, meaning protection and surveillance are insufficient), the fog will be easier to access and more accessible to hack than the cloud, increasing the likelihood of attacks. Due to its capacity to acquire private information from IoT devices and the cloud, as well as the amount of throughput data, fog will be a target for multiple cyberattacks. Various attacks could be performed to compromise the systems and get important information. These attacks may include SQL injection, Zero-day attack, Man-in-the-middle attack, or others. Botnet attack is a champion among the most lethal attacks. A botnet may be a compromised note in the frameworks linked over the web and are controlled distantly by a criminal to perform malignant action without consent and approval [53]. The essential purpose of criminals to perform Botnet is a cash-related advantage by performing Denial of service assault, Phishing, spamming, and other attacks. SDN is an open, programmable emerging paradigm that permits simple enhancement between the control plane and network devices. SDN energizes network innovation and deploys network capabilities. Research is in progress toward providing security to SDN-based fog computing architectures. There is a need for a proper framework integrated with SDN controllers to monitor fog computing and identify compromised devices. Billions of objects are connected; their administration and control of the enormous number of objects is a challenging task in a circulated system [9]. Hence, monitoring and providing security in fog computing on the internet from cyber-attack, specially Botnet using SDN, is our primary focus. In Table 3, various attacks are illustrated that can be launched against Fog nodes to degrade the performance of the system.

IV. DEEP LEARNING

In this section, we elaborate on the DL algorithms focused on in our proposed scheme. These are as follows:

A. DEEP NEURAL NETWORK (DNN)

A DNN has an input layer, an output layer, and in any case, there is a hidden layer [50]. Each level meets clear organizational types and requirements in the excellence chain

interaction. Managing unlabeled or unstructured data is one of these advanced neural networks' most important applications. The term ''deep learning'' is also used to describe this deep neural network, in which innovation uses part of artificial consciousness. Try to group and query data in ways that go beyond basic information/performance conventions. Utilizing a DL technique has the benefit of being able to automatically identify the crucial features from data without the need for a feature selection procedure. Moreover, DNN methods have proved to be dependable and generalized, if designed well, capable of detecting zero-day threats. Furthermore, DNN computational complexity is defined as follows.

$$m_i = f(\sum_{i=1}^{s} y_i + x_i + v) \tag{1}$$

where the weights for input are y_i, and x_i while the weight for output is m_i. likewise 's' show number of samples, 'v', 'f' is for bias vector and the training function respectively.

B. LONG SHORT TERM MEMORY (LSTM)

LSTM is a neural network used to identify sequence data [51]. The idea of LSTM is to take care of the contribution to the next layer with the output layer of the past layer for better learning of deliberation of information and comprehension. LSTM plays out a comparable assignment for every segment of a course of action, with the output dependent upon past computations. Another way to deal with considering LSTM is that they have a ''memory'' that gets information about what has been resolved up until this point. Also, neurons in LSTM are intended to speak with layers for and significant execution upgrade.

There are three essential gates in the LSTM: a gate for input, a forget gate and a gate for output. The input gate's responsibility is to store the training data in long-term memory. The previous time step is used to initialise the short-term memory, whilst the most recent input data is used to initialise the long-term memory. The training data is separated from the useless information by filters in the input gate, and the valuable data is passed to the sigma function. There are two indicator values for the sigma function: 0 and 1. Fundamental values are represented by a 1, whereas unimportant values are represented by a 0. Long-term memory is used to store the output from the input layer. The forget gate is one of the most significant gates in the LSTM model. Which information should be saved or ignored is determined by multiplying the forget vector values by the current input gate. The subsequent cell receives a new copy from long-term memory when the output from the forget gate has been sent.

$$\begin{aligned} a_t &= \sigma(w_a[n_{t-1}, x_t] + b_a), \\ i_t &= \sigma(w_i[n_{t-1}, x_t] + b_i), \\ \tilde{C}_t &= \tanh(w_c[n_{t-1}, x_t] + b_c), \\ C_t &= a_t * C_{t-1} + i_t * \tilde{C}_t, \\ O_t &= \sigma(w_o[n_{t-1}, x_t] + b_o), \end{aligned}$$

$$n_t = o_t * \tanh(C_t) \tag{2}$$

For the input layer, i_t is the output values, W represents weight values, and b is used for the base. The crucial information is transferred to the following cell using the σ activation function. The forget gate's output is a_t, the output gate is O_t, the cellular cell is c_t, the input information is x_t, and the output information is n_t. As opposed to conventional feed-forward neural networks, LSTM has feedback connections. It can also evaluate the entire data flow along with single data points.

C. CONVOLUTIONAL NEURAL NETWORK (CNN)

The CNN has an input layer, an output layer and many hidden layers [54]. Few among them are convolutional layers. The results are transformed into progressive layers using mathematical models to reproduce a section of the human brain in the progressive layer. When a complex model stimulates the development of the potential of artificial intelligence and proposes a structure that accurately reproduces the types of human brain activities, this is an actual example of in-depth insight.

V. PROPOSED HYBRID DEEP LEARNING APPROACH

We suggest a hybrid DL approach integrating DNN and LSTM to identify IoT botnet attacks. Hybrid models are pretty effective at getting high detection accuracy in a short amount of time [55]. Therefore, to benefit from many DL classifiers simultaneously, we have taken into account DNN and LSTM to enhance the final results. IoT devices produce massive amounts of surge data quickly; therefore, in the suggested approach, LSTM is considered owing to its capacity to perform effective learning for longer data sequences. In contrast, DNN is used to increase the algorithm's predictive capability by enhancing speed efficiency. The proposed architecture is depicted in Figure 2, and Table 5 elaborates on the specific configuration of the hybrid architecture we have presented.

A. DATASET

To evaluate and compare our proposed approach, we have used N_BaIoT [56] dataset in this study. This dataset contains traces of normal and malicious IoT traffic. There are 117 attributes total, 116 of which are network features and a tag. Features name is mentioned in [25]. The Botnet traffic consists of six IoT devices mentioned in Table 4. N_BaIoT dataset has benign traffic along with two malware families of Botnet, namely *1) GAFGYT 2) MIRAI*. *Benign* represents the normal traffic in the network. GAFGYT or BASHLITE is a well-known IoT botnet family having various variants. Its variation can enable real-time DDoS attacks, malicious command execution, and malware download and execution. The attacks that the GAFGYT node can launch are Combo (i.e., establishing a connection to a particular IP address and port to send spam), Junk (i.e., sending spam data), TCP flooding, and UDP flooding, and other attacks. While the MIRAI

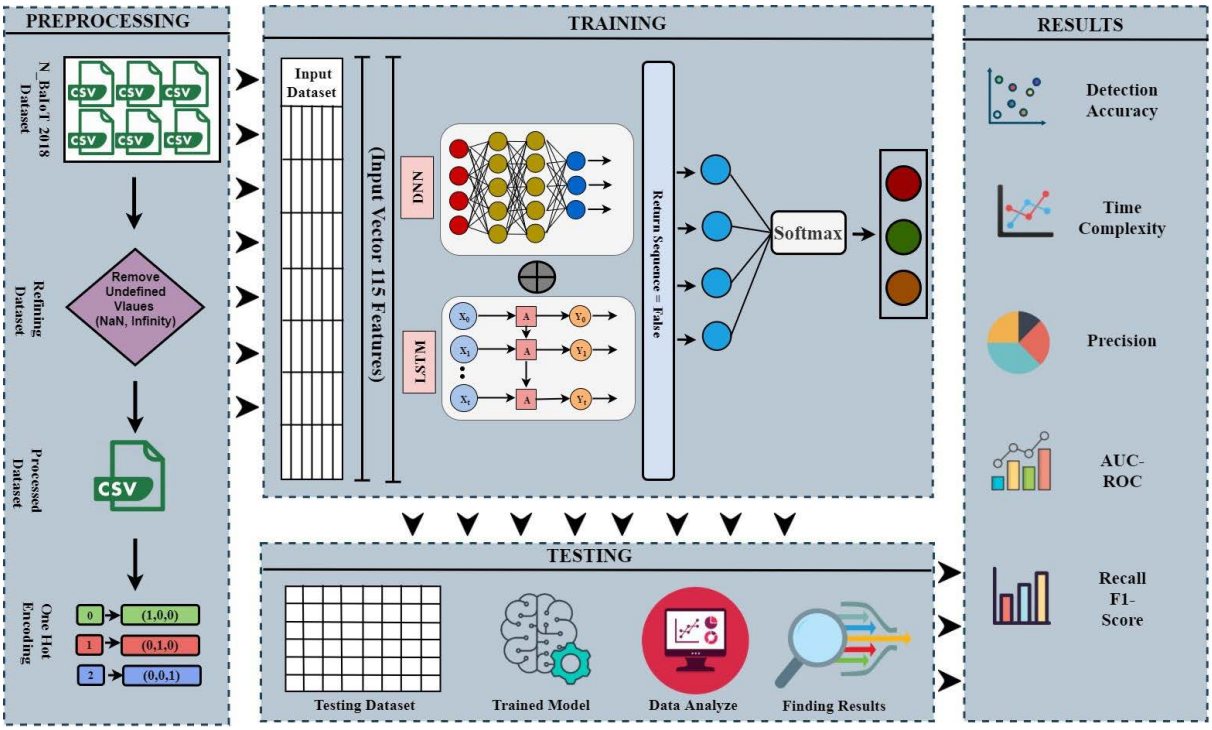

FIGURE 2. Proposed architecture.

TABLE 4. Details of N_BaIoT 2018 dataset.

Devices	Benign	Mirai	Gafgyt
Thermostat	10,000	10,000	10,000
838E Security Camera	10,000	10,000	10,000
737E Security Camera	10,000	10,000	10,000
1011 Web Cam	10,000	–	10,000
1002 Security Camera	10,000	10,000	10,000
1003 Security Camera	10,000	10,000	10,000
Total Records	**60,000**	**50,000**	**60,000**

malware uses ARC processors for targeting smart devices. It transforms them into a network of controllable "zombies" or bots. It can launch UDP flooding, Domain Name Service, TCP ACK, GRE Ethernet, TCP STOMP, HTTP, TCP SYN, UDP Plain, and GRE IP attacks. N_BaIoT dataset is composed of one lac seventy thousand (170,000) records. Of which 60,000 records are for benign traffic, 60,000 are for GAFGYT, and the remaining 50,000 belong to MIRAI traffic. This distribution is also shown in Table 4 with a detailed description of included devices and classes.

B. DETECTION METHODOLOGY

The selection of models used in work is established on the most cutting-adge performers in individual families. The model uses DL methods to produce a adaptable, reliable, and highly accurate botnet antimalware strategy. The proposed detection approach detects benign and malicious behavior by designing a DL-based system. We will use DNN and LSTM to predict results. The findings were extracted and compared to find the best potential solution. Our proposed framework is evaluated using the N_BaIoT dataset.We split the dataset into train and test sets for system analysis and preparation. To train our algorithm, we have given it 90% of the entire dataset, and it will prepare it before making predictions on the test set, which makes up 10% of the whole dataset. The technical details of the proposed and contemporary algorithms are written in Table 5. The table lists each hybrid algorithm's layers, neurons, epochs, optimizer, batch size, and activation functions. The optimizer, activation, and loss functions considered for the implementation are Relu, Categorical cross-entropy, and adam, respectively. The selection of parameters is dependent upon our rigorous experimentation and interactively determining the optimal numbers of layers and neurons along with other parameters. For the DNN-LSTM the pseudo-code is presented in algorithm 2 and the pre-processing phase is presented in algorithm 1. The working of the various phases of the suggested approach is depict in Figure 2 and are explained below:

1) PRE-PROCESSING

The N_BaIoT dataset is pre-processed in order to improve the effectiveness and performance of our proposed hybrid deep learning methodology. In order to guarantee the accuracy of the data, we first inspected the dataset and removed any missing nan and infinite values. To hasten the learning process, we used MinMaxScaler to standardise data between 0 and 1. In addition, we trained a deep learning system on target labels using OHE. The steps for pre-processing are as follows:

TABLE 5. Technical description of proposed algorithm.

Model	Layers	Neurons/Kernel	AF/ LF	Optimizer	Epochs	Batch-size
DNN-LSTM	*DNN Layer* (3)	(400, 100, 50)	*RelU/CC-E*	*Adam*	5	32
	LSTM Layer(3)	(400, 100, 50)	-			
	Merge Layer		-			
	Dense Layer	40	-			
	Dense Layer	15	-			
	Output Layer	3	*softmax*			
CNN2D-LSTM	*CNN2D Layer* (3)	(400, 100, 50)	*RelU/CC-E*	*Adam*	5	32
	LSTM Layer(3)	(400, 100, 50)	-			
	Merge Layer		-			
	Dense Layer	40	-			
	Dense Layer	15	-			
	Output Layer	3	*softmax*			
CNN2D-CNN3D	*CNN2D Layer* (3)	(400, 100, 50)	*RelU/CC-E*	*Adam*	5	32
	CNN3D Layer(3)	(400, 100, 50)	-			
	Merge Layer		-			
	Dense Layer	40	-			
	Dense Layer	15	-			
	Output Layer	3	*softmax*			

AF = Activation Function. LF = Loss Function. CC-E = categorical cross-enotropy.

Step 1 (Input): It is the input stage, where the N_BaIoT dataset is loaded that contains 170,000 IoT traffic which consists of six IoT devices to train our algorithm. Here, six CSV files containing data records for each node are loaded.

Step 2 (Refining Dataset): In this step, the dataset's NAN and infinite values are eliminated because they are the primary causes of the many errors that can occur when the gradient disappears, slowing down the network and rendering it unsafe. The dataset is refined using MinMaxScaller.

Step 3 (Processed Dataset): Here, we get the processed dataset which contains a single CSV file for data collected from all nodes which are free from nan and infinity values.

Step 4 (One Hot encoding): OHE is performed to facilitate the DL algorithm to provide better results. In our case, it normalizes the data according to the three categories based on its label (0, 1, 2).

2) TRAINING

Step 1 (Input pre-processed dataset): 90% of the processed dataset was given as input to our algorithm for training, which comprises 115 features.

Step 2 (Hybrid deep learning phase): In this phase, the hybrid DL technique is applied using two DL algorithms, namely DNN and LSTM. They are executed in parallel on the inputted dataset. Further, the result is merged to get better output.

Step 3 (Add Layer): This layer adds a list of inputs. It accepts a list of similar-shaped tensors as input and outputs a single tensor (also of the same shape).

Step 4 (Return Sequence: Boolean. Whether the final output in the output sequence should be returned or the entire output sequence should be returned. False is the default value.

Step 5 (Softmax function): The softmax function is used as we have multiple classes (Benign, GAFGYT, and MIRAI) that need to be classified properly. As discussed in the results section, it minimizes the prediction errors and improves the detection rate.

Algorithm 1 Pre-Processing

Require: Dataset files Dâ, Dataset Rows R, Dataset Columns C, Combined Dataset D, One Hot Encoding OHE

Ensure: Pre-processing of D

```
function Pre-processing(Dâ)
    D = Merge (Dâ)
    D = Convert Datetime to integer
    D = Labels = OHE(Labels)
    D = Convert Source ip to integer
    D = Convert Destination ip to integer
    for R ← 1 to Rows do
        for C ← 1 to Columns do
            if D[R][C] in ['nan','infinity','null',' ', 'NaN'] then
                Drop Sample (Row)
            end if
        end for
    end for
    Save D as CSV
end function
```

3) TESTING

In this phase, 10% of the dataset has been set aside for testing our algorithm. After performing training of the hybrid algorithm, this dataset is given to analyze the working of the trained algorithm. In the results section, the output of the trained algorithm was gathered and explained.

C. EXPERIMENT DETAILS

The suggested approach's efficiency is demonstrated using various state-of-the-art evaluation measures. These include ROC curve, f1-Score, precision, confusion matrix, recall, and

Algorithm 2 DNN-LSTM

Data training input: There are X network features for every Y unit of network traffic.

Output: is N for normal or label attacks

For each Y_i for N
|
C_i = CNN (Y_i) process
|
End
|
For each C_i process
L_i = LSTM (C_i) process
|
Merge
|
End
For Each L_i Process
N = softmax (L_i) end

TABLE 6. Hardware specification for evaluation of proposed system.

Components	Specification
CPU	Corei7-8750H@2.21GHz
RAM	12GB
OS	Windows 10
Language	Python
Libraries	Numpy, TensorFlow, Scikitlearn, Pandas
Software	Anaconda.Navigator, Spyder 4.1.5, Origin 2019b

accuracy. In table 6 details of the hardware and software resources used in the experiment are shown.

1) CONFUSION MATRIX

Subsequently, other parameters are calculated using the confusion matrix's FP, FN, TP, and TN values. The confusion matrix of proposed and contemporary algorithms are shown in Figure 3-5. Where in Figure 3 the confusion matrix of the proposed model of DNN-LSTM shows the exceptional performance of hybrid DL models to predict and identify attacks efficiently. In Figure 4 hybrid CNN2D-LSTM results are also promising here, while Figure 5 shows the performance of the hybrid DNN2D-DNN3D that performs poor results.

2) ROC CURVE

The ROC graph represents the relationship between the true and FP rates. The AUC (Area Under the Curve) reveals that the larger the AUC, the better the system performance. The ROC curve for the suggested approach is shown in 6.

3) ACCURACY

The accuracy rate shows the number of correctly categorized connections based on the proposed model, including normal and intrusive connections. The formula for accuracy is as

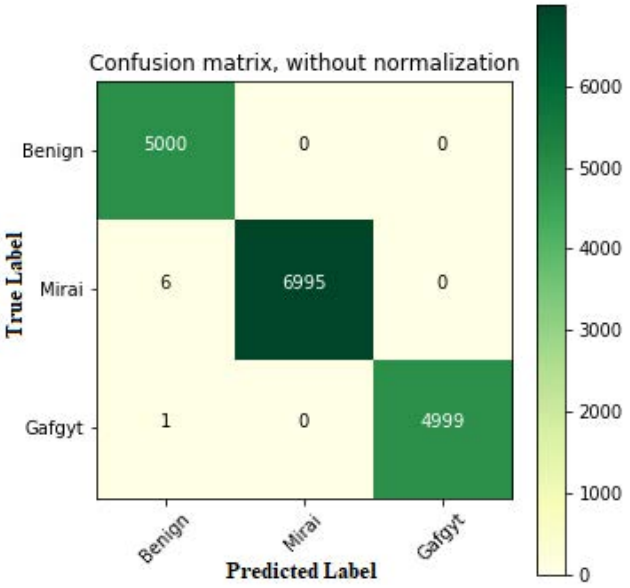

FIGURE 3. Confusion matrix of DNN-LSTM.

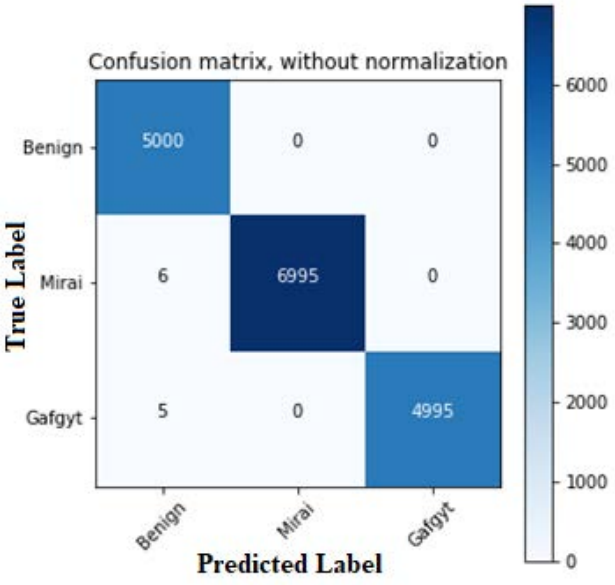

FIGURE 4. Confusion matrix of CNN2D-LSTM.

follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

4) PRECISION

The precision metric, which is the proportion of accurate positive to true positive detection, should also be considered while assessing the proposed model. The following formula can be used to calculate precision:

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

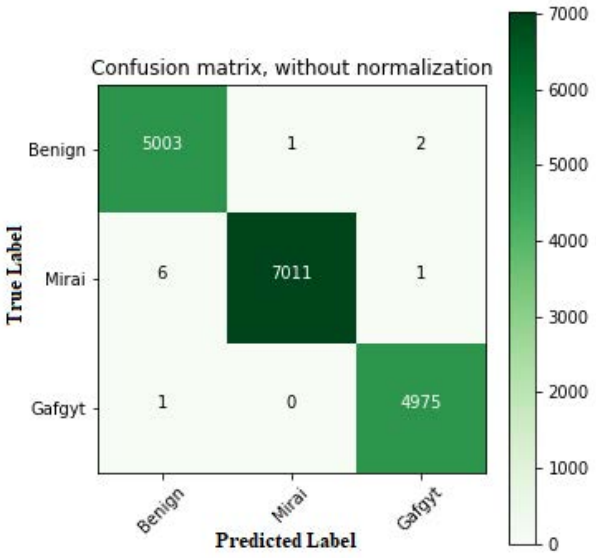

FIGURE 5. Confusion matrix of hybrid CNN2D-CNN3D.

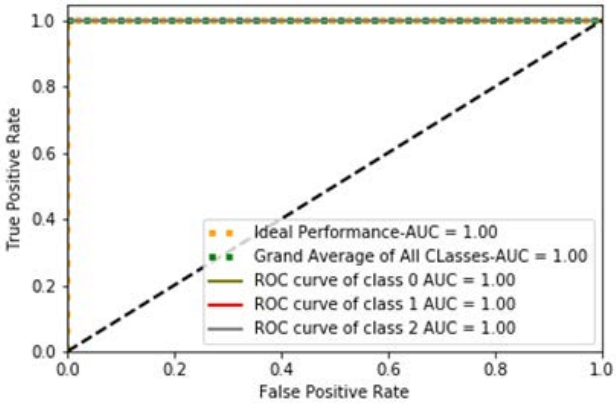

FIGURE 6. ROC curve of LSTM-DNN.

5) RECALL

It is the proportion of exact positive tests to exact malware samples. The following formula can be used to calculate recall:

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

6) F1-SCORE

The accuracy of a model on a dataset is measured by the F-score, also known as the F1-score. The f1-score value is calculated using a formula that weights the precision and recall rates. The following formula is used for the calculation of F1-score:

$$F - score = \frac{2 * TP}{2 * TP + FP + FN} \tag{6}$$

VI. RESULT AND DISCUSSION

We also compare our proposed algorithm with other constructed algorithms to evaluate the proposed system. We used a cross-validation method that divided the training data into several folds (k), using a subset of the data as a test set and the remaining (k-1) subset as a training set each time to gather precise information about the performance of our model. As a result, the data over-or under-fitting problem was resolved, and detailed information on the model's performance for unexpected data during training was provided. Table 7 shows the 10-fold cross-validated findings of our hybrid algorithms compared to other proposed hybrid DL models.

A. F1-SCORE, RECALL, PRECISION AND ACCURACY

The DL-based botnet detection in fog computing using SDN provides an efficient and scalable solution. In order to attain a high accuracy rate, hybrid DL models are used. We employed 10-fold cross-validation to acquire the average detection accuracy, precision, recall, and F1-score to get objective results. The results are presented in a ROC Curve, which can maximize the true positive value while minimizing the false positive value. The detection accuracy, precision, recall, and f1-score of hybrid DNN-LSTM, hybrid CNN2D-LSTM, and CNN2D-CNN3D are also evaluated as part of the evaluation matrix. For Hybrid DNN-LSTM, Hybrid CNN2D-LSTM, and Hybrid CNN2D-CNN3D, Figure 7 shows accuracy, recall, precision, and F1-score. Hybrid DNN-LSTM outperforms other contemporary models in terms of accuracy. For the proposed hybrid DNN-LSTM the accuracy is 99.98%, precision is 99.97%, recall is 99.87%, and f1-score is 99.87%. Likewise, for the hybrid CNN2D-LSTM, the accuracy is 99.95%, precision is 99.94%, and for the recall and f1-score, it is 99.85%. Additionally, for the hybrid CNN2D-CNN3D algorithm, the accuracy is 99.93%, precision is 99.91%, and for the recall and f1-score, it is 99.86%.

B. FPR, FDR, FNR AND FOR

To further enhance the evaluation criteria and evaluate models with better insight, advanced parameters such as FOR, FDR, FPR and FNR have also been used in the evaluation, presented in Figure 8. The high result values of TNR and low values of FNR signify the importance of the proposed Hybrid models for attack detection. The Hybrid DNN-LSTM gain FPR of 0.00009%, FDR 0.0002%, FNR 0.0012% and FOR 0.0005%. Furthermore, the hybrid CNN2D-LSTM gained an FPR of 0005%, FDR 0.0001%, FNR 0.0014% and FOR 0.0005%. Additionally, for the hybrid CNN2D-CNN3D gain FPR of 0.0002%, FDR 0.0006%, FNR 0.0014% and FOR 0.0007%.

For a more detailed examination of the suggested framework, the authors calculated FNR, FPR, FDR, and FOR values. Furthermore, the low rates of the early indicators indicate that the algorithm is performing well. Where the FPR measures the percentage of false positives against all positive predictions, The FDR is the proportion of hypotheses that we falsely think are true, FNR is negative outcomes that the model predicted incorrectly, and The FOR measures most individuals whose test results were negative but whose actual condition was positive.

TABLE 7. 10 Fold of hybrid DNN-LSTM, hybrid CNN2D-LSTM, and hybrid CNN2D-CNN3D.

	Accuracy (%)			Recall (%)			Precision (%)			F1-Score (%)		
Folds	!!	@@	++	!!	@@	++	!!	@@	++	!!	@@	++
1	**99.99**	99.96	99.96	**99.99**	**99.99**	**99.99**	**99.99**	99.90	99.90	**99.99**	**99.99**	**99.99**
2	99.93	99.73	99.72	99.90	**99.99**	**99.99**	99.93	99.90	99.90	99.90	**99.99**	**99.99**
3	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**
4	**99.99**	99.86	99.86	**99.99**	99.70	99.70	**99.99**	99.90	99.90	**99.99**	99.70	99.70
5	99.96	99.93	99.93	99.89	99.90	99.90	**99.99**	99.90	99.90	99.89	99.90	99.90
6	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**
7	99.96	**99.99**	**99.99**	99.90	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	99.90	**99.99**	**99.99**
8	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**
9	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**	**99.99**
10	**99.99**	99.96	99.96	**99.99**	99.90	9.90	**99.99**	**99.99**	**99.99**	**99.99**	99.90	99.90

Abbreviation Terms: !! Hybrid (DNN & LSTM), @@ Hybrid (CNN2D & LSTM), ++ Hybrid (CNN2D & CNN3D)

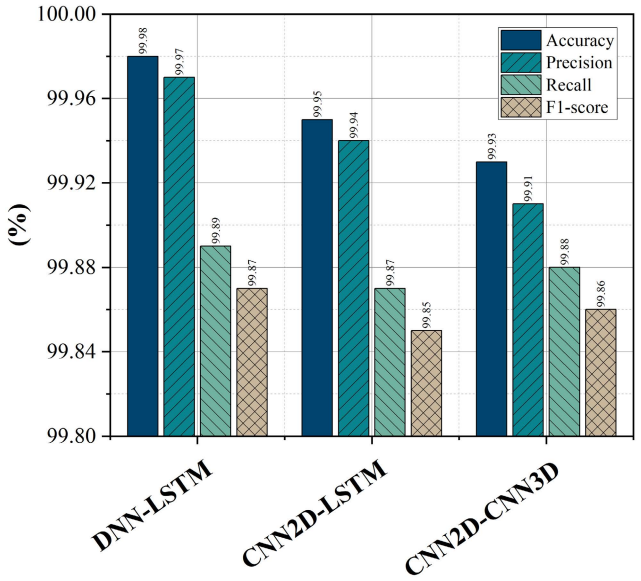

FIGURE 7. Accuracy, Precision, Recall and F1-Score of DNN-LSTM, CNN2D-LSTM, and CNN2D-CNN3D.

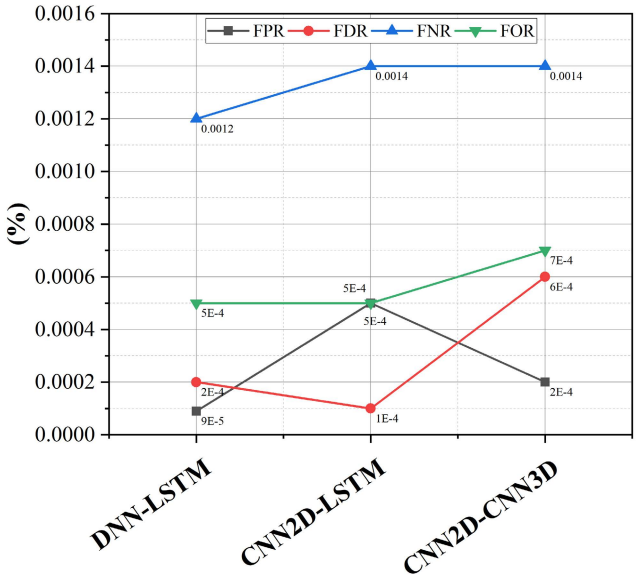

FIGURE 8. FPR, FDR, FNR, and FOR of DNN-LSTM, CNN2D-LSTM, and CNN2D-CNN3D.

C. TNR, NPV AND MCC

We calculated additional measures (i.e., TNR, NPV, MCC) from the confusion matrix to thoroughly analyze our proposed algorithm. The confusion matrix depicts detection performance with TP, TN, FP, and FN values, providing precise results where our proposed algorithm performs best.

The parameter's i.e. TNR, NPV, and MCC are presented in Figure 9. For the hybrid DNN-LSTM algorithm, the value of TNR is 99.99%, the MCC value is 99.93%, and the NPV value is 99.94%. For the CNN2D-LSTM algorithm, the value of TNR is 99.97%, MCC is 99.92%, and NPV is 99.94%. Likewise, for the hybrid CNN2D-CNN3D algorithm, the value of TNR is 99.96%, MCC is 99.91%, and NPV is 99.93%. Our proposed algorithm has the highest value for TNR and MCC while having the same NPV value as CNN2D-LSTM.

D. BM, MK AND TS

BM is a worldwide test performance measure used to assess a diagnostic procedure's overall discriminative power. MK is the state of standing out as divergent compared to the standard form. TS measures the ability of an algorithm and is often used to compute the results over time. The hybrid DNN-LSTM algorithm value for BM is 99.85%, MK is 99.92%, and TS is 99.84%. For the hybrid CNN2D-LSTM, the value for BM is 99.85%, MK is 99.89%, and TS is 99.82%. Additionally, The hybrid CNN2D-CNN3D algorithm shows the value of BM, MK, and TS as 99.82%, 99.86%, and 99.71%, respectively. The proposed algorithm values for BM, MK, and TS are presented in Figure 10, where our proposed algorithm has the highest value for MK and TS while having the same BM value as CNN2D-LSTM.

E. TESTING TIME

Figure 11 depicts the testing time of the three hybrid combinations of DL algorithms used in this research work, i.e., DNN-LSTM, CNN2D-CNN3D, and CNN2D-LSTM. The CNN2D-LSTM and CNN2D-CNN3D hybrid approaches have a 0.14ms and 0.19ms testing time, respectively,

TABLE 8. Comparison of proposed work with other solutions.

Method	Models	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Detection Time
Proposed	DNN-LSTM	N_BaIoT 2018	99.98	99.97	99.87	99.87	0.22(ms)
[16]	LSTM	N_BaIoT 2018	99.97	99.90	99.90	99.85	2.65
[45]	CNN-RNN	CTU-13, ISOT	99.00	98.00	97.00	97.00	–
[47]	LSTM-CNN	CIDDS2017	99.92	99.85	99.85	99.91	29
[57]	DNN	Network Flow	99.00	–	–	–	–
[58]	MLP-AE	N_BaIoT 2018	99.25	98.84	98.00	98.00	3.75(ms)
[59]	DBN	N_BaIoT 2018	95.60	98.27	92.82	92.82	–
[60]	CNN, LSTM	N_BaIoT 2018	94.30	93.48	93.67	93.58	–
[61]	KNN, RF, NB	N_BaIoT 2018	99.00	86.65	99.00	99.00	–

Abbreviation Terms: ms– millisecond.

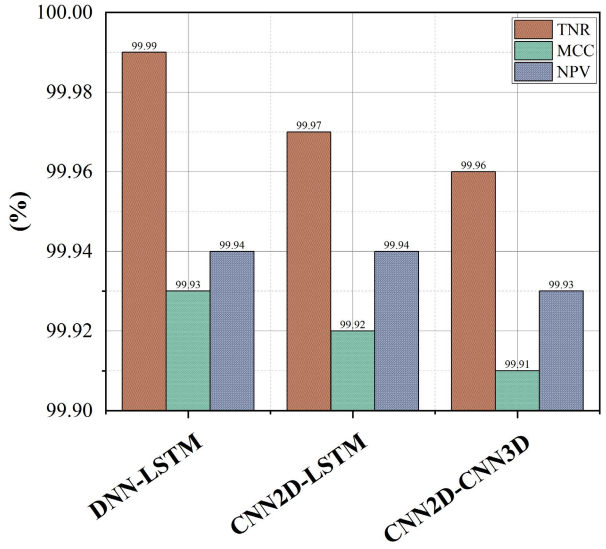

FIGURE 9. TNR, NPV and MCC of DNN-LSTM, CNN2D-LSTM, and CNN2D-CNN3D.

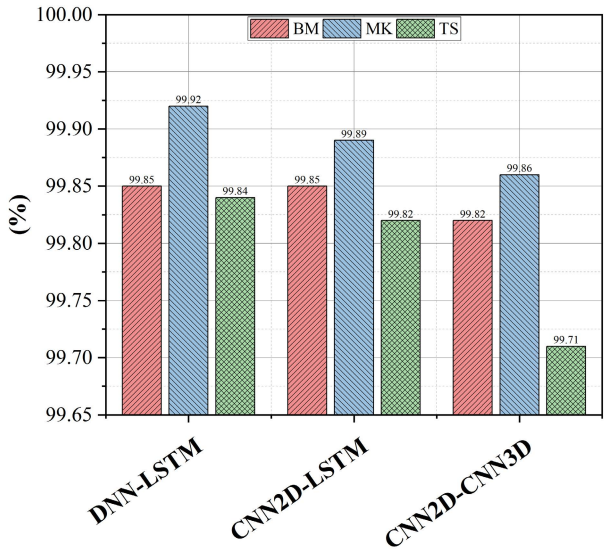

FIGURE 10. BM, MK and TS of DNN-LSTM, CNN2D-LSTM, and CNN2D-CNN3D.

as compared to our suggested hybrid technique DNN-LSTM, which takes 0.22ms. The suggested hybrid DL DNN-LSTM

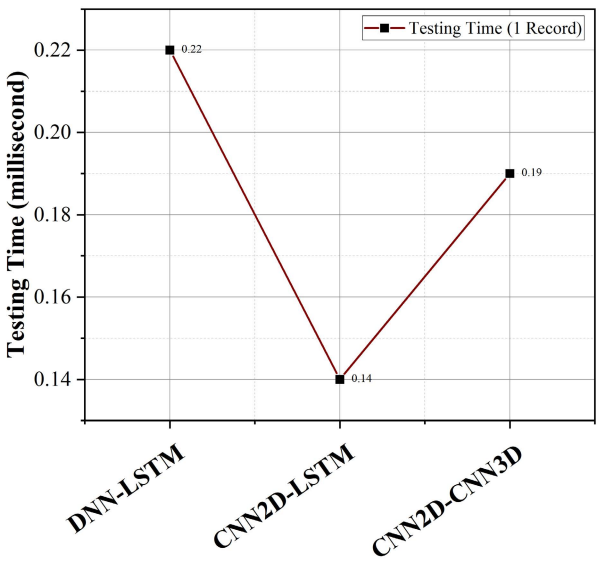

FIGURE 11. Testing time of DNN-LSTM, CNN2D-LSTM, and CNN2D-CNN3D.

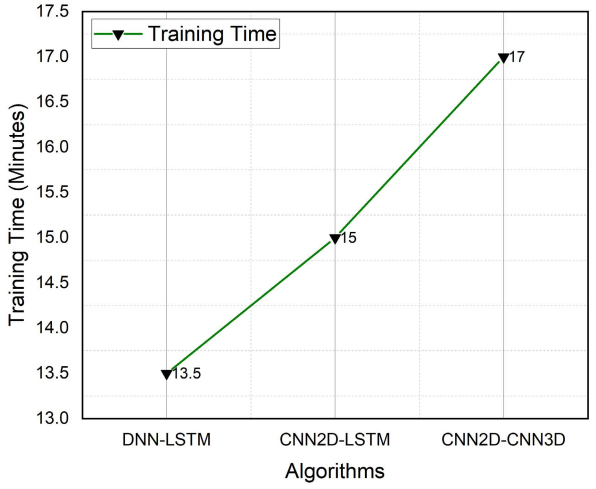

FIGURE 12. Training time of DNN-LSTM, CNN2D-LSTM and CNN2D-CNN3D.

has a negligible compromise in testing time compared to CNN2D-LSTM and CNN2D-CNN3D hybrid models.

Hence, we compared these combinations in various ways in Table 8 to highlight the selection of our suggested model

in terms of the f1-score, accuracy, precision and recall. The findings show that the suggested hybrid model, which uses the DNN-LSTM approach, achieves better f1-score, recall, and accuracy while precision is equivalent to CNN2D-LSTM and CNN2D-CNN3D hybrid models.

F. TRAINING TIME

The training time for each algorithm is presented in Figure 12. Training our proposed algorithm took 13.5 minutes using the N_BaIoT dataset. This time is mentioned here to show that the proposed model takes minimum time for training. The results depicted in Figure 12 show the difference between various algorithms according to their time utilization for training. It shows that the proposed scheme DNN-LSTM utilizes minimum time compared to other counterparts.

VII. CONCLUSION

SDN-based fog computing architectures are the trending networking paradigms for several applications based on the IoT infrastructure. Fog computing systems are vulnerable to various types of Botnet attacks. Hence, there is a need to integrate a security framework that empowers the SDN to monitor the network anomalies against the Botnet attacks. DL algorithms are considered more effective for the IoT-based infrastructures that work on unstructured and large amounts of data. DL-based intrusion detection schemes can detect Botnet attacks in the SDN-enabled fog computing IoT system.

We created a framework that utilizes a hybrid DL detection scheme to identify the IoT botnet attacks. It is trained against the dataset that contains normal and malicious data, and then we used this framework to identify botnet attacks that targeted different IoT devices. Our methodology comprises a botnet dataset, a botnet training paradigm, and a botnet detection paradigm.

Our botnet dataset was built using the N_BaIoT dataset, which was produced by driving botnet attacks from the Gafgyt and Mirai botnets into six distinct types of IoT devices. Five attack types, including UDP, TCP, and ACK, are included in both Gafgyt and Mirai attacks. We developed a botnet detection based on three hybrid models—DNN-LSTM, CNN2D-LSTM, and CNN2D-CNN3D. Using this training model as a foundation, we developed a botnet detection paradigm that can recognise significant botnet attacks. The botnet detection approach is part of a multi-class classification model that can distinguish between the sub-attacks and innocuous data. The fact-finding analysis showed that our hybrid framework DNN-LSTM model had the highest accuracy of 99.98% at identifying the gafgyt and Mirai botnets in the N_BaIoT environment. In 2014 and 2016, the gafgyt and Mirai botnets essentially targeted home routers and IP cameras. The N_BaIoT dataset we used for our experiments revealed that rather than the type of IoT devices, the type of training models has a more significant impact on botnet detection performance. We think creating DNN-LSTM-based IoT botnet detection models would be an excellent strategy to enhance botnet identification for different IoT devices.

In the future, we have in mind to compare the performance of the proposed hybrid algorithm to that of other IoT datasets with a more considerable number of nodes. Further, there is a need to test more combinations of DL algorithms and traditional machine learning algorithms.

REFERENCES

[1] Z. Hussain, A. Akhunzada, J. Iqbal, I. Bibi, and A. Gani, "Secure IIoT-enabled industry 4.0," *Sustainability*, vol. 13, no. 22, p. 12384, Nov. 2021.

[2] R. K. Barik, H. Dubey, K. Mankodiya, S. A. Sasane, and C. Misra, "GeoFog4Health: A fog-based SDI framework for geospatial health big data analysis," *J. Ambient Intell. Hum. Comput.*, vol. 10, no. 2, pp. 551–567, Feb. 2019.

[3] S. Khan, S. Parkinson, and Y. Qin, "Fog computing security: A review of current applications and security solutions," *J. Cloud Comput.*, vol. 6, no. 1, pp. 1–22, Dec. 2017.

[4] J. Malik, A. Akhunzada, I. Bibi, M. Talha, M. A. Jan, and M. Usman, "Security-aware data-driven intelligent transportation systems," *IEEE Sensors J.*, vol. 21, no. 14, pp. 15859–15866, Jul. 2021.

[5] Z. Ning, X. Hu, Z. Chen, M. Zhou, B. Hu, J. Cheng, and M. S. Obaidat, "A cooperative quality-aware service access system for social internet of vehicles," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2506–2517, Aug. 2018.

[6] X. Wang, Z. Ning, M. C. Zhou, X. Hu, L. Wang, Y. Zhang, F. R. Yu, and B. Hu, "Privacy-preserving content dissemination for vehicular social networks: Challenges and solutions," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1314–1345, 2nd Quart., 2019.

[7] Z. Ning, Y. Li, P. Dong, X. Wang, M. S. Obaidat, X. Hu, L. Guo, Y. Guo, J. Huang, and B. Hu, "When deep reinforcement learning meets 5G-enabled vehicular networks: A distributed offloading framework for traffic big data," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1352–1361, Feb. 2020.

[8] X. Wang, Z. Ning, and L. Wang, "Offloading in internet of vehicles: A fog-enabled real-time traffic management system," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4568–4578, Oct. 2018.

[9] H. Dubey, J. Yang, N. Constant, A. M. Amiri, Q. Yang, and K. Makodiya, "Fog data: Enhancing telehealth big data through fog computing," in *Proc. ASE BigData Socialinform.*, 2015, pp. 1–6.

[10] W. U. Khan, T. N. Nguyen, F. Jameel, M. A. Jamshed, H. Pervaiz, M. A. Javed, and R. Jäntti, "Learning-based resource allocation for backscatter-aided vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, early access, Nov. 18, 2021, doi: 10.1109/TITS.2021.3126766.

[11] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, and P. Liljeberg, "Exploiting smart e-health gateways at the edge of healthcare Internet-of-Things: A fog computing approach," *Future Gener. Comput. Syst.*, vol. 78, pp. 641–658, Jan. 2018.

[12] Z. Xiao and Y. Xiao, "Security and privacy in cloud computing," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 843–859, 2nd Quart., 2013.

[13] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 602–622, 1st Quart., 2016.

[14] S. S. C. Silva, R. M. P. Silva, R. C. G. Pinto, and R. M. Salles, "Botnets: A survey," *Comput. Netw.*, vol. 57, no. 2, pp. 378–403, 2013.

[15] 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.

[16] T. Hasan, A. Adnan, T. Giannetsos, and J. Malik, "Orchestrating SDN control plane towards enhanced IoT security," in *Proc. 6th IEEE Conf. Netw. Softw. (NetSoft)*, Jun. 2020, pp. 457–464.

[17] W. U. Khan, J. Liu, F. Jameel, V. Sharma, R. Jäntti, and Z. Han, "Spectral efficiency optimization for next generation NOMA-enabled IoT networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 15284–15297, Dec. 2020.

[18] L. Yu, Q. Wang, G. Barrineau, J. Oakley, R. R. Brooks, and K.-C. Wang, "TARN: A SDN-based traffic analysis resistant network architecture," in *Proc. 12th Int. Conf. Malicious Unwanted Softw. (MALWARE)*, Oct. 2017, pp. 91–98.

[19] E. Rodríguez, B. Otero, N. Gutiérrez, and R. Canal, "A survey of deep learning techniques for cybersecurity in mobile networks," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1920–1955, 3rd Quart., 2021.

[20] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep recurrent neural network for intrusion detection in SDN-based networks," in *Proc. 4th IEEE Conf. Netw. Softw. Workshops (NetSoft)*, Jun. 2018, pp. 202–206.

[21] R. Chen, W. Niu, X. Zhang, Z. Zhuo, and F. Lv, "An effective conversation-based botnet detection method," *Math. Problems Eng.*, vol. 2017, pp. 1–9, Apr. 2017.

[22] S. Al-mashhadi, M. Anbar, I. Hasbullah, and T. A. Alamiedy, "Hybrid rule-based botnet detection approach using machine learning for analysing DNS traffic," *PeerJ Comput. Sci.*, vol. 7, p. e640, Aug. 2021.

[23] A. O. Prokofiev, Y. S. Smirnova, and V. A. Surov, "A method to detect Internet of Things botnets," in *Proc. IEEE Conf. Russian Young Res. Electr. Electron. Eng. (EIConRus)*, Jan. 2018, pp. 105–108.

[24] M. Waqas, K. Kumar, A. A. Laghari, U. Saeed, M. M. Rind, A. A. Shaikh, F. Hussain, A. Rai, and A. Q. Qazi, "Botnet attack detection in Internet of Things devices over cloud environment via machine learning," *Concurrency Comput., Pract. Exp.*, vol. 34, no. 4, Feb. 2022, Art. no. e6662.

[25] J. A. Faysal, S. T. Mostafa, J. S. Tamanna, K. M. Mumenin, M. M. Arifin, M. A. Awal, A. Shome, and S. S. Mostafa, "XGB-RF: A hybrid machine learning approach for IoT intrusion detection," *Telecom*, vol. 3, no. 1, pp. 52–69, Jan. 2022.

[26] A. O. Khadidos, H. Manoharan, S. Selvarajan, A. O. Khadidos, K. H. Alyoubi, and A. Yafoz, "A classy multifacet clustering and fused optimization based classification methodologies for SCADA security," *Energies*, vol. 15, no. 10, p. 3624, May 2022.

[27] S. Shitharth, K. M. Prasad, K. Sangeetha, P. R. Kshirsagar, T. S. Babu, and H. H. Alhelou, "An enriched RPCO-BCNN mechanisms for attack detection and classification in SCADA systems," *IEEE Access*, vol. 9, pp. 156297–156312, 2021.

[28] B. Jo, R. Khan, and Y.-S. Lee, "Hybrid blockchain and Internet-of-Things network for underground structure health monitoring," *Sensors*, vol. 18, no. 12, p. 4268, Dec. 2018.

[29] T. G. Nguyen, T. V. Phan, B. T. Nguyen, C. So-In, Z. A. Baig, and S. Sanguanpong, "SeArch: A collaborative and intelligent NIDS architecture for SDN-based cloud IoT networks," *IEEE Access*, vol. 7, pp. 107678–107694, 2019.

[30] S. M. Umran, S. Lu, Z. A. Abduljabbar, J. Zhu, and J. Wu, "Secure data of industrial Internet of Things in a cement factory based on a blockchain technology," *Appl. Sci.*, vol. 11, no. 14, p. 6376, Jul. 2021.

[31] S. Miller and C. Busby-Earle, "The role of machine learning in botnet detection," in *Proc. 11th Int. Conf. for Internet Technol. Secured Trans. (ICITST)*, Dec. 2016, pp. 359–364.

[32] P. C. Tikekar, S. S. Sherekar, and V. M. Thakre, "Features representation of botnet detection using machine learning approaches," in *Proc. Int. Conf. Comput. Intell. Comput. Appl. (ICCICA)*, Nov. 2021, pp. 1–5.

[33] A. Prakash and R. Priyadarshini, "An intelligent software defined network controller for preventing distributed denial of service attack," in *Proc. 2nd Int. Conf. Inventive Commun. Comput. Technol. (ICICCT)*, Apr. 2018, pp. 585–589.

[34] T. V. Phan, T. Van Toan, D. Van Tuyen, T. T. Huong, and N. H. Thanh, "OpenFlowSIA: An optimized protection scheme for software-defined networks from flooding attacks," in *Proc. IEEE 6th Int. Conf. Commun. Electron. (ICCE)*, Jul. 2016, pp. 13–18.

[35] T. Abhiroop, S. Babu, and B. S. Manoj, "A machine learning approach for detecting DoS attacks in SDN switches," in *Proc. 24th Nat. Conf. Commun. (NCC)*, Feb. 2018, pp. 1–6.

[36] J. Ye, X. Cheng, J. Zhu, L. Feng, and L. Song, "A DDoS attack detection method based on SVM in software defined network," *Secur. Commun. Netw.*, vol. 2018, pp. 1–8, Apr. 2018.

[37] S. Almutairi, S. Mahfoudh, S. Almutairi, and J. S. Alowibdi, "Hybrid botnet detection based on host and network analysis," *J. Comput. Netw. Commun.*, vol. 2020, pp. 1–16, Jan. 2020.

[38] G. Kocher and G. Kumar, "Machine learning and deep learning methods for intrusion detection systems: Recent developments and challenges," *Soft Comput.*, vol. 25, no. 15, pp. 9731–9763, Aug. 2021.

[39] 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.

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

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

[42] D. Javeed, T. Gao, M. T. Khan, and I. Ahmad, "A hybrid deep learning-driven SDN enabled mechanism for secure communication in Internet of Things (IoT)," *Sensors*, vol. 21, no. 14, p. 4884, Jul. 2021.

[43] C. D. McDermott, F. Majdani, and A. V. Petrovski, "Botnet detection in the Internet of Things using deep learning approaches," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.

[44] A. Dawoud, S. Shahristani, and C. Raun, "Deep learning and software-defined networks: Towards secure IoT architecture," *Internet Things*, vols. 3–4, pp. 82–89, Oct. 2018.

[45] A. Pektaş and T. Acarman, "Deep learning to detect botnet via network flow summaries," *Neural Comput. Appl.*, vol. 31, no. 11, pp. 8021–8033, Nov. 2019.

[46] S. K. Dey and M. M. Rahman, "Flow based anomaly detection in software defined networking: A deep learning approach with feature selection method," in *Proc. 4th Int. Conf. Electr. Eng. Inf. Commun. Technol. (iCEEiCT)*, Sep. 2018, pp. 630–635.

[47] I. Ullah, B. Raza, S. Ali, I. A. Abbasi, S. Baseer, and A. Irshad, "Software defined network enabled fog-to-things hybrid deep learning driven cyber threat detection system," *Secur. Commun. Netw.*, vol. 2021, pp. 1–15, Dec. 2021.

[48] P. Jithu, J. Shareena, A. Ramdas, and A. P. Haripriya, "Intrusion detection system for IoT botnet attacks using deep learning," *Social Netw. Comput. Sci.*, vol. 2, no. 3, pp. 1–8, May 2021.

[49] S. Shitharth, P. R. Kshirsagar, P. K. Balachandran, K. H. Alyoubi, and A. O. Khadidos, "An innovative perceptual pigeon galvanized optimization (PPGO) based likelihood Naïve Bayes (LNB) classification approach for network intrusion detection system," *IEEE Access*, vol. 10, pp. 46424–46441, 2022.

[50] A. Canziani, A. Paszke, and E. Culurciello, "An analysis of deep neural network models for practical applications," 2016, *arXiv:1605.07678*.

[51] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[52] *'Top Providers of SDN Services*. Accessed: Jan. 15, 2022. [Online]. Available: https://www.cambridgewireless.co.uk and https://www.cambridgewireless.co.uk/news/2020/jan/28/top-providers-sdn-services/

[53] G. Kaur, "A novel distributed machine learning framework for semi-supervised detection of botnet attacks," in *Proc. 11th Int. Conf. Contemp. Comput. (IC3)*, Aug. 2018, pp. 1–7.

[54] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proc. Int. Conf. Eng. Technol. (ICET)*, Aug. 2017, pp. 1–6.

[55] 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.

[56] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-BaIoT—Network-based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Comput.*, vol. 17, no. 3, pp. 12–22, Jul./Aug. 2018.

[57] A. Pektaş and T. Acarman, "Botnet detection based on network flow summary and deep learning," *Int. J. Netw. Manage.*, vol. 28, no. 6, Nov. 2018, Art. no. e2039.

[58] V. Rey, P. M. S. Sánchez, A. H. Celdrán, and G. Bovet, "Federated learning for malware detection in IoT devices," *Comput. Netw.*, vol. 204, Feb. 2022, Art. no. 108693.

[59] T. V. Khoa, Y. M. Saputra, D. T. Hoang, N. L. Trung, D. Nguyen, N. V. Ha, and E. Dutkiewicz, "Collaborative learning model for cyberattack detection systems in IoT industry 4.0," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, May 2020, pp. 1–6.

[60] G. De La Torre Parra, P. Rad, K.-K.-R. Choo, and N. Beebe, "Detecting Internet of Things attacks using distributed deep learning," *J. Netw. Comput. Appl.*, vol. 163, Aug. 2020, Art. no. 102662.

[61] H. Alazzam, A. Alsmady, and A. A. Shorman, "Supervised detection of IoT botnet attacks," in *Proc. 2nd Int. Conf. Data Sci., E-Learn. Inf. Syst.*, 2019, pp. 1–6.

FRAIDOON SATTARI received the bachelor's degree in computer science from Gomal University, D. I. Khan, Pakistan. He is currently pursuing the M.S. degree in information security with COMSATS University Islamabad, Islamabad Campus, Pakistan. His research interests include network security, cloud computing, malware analysis and detection, software-defined networking, fog security, and the Internet of Things.

ASHFAQ HUSSAIN FAROOQI received the master's and Ph.D. degrees in computer science from the National University of Computer and Emerging Sciences, Islamabad, Pakistan, in 2009 and 2018, respectively. He is currently working as an Assistant Professor with the Department of Computer Science, Air University, Islamabad. He has authored several articles in refereed journals. His research interests include wireless communication, network security, computational intelligence, risk assessment, and risk management.

ZAKRIA QADIR received the M.Sc. degree in sustainable environment and energy systems from Middle East Technical University, Turkey, in 2019. He is currently pursuing the Ph.D. degree in wireless communication and cloud computing with Western Sydney University, Australia. His research interests include sustainable cities, artificial intelligence, machine learning, optimization techniques, wireless communication, the Internet of things, renewable energy technology, and cloud computing.

BASIT RAZA received the master's degree in computer science from the University of Central Punjab, Lahore, Pakistan, and the dual Ph.D. degree in computer science from International Islamic University Islamabad and the University of Technology Malaysia, in 2014. He is currently an Associate Professor with the Department of Computer Science, COMSATS University Islamabad (CUI), Islamabad, Pakistan. He has authored several articles in refereed journals. His research interests include database management systems, data mining, data warehousing, machine learning, deep learning, and artificial intelligence. He has been serving as a Reviewer for prestigious journals, such as *Applied Soft Computing*, *Swarm and Evolutionary Computation*, *Swarm Intelligence*, *Applied Intelligence*, IEEE Access, and *Future Generation Computer Systems*.

HADI NAZARI received the bachelor's degree in computer science from Gomal University, D. I. Khan, Pakistan. He is currently pursuing the M.S. degree in information security with the Department of Computer Science, COMSATS University Islamabad, Islamabad, Pakistan. His research interests include an access control systems, software-defined networking, smart devices security, threat detection and intelligence, malware analysis and detection, application of deep learning and machine learning in cyber defense, and network security.

MUHANNAD ALMUTIRY (Member, IEEE) received the B.Sc. degree in electrical engineering from Umm Al Qura University, Makkah, Saudi Arabia, in 2007, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of Dayton, Dayton, OH, USA, in 2010 and 2016, respectively. From 2013 to 2016, he was a Research Assistant with the Mumma Radar Laboratory, University of Dayton. He has been an Assistant Professor with the Department of Electrical Engineering, Northern Border University, Saudi Arabia, since 2016. His research interests include radar imaging, subsurface sensing, UAV detection, sonar, and intelligent adaptive radar networks.