Hindawi
Security and Communication Networks
Volume 2021, Article ID 3806459, 23 pages
https://doi.org/10.1155/2021/3806459

Research Article

Botnet Attack Detection by Using CNN-LSTM Model for Internet of Things Applications

Hasan Alkahtani[1] **and Theyazn H. H. Aldhyani**[2]

[1]*College of Computer Science and Information Technology, King Faisal University, P.O. Box 4000, Al-Ahsa, Saudi Arabia*
[2]*Community College of Abqaiq, King Faisal University, P.O. Box 4000, Al-Ahsa, Saudi Arabia*

Correspondence should be addressed to Theyazn H. H. Aldhyani; taldhyani@kfu.edu.sa

Received 6 July 2021; Revised 14 August 2021; Accepted 24 August 2021; Published 10 September 2021

Academic Editor: Abdallah Meraoumia

Copyright © 2021 Hasan Alkahtani and Theyazn H. H. Aldhyani. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Internet of Things (IoT) has grown rapidly, and nowadays, it is exploited by cyber attacks on IoT devices. An accurate system to identify malicious attacks on the IoT environment has become very important for minimizing security risks on IoT devices. Botnet attacks are among the most serious and widespread attacks, and they threaten IoT devices. Motionless IoT devices have a security weakness due to lack of sufficient memory and computation results for a security platform. In addition, numerous existing systems present themselves for finding unknown patterns from IoT networks to improve security. In this study, hybrid deep learning, a convolutional neural network and long short-term memory (CNN-LSTM) algorithm, was proposed to detect botnet attacks, namely, BASHLITE and Mirai, on nine commercial IoT devices. Extensive empirical research was performed by employing a real N-BaIoT dataset extracted from a real system, including benign and malicious patterns. The experimental results exposed the superiority of the CNN-LSTM model with accuracies of 90.88% and 88.61% in detecting botnet attacks from doorbells (Danminin and Ennio brands), whereas the proposed system achieved good accuracy (88.53%) in identifying botnet attacks from thermostat devices. The accuracies of the proposed system in detecting botnet attacks from security cameras were 87.19%, 89.23%, 87.76%, and 89.64%, with respect to accuracy metrics. Overall, the CNN-LSTM model was successful in detecting botnet attacks from various IoT devices with optimal accuracy.

1. Introduction

The fourth industrial revolution, as described by Klaus Schwab, was built on the great achievements of the third revolution, especially the Internet, enormous processing capacity, the ability to store information, and the unlimited potential for access to knowledge [1]. Today, these achievements open the doors to unlimited possibilities through major breakthroughs of emerging technologies in the field of artificial intelligence, robotics, the Internet of Things, autonomous vehicles, 3D printing, nanotechnology, biotechnology, materials science, quantum computing, block chain, and others. The Internet of Things (IoT) aims to interconnect thousands of smart objects/devices in a seamless manner by sensing, processing, and analyzing large amounts of data obtained from heterogeneous IoT devices [2]. The IoT is recognized as one of the Gartner top 10 strategic technology trends in 2020, which projected that IoT will be used to develop 20 times more smart devices than conventional IT devices in 2023 [3]. According to Gartner, the overall usage of IoT in various areas, such as utilities, healthcare, government, physical security, and vehicles, is expected to increase [4]. This rapid development of infrastructure for the Internet of Things comes at the cost of numerous attacks and increased security threats. Symantec reported that every two minutes an IoT device is attacked [5]. Furthermore, Kaspersky reported [6] collecting 121,588 malware samples that had attacked IoT devices in 2018; this indicates that attacks averaged around four times more than in 2017 [7]. There are several types of malware that access IoT devices, such as BASHLITE and Mirai, which are strong and dangerous to the IoT infrastructure, because of it being accessible to vulnerabilities and known authentication authorizations. In 2016, 2.5 million IoT devices were infected

by Mirai attacks [8]. BASHLITE and Mirai attacks have features similar to distributed denial-of-service attacks (DDoS), which are carried by devices that are connected to the Internet.

According to [9–11], Owari, Mirai, and BASHLITE are botnet attacks that have risen in popularity. Botnet attacks are used to run bots on all devices that connect to the Internet and control by employing command and control (C&C) [12]. A botnet attack is a very serious attack known for spreading rapidly between devices connected to the Internet. There are major gaps in previous technologies for finding appropriate and effective mechanisms to protect IoT devices from botnet attacks. The intrusion detection system (IDS) is one solution for dealing with botnet attacks. It uses artificial intelligence for discovering new patterns of botnet attacks. The IDS is divided into two types: the anomaly and misuse methods. These types depend on being signature based. There are numerous IDSs available, such as Snort [13] and Suricata [14].

Currently, artificial intelligence (AI) algorithms are used to detect IoT attacks with more assured detection. Artificial intelligence technology even has the ability to detect variances in channels and methods of attacks. This was one of the challenges faced by security solutions for dealing with IoT attacks: hackers make small changes in previous attacks that security solutions are unable to detect. Developers and researchers use AI technologies for preventing any threats to the IoT environment by analyzing network traffic [15, 16]. Deep learning and machine learning have been built into security systems to detect such attacks efficiently. Deep learning is one of the artificial intelligence advances that are present in many real-life applications to handle complex nonlinear data. Deep recurrent neural network (DRNN) has been implemented to identify botnet attacks from IoT devices [17–19].

In this research, we present the convolutional neural network and long short-term memory (CNN-LSTM) model to detect botnet attacks from selected IoT devices. The proposed system differs from existing systems by training full datasets. Most researchers have used feature selection to select the most significant features for improving accuracy, but our system has achieved better accuracy by using all the training data. The main innovations of this study are as follows:

(a) Using advanced artificial intelligence algorithms such as CNN-LSTM to detect serious botnet attacks against the nine IoT devices infection by ten attacks

(b) The proposed system has attained good accuracy by training all input samples

(c) The system has the ability to analyze large amounts of data with good accuracy

(d) CNN-LSTM has the ability to detect any botnet attack from any IoT device

2. Related Works

Numerous researchers have focused on developing efficient frameworks to detect botnet attacks and protect the IoT environment. However, botnet attacks represent most of the DDoS attacks that infect IoT devices. The intrusion detection system is a powerful mechanism that is used to protect network systems against any malicious activities. The proposed system can help detect new attack batching by matching with signature attacks. Intrusion detection has two main methods, anomaly-based detection and signature-based detection, that detect attacks by extracting unknown patterns from network datasets.

Al-Garadi et al. [20] applied a deep learning algorithm for designing numerous applications, such as image recognition, localization, and security. Xie et al. [21] demonstrated an intrusion detection system for developing smart cities by using a short-term memory neural network (LSTM-NN) and multilayer perceptron (MLP) models. It is noted that the LSTM-G-NB has the highest accuracy. Alam et al. [22] introduced significant classification algorithms that can be used in IoT environments: support vector machine (SVM), K-nearest neighbor (KNN), and naive Bayes (NB). It is observed that the linear discriminant analysis (LDA) provides better results in terms of time. In this research [23], we developed a novel framework based on machine learning and deep learning to detect anomalies. The authors have used the pros and cons of the existing methods. An advanced algorithm has been proposed to make a paradigm in the security system. The authors' target is to improve the existing system by focusing on detecting attacks from the network layer. The authors [24] proposed a convolutional neural network (CNN) model based on a system of detecting intrusions from wireless networks. The results of the CNN model have achieved the highest accuracy and low false positive rate.

An IoT malware attack is a DDoS that attacks IoT devices. Most of the IoT environment does not have any mechanism for automatic updation of the devices themselves; therefore, these attacks cause widespread malware. Setting up an IDS has become very necessary for protection against malware. HaddadPajouh et al. [23] used the long short-term memory (LSTM) classifier to detect malware attacks based on the IoT infrastructure. The authors used 100 samples of malware as training data. The accuracy of the system has reached up to 97%. McDermott et al. [25] suggested deep learning approaches to detect botnet attacks. The Mirai botnet was classified in the research study. Bidirectional long short-term memory (BLSTM) using recurrent neural network (RNN) models was considered as an appropriate approach for protecting systems against botnet attacks. The performance of LSTM has an accuracy of 99.51%, and BLSTM accuracy is 99.98%. Brun et al. [26] applied dense RNN to detect attacks. This system has the ability to detect various types of attacks, such as UDP flooding, TCP SYN flooding, sleep-deprivation attacks, barrage attacks, and broadcast attacks. Captured packets extract statistical sequence data. This study was developed in a 3G SIM card environment, with a lot of IoT devices connected to this network. Meidan et al. [27] employed packet-captured data from IoT devices; the environments of the IoT were a security camera, smoke detector, socket, thermostat, TV, and a watch. The random forest tree algorithm was suggested to detect unauthorized IoT devices;

since then, the proposed system obtained a metric of 94% with respect to accuracy. Doshi et al. [28] proposed KNN, a Lagrangian support vector machine (LSVM), decision tree (DT), random forest (RF), and neural network (NN) to predict denial-of-service (DoS) attacks from IoT traffic. The network feature was divided into stateless and stateful features: stateless features include packet size and protocol features, whereas the stateful features include bandwidth and packet headers, such as source and destination address. Hodo et al. [29] applied artificial neural network (ANN) algorithms to detect DDoS/DoS attacks based on the characteristics of host-based IDS and network-based IDS. The proposed system has obtained 99.4% accuracy.

Meidan et al. [30] proposed deep autoencoder for anomaly detection. The N-BaIoT dataset was considered. The system was developed for protecting IoT environments from botnet attacks. When the system is compared with various existing systems, such as SVM and decision tree algorithm, it is noted that the system has the ability to detect botnet attacks with successful results. HaddadPajouh et al. [23] applied an LSTM classifier with Advanced RISC Machine- (ARM-) based IoT applications. To test the LSTM classifier, the authors used 100 examples of malware data not used in model training. The proposed model offered 97% average accuracy.

A study [31] used traditional machine learning, such as linear nearest neighbor lasso step (LNNLS-KH), to extract significant features for enhancing a system. The LNNLS-KH method is used to renew krill herd position to obtain the optimal global solution. Another study [32] used a features selection method, namely, wrapper and filter-based method, to handle dimensionality reduction to improve the classifiers process. The outputs from feature selection methods are processed using Bayesian networks (BN) and C4.5 algorithms. The authors employed the KDD CUP 99 network dataset to examine the proposed system. The designing of an efficient intrusion detection system can be completed by using numerous advanced artificial intelligence algorithms, such as nature-inspired computation intelligence [33–36] and other methods of machine learning [37–39].

3. Materials and Methods

In this section, the system architecture for developing system-based IoT botnet detection is presented. The system used is an example of an advanced artificial intelligence (CNN-LSTM) model to detect intrusion from IoT devices. The system was tested by employing real traffic data gathered from nine commercial IoT devices authentically infected by two common botnet attacks, namely, Mirai and BASHLITE. The system was set to recognize zero-day attacks from IoT devices to identify well-known attacks. Figure 1 shows the system architecture of the developing system. The main components of the proposed system are described in the next section.

3.1. N-BaIoT Dataset. The N-BaIoT dataset was collected from a machine-learning repository. The network data consisted of 155 features gathered from port mirroring of switch devices in IoT environments. The dataset was generated from real network traffic, including nine commercial IoT devices; 23 main features were extracted at different time intervals (100 ms, 500 ms, 10 s, 10 min, and 1 min). Table 1 displays nine commercial devices used to extract network traffic, including botnet attacks. The dataset has two main attacks, namely, Mirai and BASHLITE (https://archive.ics.uci.edu/ml/datasets/
detection_of_IoT_botnet_attacks_N_BaIoT).

Figure 2 displays the lab setup for collecting the botnet attacks from IoT devices. These devices were connected to Wi-Fi using many access point devices. Port mirroring has been set up on the switch devices for obtaining and sniffing real network traffic. The datasets were recorded using Wireshark software. Table 2 summarizes attack types in the dataset, including two common botnet attacks, namely, BASHLITE and Mirai. BASHLITE attacks, one type of botnet attack representing DDoS attacks, were developed using C programming for infecting Linux systems. This attack is the most common botnet attack that infects IoT devices, such as cameras. In contrast, Mirai botnet attacks, discovered in 2016 by Paras, use malware run on ARC processors to infect large-scale IoT networks.

3.2. Deep Learning Algorithms. Deep learning is one of the artificial intelligence algorithms used to handle analysis, complex processes, and big data. The deep learning model is applied to detecting botnet attacks from an IoT environment. In this proposed research, we have applied a multichannel CNN-LSTM deep learning model to identify and classify botnet attacks from different IoT devices.

3.2.1. Convolutional Neural Networks (CNN). CNN is a deep learning algorithm that is used to build an efficient system for image classification. However, the CNN model can also help design efficient systems for security purposes. The CNN algorithm is similar to the ordinary neural network: the CNN algorithm consists of four main layers, namely, the input layer, convolutional layer, pooling layer, and fully connected layer [41, 42].

(1) Convolutional Layer. The convolutional layer is used to explore, size, and filter the training sample, including numerous filters known as convolution kernels. The convolutional layer develops the weight matrix for the input sample and recodes the weighted summation kernel layer. The filter is integer values that are used to subset the input pixel values. Three significant hyperparameters, such as filter size, stride, and zero padding, play roles in increasing the performance of the convolutional kernels, choosing appropriate values that can help reduce the complexity of the neural network and increase the performance of the system. Figure 3 shows the details of the CNN algorithm layers. The input shape is (115, 1). We have used two values for filters, 64 and 128, with some kernel size, 5. The values of parameters are strides, 1, and padding, some. The convolutional layer is processed using

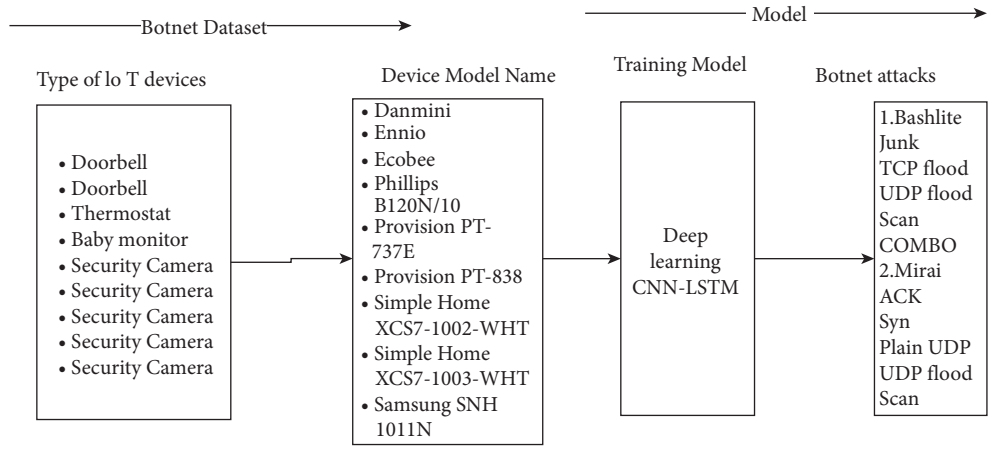


Figure 1: The proposed system.

$$x_i = f\left(w_i \otimes x_{i-1} + b_i\right), \tag{1}$$

where X is the sample of training input data, w_i is the weighted matrix, x_{i-1}: X is the sample of training input data, $\otimes$ is the convolution operation, f is the activation function, and b_i is the basis of neural network.

A rectified linear (ReLU) is a nonlinear activation function used to apply the element-wise activation function of a features map from convolutional layers. The ReLU function returns 0 for negative values, and for positive values, it returns any value x. Figure 4 shows the ReLU function: the ReLU function has a range from 0 to infinity [43]:

$$\mathrm{ReLU}(x) = \left[\, 0, \quad \text{if } x < 0 \quad x, \quad \text{if } x \geq 0 \,\right]. \tag{2}$$

(2) Pooling Layer. A pooling layer is used to reduce the number of parameters in the features map by selecting the maximum values in each region for designing a fit matrix average pooling. This matrix is processed into the next layer. We have considered the maximum pooling size of 5. Figure 5 shows the pooling layer.

$$Q_j = \mathrm{Max}\left(P_j^0, P_j^1, P_j^2, P_j^3, \ldots, P_j^t\right), \tag{3}$$

where Q_j is the output results from the IoT cybersecurity dataset, j is the pooling region, Max is the operation, and P_j^t is the element of the pooling.

(3) Fully Connected Layer. The last layer of the convolutional neural network is represented by the fully connected layer. Each node in the fully connected layer is connected directly to each node in layers $(L-1)$ and $(L+1)$. There is not any connection between nodes in the same layer, in contrast with the traditional ANN [44]. Therefore, this layer takes a long training and testing time. At the same network, more than one fully connected layer can be used, as shown in Figure 6.

Table 1: IoT devices used for developing datasets.

Device type	Devices used in the model
Doorbell	Danmini
Doorbell	Ennio
Thermostat	Ecobee
Baby monitor	Phillips B120N/10
Security camera	Provision PT-737E
Security camera	Provision PT-838
Security camera	Simple Home XCS7-1002-WHT
Security camera	Simple Home XCS7-1003-WHT
Security camera	Samsung SNH1011N

3.2.2. Long Short-Term Memory (LSTM). The recurrent neural network (RNN) algorithm is one of the deep learning models used in many real-life applications. Figure 7 displays the structure of the RNN model, where x represents input and y represents classification output. The long short-term memory model is one type of RNN. The LSTM is used to process sequence data that have feedback connect dissimilar to standard feedforward neural networks.

The LSTM has three main gates: input gate, forget gate, and output gate. The input gate is used to store the training data in long-term memory. While the long-term memory initializes from the current input data, the short-term memory initializes from the previous time step. The input gate has filters used to extract training data and discard unuseful information, whereas the useful information passes into sigma function. The sigma function has two indicator values: 0 and 1. The 1 value indicates the values that are very important, while the 0 value indicates values that are unimportant. The output from the input layer is saved in long-term memory. The forget gate is one of most significant gates in the LSTM model. It is used to decide which information to save or discard, by multiplying the forget vector values by current input gate The output from the forget gate will be passed to the next cell to obtain a new version from long-term memory. Figure 8 shows the structure of the LSTM model.

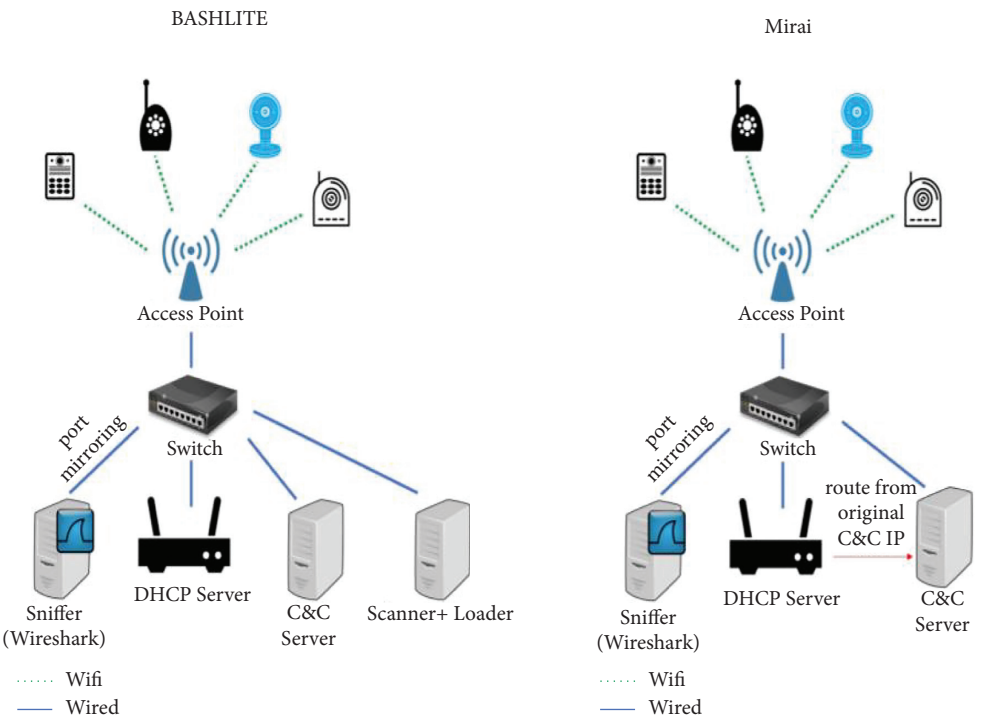


Figure 2: The lab setup for collecting the botnet attack from IoT devices [40].

Table 2: Botnet attacks.

Major attacks	Subattacks	Description
BASHLITE	Junk	By sending spam data
	TCP flood	Sends flood of request
	UDP flood	Sends flood of request
	Scan	Scans the network for victim devices
	COMBO	Opens connection IP address and network port by sending spam data
Mirai	ACK	Sends flood of acknowledgment
	SYN	Sends synchronize-packet-flood
	Plain UDP	UDP flood by optimizing seeding packet per second
	UDP flood	Scans the network for victim devices
	Scan	

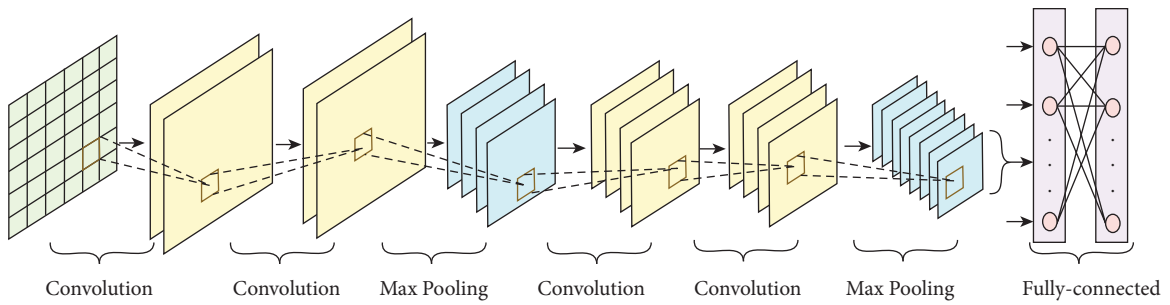


Figure 3: A generic architecture of a convolutional neural network (CNN).

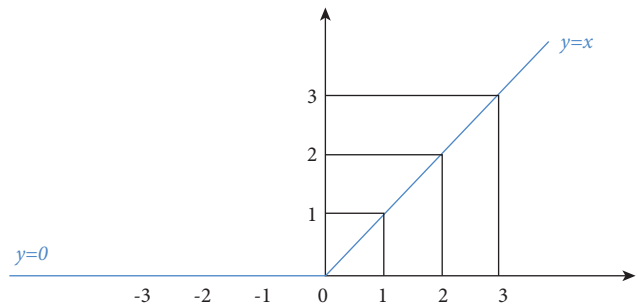


Figure 4: ReLU function.

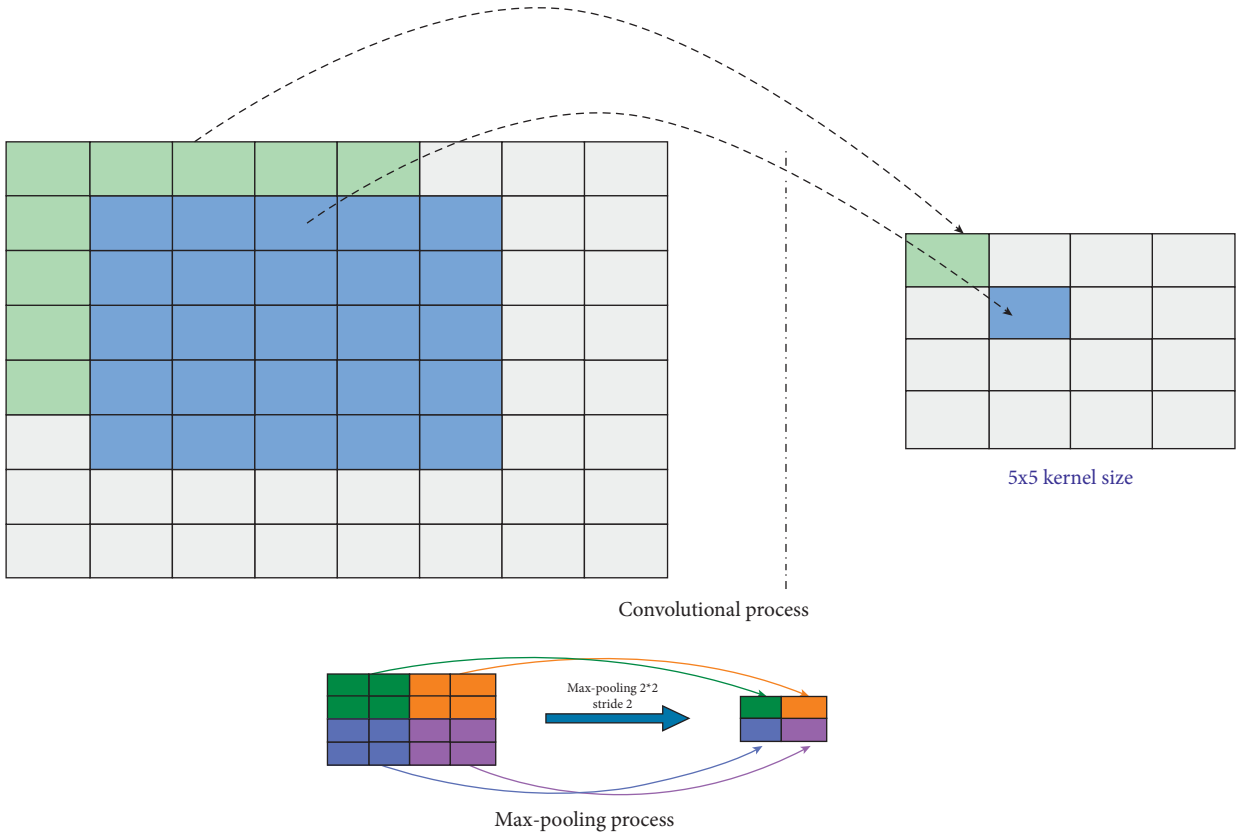


Figure 5: Max-pooling layer.

$$
\begin{aligned}
f_t &= \sigma\left(w_f\left[h_{t-1}, x_t\right] + b_f\right),\\
i_t &= \sigma\left(w_i\left[h_{t-1}, x_t\right] + b_i\right),\\
\widetilde{C}_t &= \tanh\left(w_C\left[h_{t-1}, x_t\right] + b_C\right),\\
C_t &= f_t * C_{t-1} + i_t * \widetilde{C}_t,\\
o_t &= \sigma\left(w_o\left[h_{t-1}, x_t\right] + b_o\right),\\
h_t &= o_t * \tanh\left(C_t\right),
\end{aligned}
\tag{4}
$$

where i_t is the output values for input layer, W is the weight values, and b is the bias. The σ activation function is used to transfer the important information to the next cell. f_t is the output from the forget gate, O_t is the output gate, c_t is the cellular cell, x_t is the input information, and h_t is the output information. Unlike standard feedforward neural networks, LSTM has feedback connections. It can process not only single data points but also entire sequences of data.

In this research, we have hybridized the CNN and LSTM models to detect botnet attacks from various types of IoT devices. Figure 9 displays a generic structure of the hybrid CNN-LSTM model that was used in our study.

The main components of the proposed system to detect botnet attacks from IoT devices are presented in Table 3. We have put the size of kernel convolution as 5, and epochs system was 20. The ReLU function was used as the activation function. A snapshot of the CNN-LSTM model is presented in Figure 10.

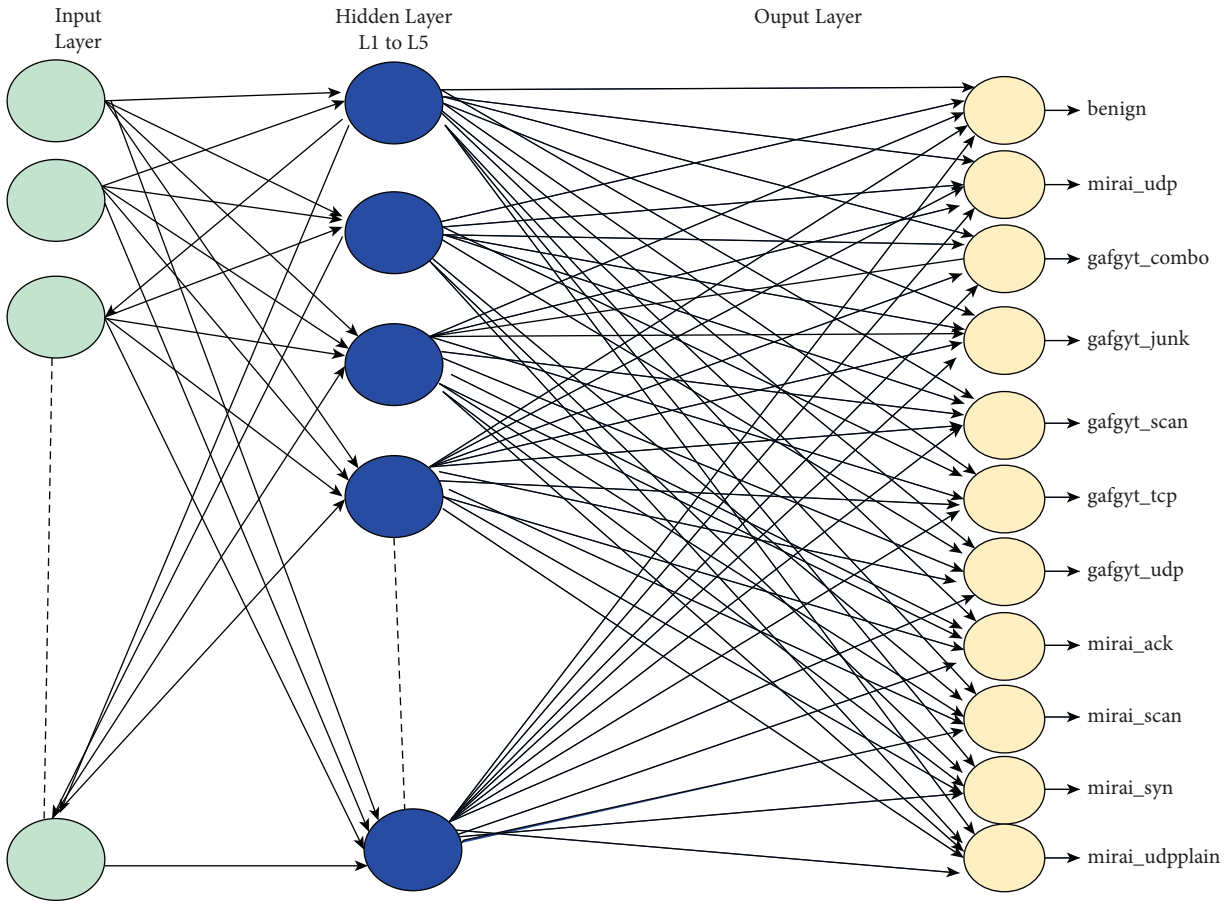


Figure 6: Fully connected layer.

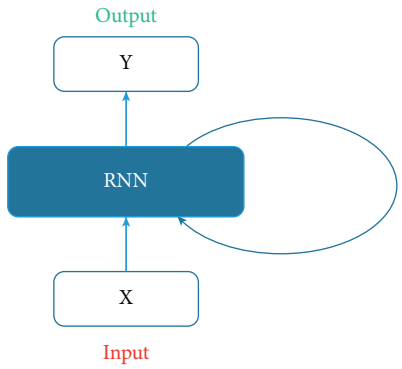


Figure 7: Structure of RNN.

4. Experimental Results

In this section, the results of the proposed system to detect botnet attacks are presented.

4.1. Experiment Environment Setup. The proposed research was completed using different software and hardware environments. Table 4 shows the requirements used to develop the proposed system. It was noted that these requirements were appropriate for developing a system to detect botnet attacks from IoT devices.

4.2. Evaluation Metrics. Accuracy, recall, precision, and $F1$-score metrics were considered to test the system for detection of botnet attacks. The equations are defined as follows:

$$\begin{aligned}
\text{accuracy} &= \frac{\text{TP} + \text{TN}}{\text{FP} + \text{FN} + \text{TP} + \text{TN}} \times 100\%, \\
\text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\%, \\
F1 - \text{score} &= 2 * \frac{\text{precision} \times \text{sensitivity}}{\text{precision} + \text{sensitivity}} \times 100\%, \\
\text{sensitivity} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%, \\
\text{recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%,
\end{aligned} \tag{5}$$

where TP is true positive, FP is false positive, TN is true negative, and FN is false negative.

4.3. Results and Discussion. To evaluate and examine the proposed system, five experiments were conducted on different IoT platforms. The machine learning and deep learning algorithms were implemented to detect botnet

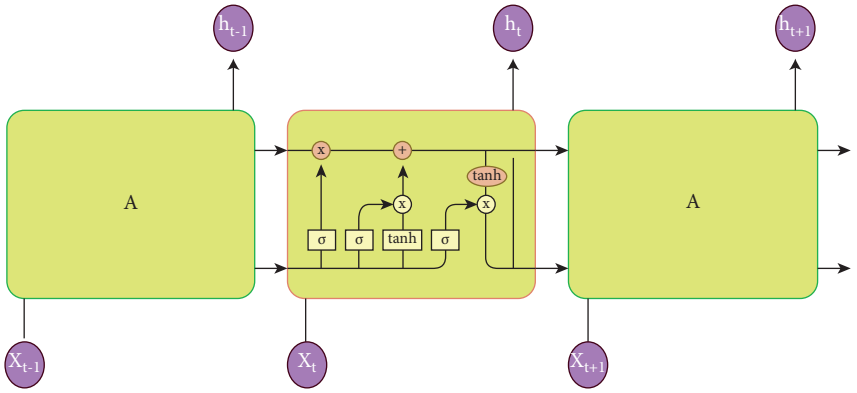


Figure 8: Structure of the LSTM model.

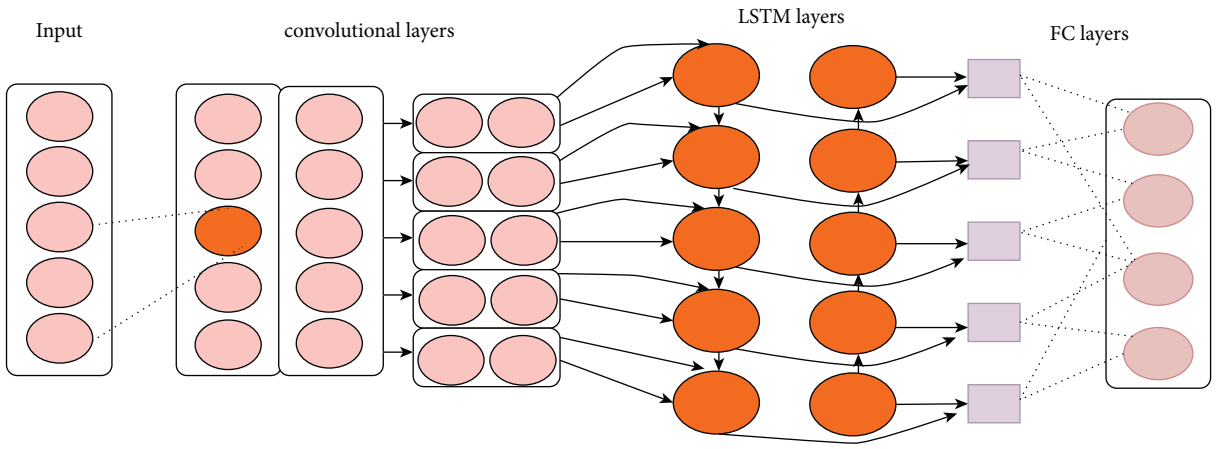


Figure 9: Structure of the hybrid CNN-LSTM model.

Input training data: M is number of network features for N instance for network traffic
Output: is label attacks or normal Y
For each N_i for Y
|
C_i= CNN (N_i) process
|
End
For each C_i process
Q_i = LSTM (C_i) process
|
End
For each Q_i process
Y = sofmax (Q_i) end

Algorithm 1: CNN-LSTM.

attacks by using a network dataset extracted from an IoT setup. In order to validate the system, the datasets were divided into 20% testing data and 70% training data. Table 5 shows input samples for nine commercial IoT devices, including botnet attacks. The five experiments' details are presented in the next section.

4.3.1. Experiment 1: Doorbell Devices. The CNN-LSTM model was applied to detect the anomaly from network data extracted from doorbells (Danminin and Ennio). Table 6 shows the results of the hybrid CNN-LSTM model. The weighted averages of the performance of the proposed system in detecting attack anomalies from the doorbell

Model : "sequential"

Layer (type)	output Shape	Param #
conv1d (Conv1D)	(None, 115, 64)	384
conv1d_1 (Conv1D)	(None, 115, 128)	41088
1stm (LSTM)	(None, 115, 64)	49408
1stm_1 (LSTM)	(None, 115, 32)	12416
flatten (Flatten)	(None, 3680)	0
dense (Dense)	(None, 128)	471168
dense_1 (Dense)	(None, 256)	33024
dense_2 (Dense)	(None, 11)	2827

Total params : 610, 315
Trainable params : 610, 315
Non-trainable params : 0

Figure 10: Snapshot of the CNN-LSTM model.

Table 3: Parameters of the CNN-LSTM model.

Parameters	Value
Convolution filters	100
Kernel size of filter	5
Fully connected layer	256
Activation function	ReLU
Classification function	Softmax
Optimizer	RSMprop
Epochs	20

Table 4: Experiment environment setup.

Hardware\software	Environment
Operating system	Windows 10
CPU	I7
Memory	8
Development environment	Jupyter Python 3.6
Matplotlib	Version 3.2.0
NumPy	Version 1.18.1
Pandas	Version 1.01
Scikit-learn	Version 0.22.1
Keras	Version 2.3.1
TensorFlow	Version 2.10

(Danminin) are 93, 91, and 88% with respect to precision, recall, and *F*1-score metrics, whereas the weighted averages of the proposed system for detecting intrusions from the doorbell (Ennio) are 91% (precision), 89% (recall), and 85% (*F*1-score).

Utilizing confusion metrics parameters, namely, true positives, false negatives, true negatives, and false positives, to detect the botnet attacks, Figure 11 shows the confusion metrics of the training model for identifying the pattern of unknown botnet attacks from Danminin and Ennio devices.

Figure 12 shows accuracy performance of the CNN-LSTM model for identifying intrusion from Danminin and Ennio devices. The accuracy of the proposed system in detecting ten attacks and benign traffic from Danminin devices is presented; it is noted that performance begins at approximately 84% and reaches 91%, whereas accuracy for Ennio of the proposed model starts at 74%, growing to 89% with 20 epochs.

Figure 13 shows the cross-entropy loss of CNN-LSTM when training Danminin and Ennio devices. Figure 13(a) shows the training loss of the system to detect the attacks from Danminin devices; the training loss has been reduced from 20.0 to 0.13. Figure 13(b) shows the training loss reduced from 20.0 to 0.17 in detecting intrusion from Ennio devices.

4.3.2. Experiment 2: Thermostat Device. In this experiment, we have implemented the hybrid CNN-LSTM model to detect intrusion from data extracted from a thermostat device. Table 7 summarizes the results of the CNN-LSTM for detecting botnet attacks. The weighted averages of evaluation metrics are 94%, 89%, and 85% for precision, recall, and *F*1-score metrics, respectively.

Figure 14 shows the confusion metrics of CNN-LSTM in classifying botnet attacks from the network data that were extracted from thermostat devices. It is observed that the system detected most botnet attacks.

Figure 15 shows the performance of the CNN-LSTM model in identifying botnet attacks from thermostat devices that are set up in the IoT environment. Figure 15(a) shows that the accuracy of the CNN-LSTM model increases from 80% to 88.53% with 20 epochs. The training loss of the system is shown in Figure 15(b); it is noted that training loss is reduced from 20.0 to 0.16.

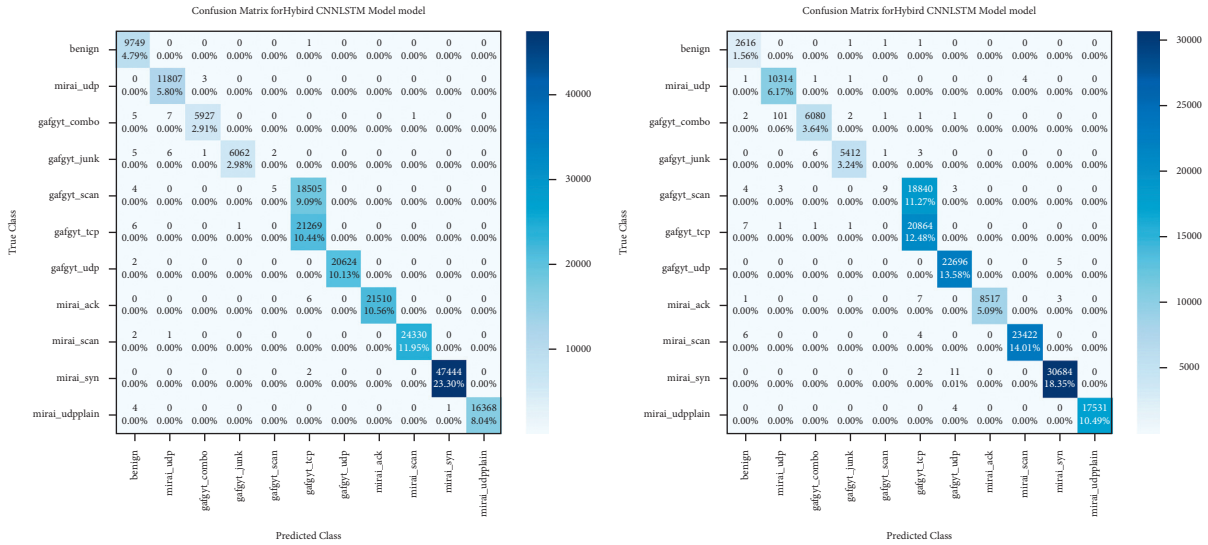
4.3.3. Experiment 3: Baby Monitor Device. In this experiment, we tested the CNN-LSTM model to detect intrusion from baby monitor (Philips B120N/10) IoT devices. The results of the proposed system are expressed in Table 8. From the optimal results, the system has achieved good accuracy in finding unknown patterns from datasets to handle botnet attacks. The weighted averages of the system are 93%, 92%, and 89% for precision, recall, and *F*1-score metrics, respectively. The confusion metrics obtained through using the CNN-LSTM model are presented in Figure 16. It is shown that the system has the ability to train all the botnet attacks.

Figure 17 shows the performance of CNN-LSTM in detecting botnet attacks from a baby monitor device. The accuracy has been increased from 84% to 92%, whereas the training loss decreases from 20.0 to 0.12 with 20 epochs.

4.3.4. Experiment 4: Security Camera Devices. In this section, we have applied a hybrid CNN-LSTM deep learning model to detect intrusion from security cameras, including four devices, namely, Provision PT-737E, Provision PT-838,

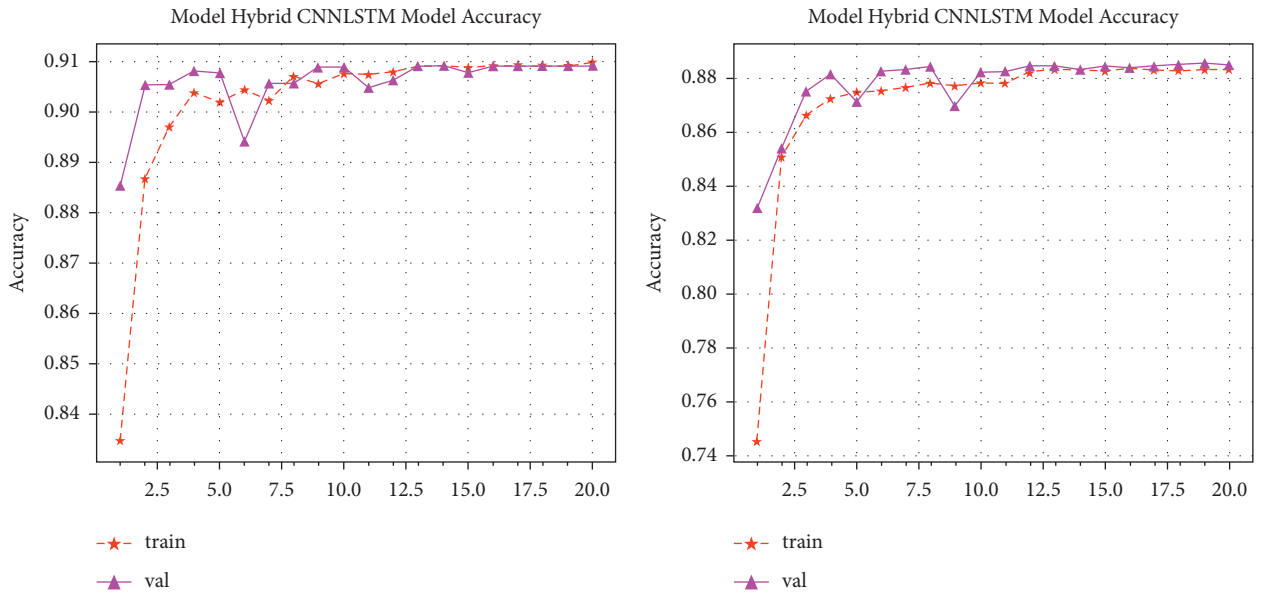
Table 5: The training samples of nine commercial IoT devices.

Major attacks	Subattacks	Training data of doorbell	Training data of baby monitor	Training data of security camera	Training data of webcam
BASHLITE	Junk	8624	8638	8788	8248
	TCP flood	27574	27886	26770	29217
	UDP flood	31932	31779	31371	30852
	Scan	8960	8286	8329	8624
	COMBO	17866	17241	17352	17923
Mirai	ACK	30754	27587	17449	32505
	SYN	36788	35525	18653	36507
	Plain UDP	24349	24258	16051	25083
	UDP flood	71330	64998	47514	47273
	Scan	32359	31037	29123	13164
Normal		14954	52369	29668	5852



(a) (b)

Figure 11: Confusion metrics of CNN-LSTN: (a) Danminin devices and (b) Ennio devices.



(a) (b)

Figure 12: Performance of the CNN-LSTM model to detect botnet attacks from doorbell devices: (a) Danminin and (b) Ennio.

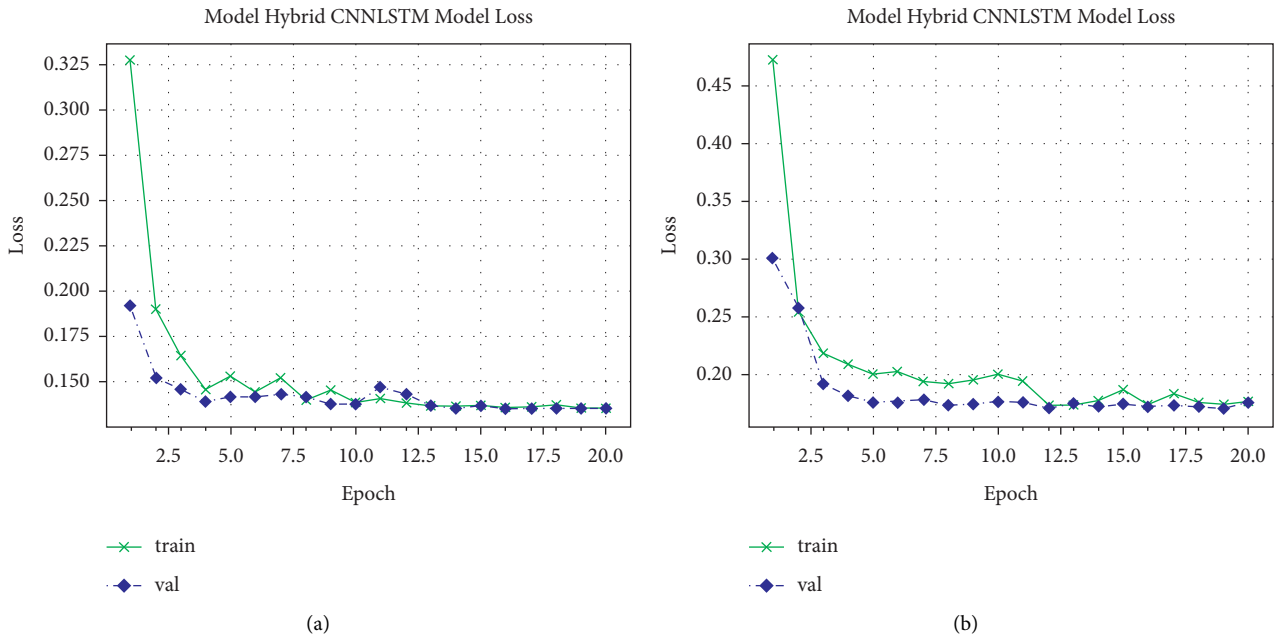


Figure 13: Training loss of the CNN-LSTM model to detect N-BaIoT attacks from doorbell devices: (a) Danminin devices and (b) Ennio devices.

Table 6: Performance of the CNN-LSTM model in detecting attacks from doorbell (Danminin and Ennio) IoT devices.

Attacks	Precision	Recall	*F*1-score
	Doorbell (Danminin)		
Benign	100	100	100
mirai_udp	100	100	100
COMBO	100	100	100
Junk	100	100	100
Scan	71	0.00	0.00
TCP	100	100	70
UDP	100	100	100
ACK	100	100	100
Mirai-Scan	100	100	100
Mirai-SYN	100	100	100
mirai_udpplain	100	100	100
Accuracy		90.88	
Weighted average	93	91	88
Loss		0.13	
	Doorbell (Ennio)		
Benign	99	100	100
mirai_udp	99	100	99
COMBO	100	98	99
Junk	100	100	100
Scan	75	0.00	0.00
TCP	53	100	69
UDP	100	100	100
ACK	100	100	100
Mirai-Scan	100	100	100
Mirai-SYN	100	100	100
mirai_udpplain	100	100	100
Accuracy	88.61		
Weighted average	91	89	85
Loss	0.17		

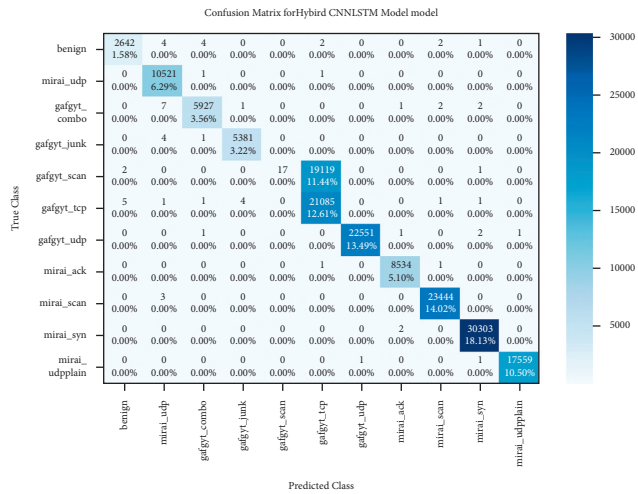


Figure 14: Confusion metrics of the CNN-LSTN model to detect botnet attacks from a thermostat device.

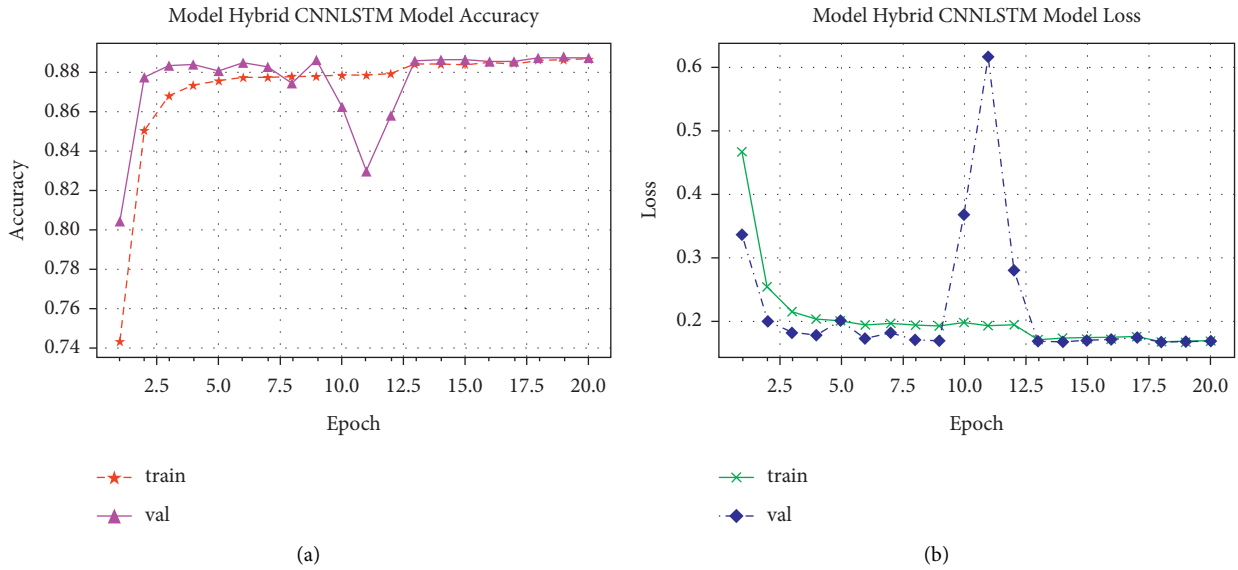


Figure 15: Performance of the CNN-LSTM model in detecting botnet attacks from thermostat devices: (a) accuracy and (b) training loss.

Table 7: Performance of the CNN-LSTM model in detecting attacks from thermostat IoT devices.

Attacks	Precision	Recall	*F*1-score
Benign	100	100	100
mirai_udp	100	100	99
COMBO	100	98	99
Junk	100	100	100
Scan	100	0.00	0.00
TCP	52	100	69
UDP	100	100	100
ACK	100	100	100
Mirai-Scan	100	100	100
Mirai-SYN	100	100	100
mirai_udpplain	100	100	100
Accuracy	88.53		
Weighted average	94	89	85
Loss	0.16		

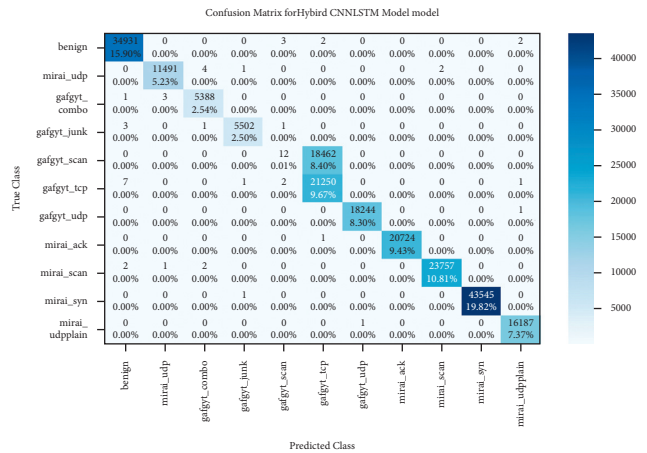


Figure 16: Confusion metrics of CNN-LSTN to detect botnet attacks from a baby monitor.

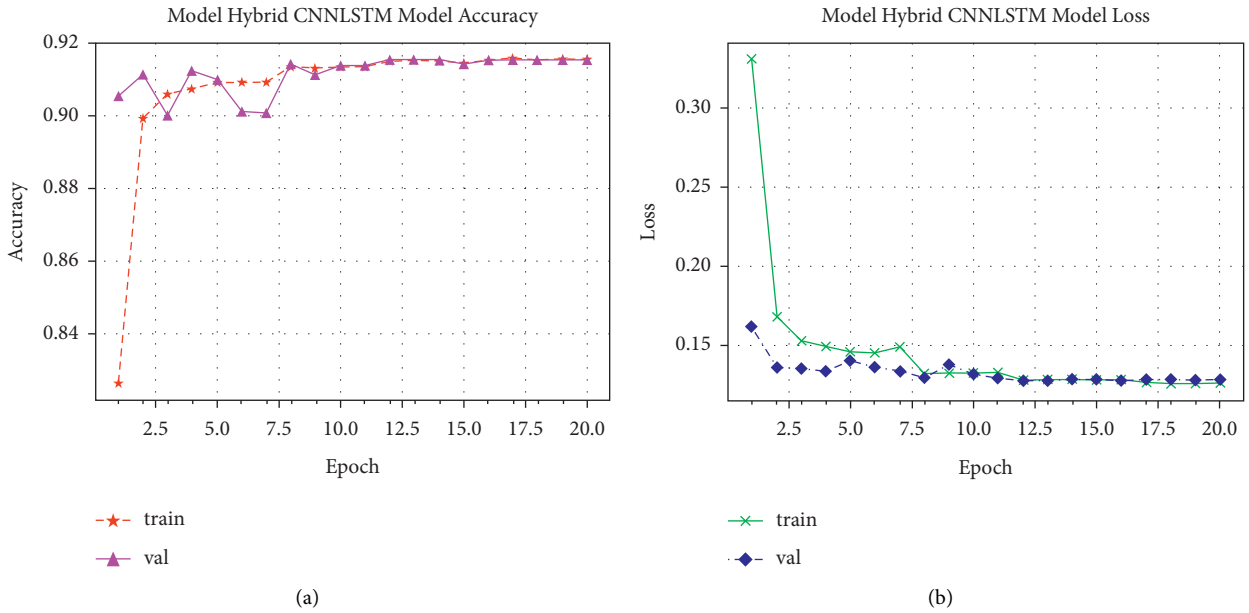


Figure 17: Performance of the CNN-LSTM model in detecting botnet attacks from baby monitor devices: (a) accuracy and (b) training loss.

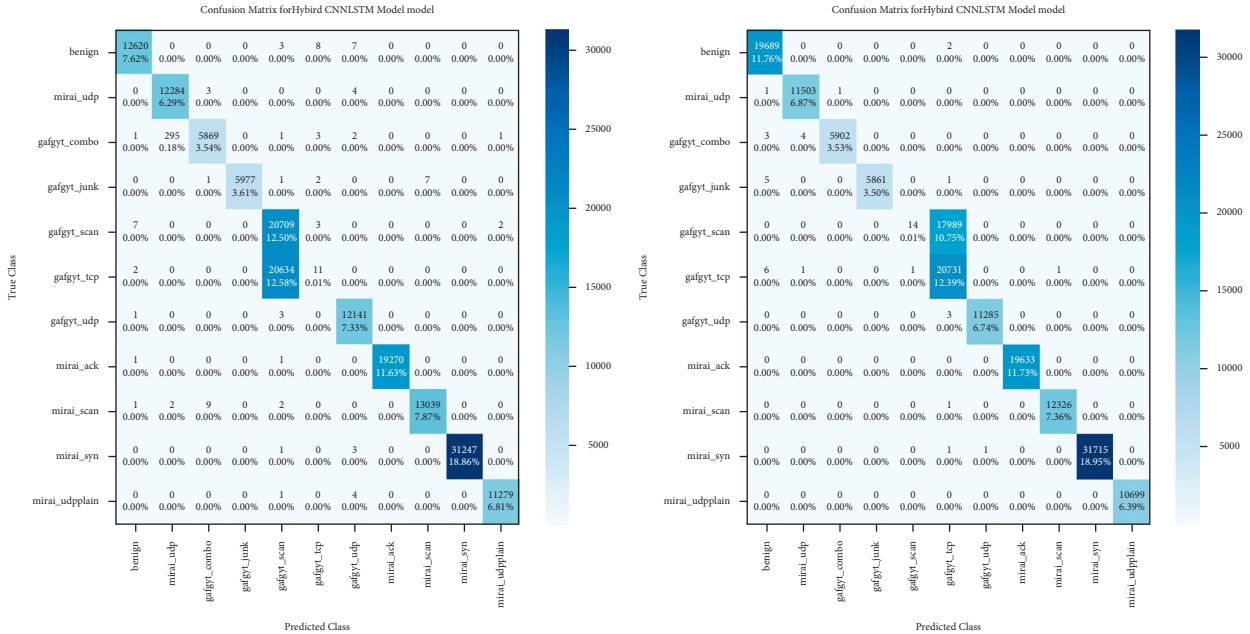
Table 8: Performance of the CNN-LSTM model in detecting attacks from baby monitor IoT devices.

Attacks	Precision	Recall	*F*1-score
Benign	99	100	100
mirai_udp	99	100	99
COMBO	100	98	99
Junk	100	100	100
Scan	67	0.00	0.00
TCP	54	100	70
UDP	100	100	100
ACK	100	100	100
Mirai-Scan	100	100	100
Mirai-SYN	100	100	100
mirai_udpplain	100	100	100
Accuracy		91.58	
Weighted average	93	92	89
Loss		0.12	

Simple Home XCS7-1002-WHT, and Simple Home XCS7-1003-WHT. Table 9 shows the results of the CNN-LSTM model in detecting botnet attacks from these devices when established in the IoT platform.

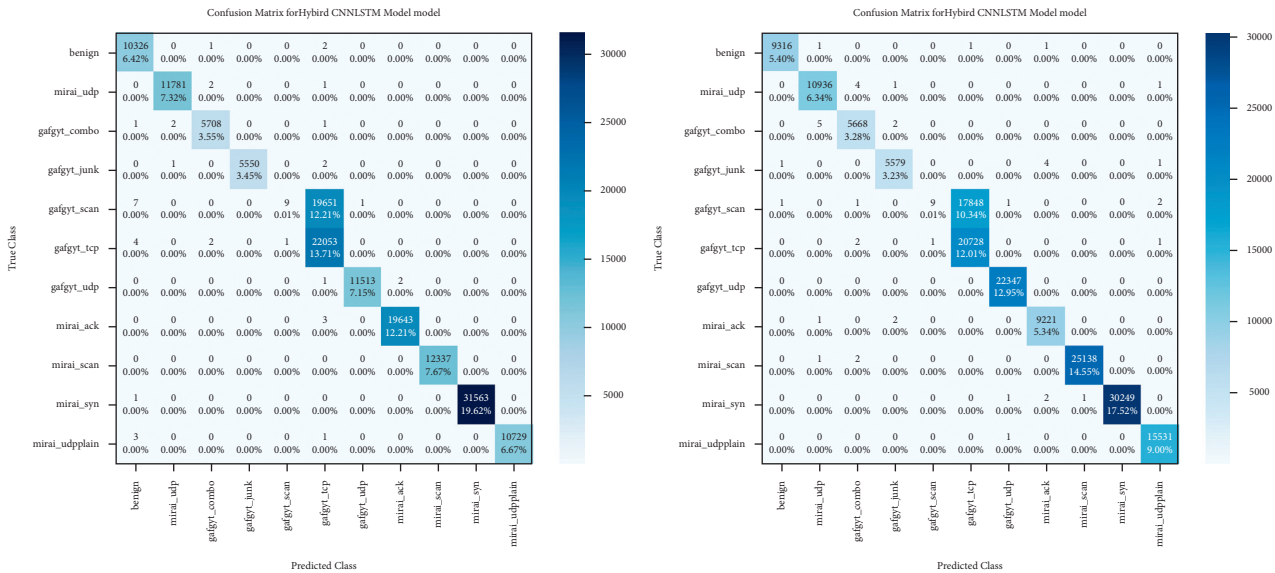
Figure 18 shows the confusion metrics of the system to classify the ten attacks and benign patterns from Provision PT-737E and Provision PT-838 security camera devices. The confusion metrics of the proposed system to detect attacks from Simple Home XCS7-1002-W and Simple Home XCS7-1003-WHT security camera devices are shown in Figure 19. We observed that the framework has achieved good accuracy in detecting most attacks from security cameras.

Accuracy performances of the CNN-LSTM model for developing a security system to detect attacks from security cameras in the IoT environment are demonstrated in Figure 20. The system has achieved good performance in



(a) (b)

Figure 18: Confusion metrics of CNN-LSTN: (a) Provision PT-737E device and (b) Provision PT-838 device.



(a) (b)

Figure 19: Confusion metrics of CNN-LSTM: (a) Simple Home XCS7-1002-W devices and (b) Simple Home XCS7-1003-WHT.

detecting attacks in Simple Home XCS7-1003-WHT: accuracy is increased from 78% to 90%, whereas the system has attained low accuracy (87.19%) in identifying attacks from Provision PT-737E devices.

The cross-entropy losses of the CNN-LSTM model training the dataset from security camera IoT devices are presented in Figure 21. Figure 21(a) shows that the training loss is reduced from 20.0 to 0.18 for extracting unknown attacks from a PT-737E device. The training loss of the CNN-LSTM model in detecting attacks from Provision PT-838 was 20.0–0.16, whereas Figure 21(c) shows CNN-LSTM reduced the loss from 20.0 to 0.18 when training data from the Simple Home XCS7-1002-W. Finally, the training loss of the system in detecting attacks from the Simple Home XCS7-1003-WHT is reduced from 20.0 to 0.15, as shown in Figure 21(d).

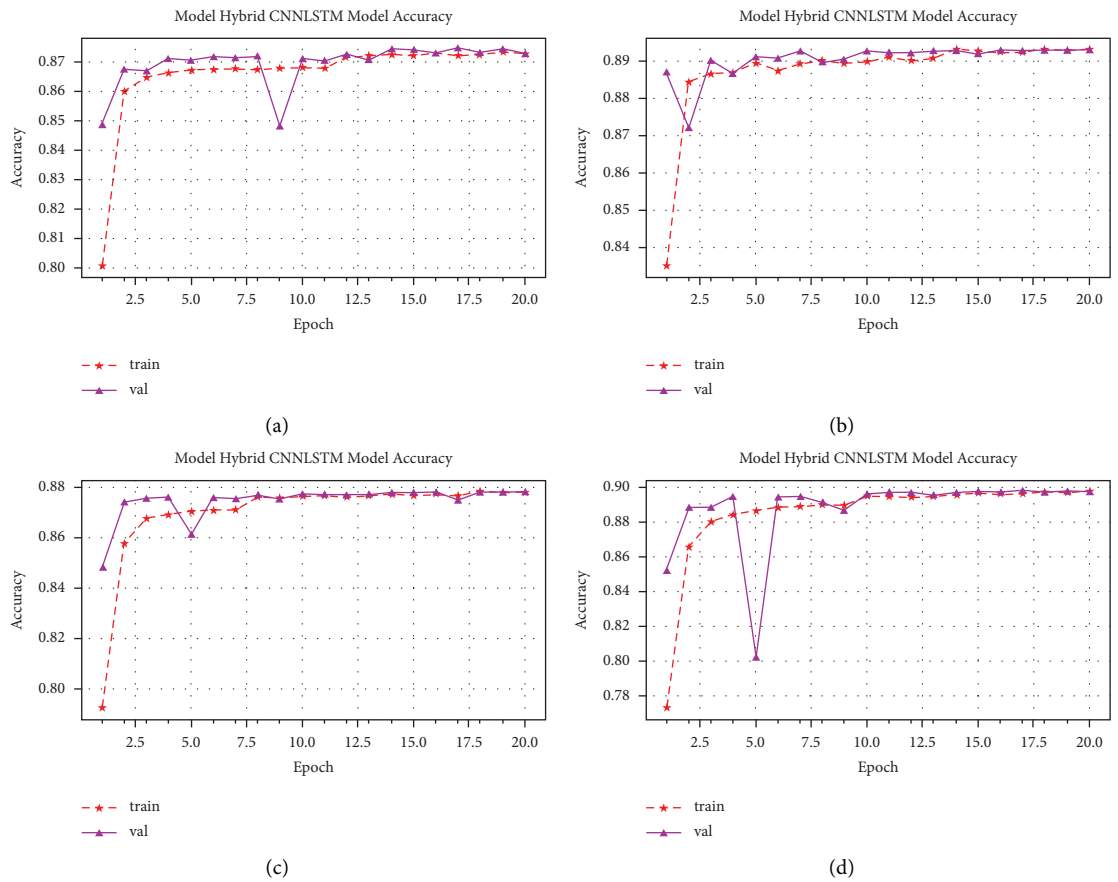


Figure 20: Performance of the CNN-LSTM model in detecting botnet attacks from security camera devices: (a) PT-737E, (b) Provision PT-838, (c) Simple Home XCS7-1002-W, and (d) Simple Home XCS7-1003-WHT.

4.3.5. Experiment 5: Webcam (Samsung). In this experiment, we have trained the CNN-LSTM model using data from webcam (Samsung SNH1011N) IoT devices. Table 10 shows the results of the CNN-LSTM model in detecting attacks from a webcam (Samsung SNH1011N). The weighted averages for the system are 94%, 88%, and 84% in terms of precision, recall, and *F*1-score metrics, respectively. Figure 22 shows confusion metrics of the CNN-LSTM in classifying the various types of botnet attacks.

The performance and training loss of the CNN-LSTM model in identifying attacks from a webcam IoT device are displayed in Figure 23. The performance of the system grows from 78% to 88%, as shown in Figure 23(a), whereas the training model loss is reduced from 20.0 to 0.16, as shown in Figure 23(b).

5. Discussion

Botnet attacks are one of the serious attacks that threaten IoT devices. As we know, most of our real-life applications are based on IoT technology. The attackers have developed batch files of botnet attacks for preventing security system devices from recognizing these attacks. This makes it difficult for technology companies to design zero-day security system devices to protect the IoT environment.

Therefore, using artificial intelligence models to detect botnet attacks, by extracting various unknown patterns that are developed by attackers, can easily help protect an IoT platform. In this research, we applied the CNN-LSTM model to detect botnets. This system was tested by a dataset generated from nine commercial device injections from ten attacks.

The results of the first experiment, to identify botnet attacks from doorbell IoT devices, are shown in Table 6. We observed the following points:

(1) The CNN-LSTM model achieved 100% with respect to precision, recall, and *F*1-score in detecting most attacks from a doorbell (Danminin version)

(2) The CNN-LSTM system showed low performance in detecting Scan attacks: precision, 71%; recall, 0.0; and *F*1-score, 0.0, from a doorbell (Danminin version)

(3) The system achieved good performance between 100–99% in detection of most of the attacks from a doorbell (Ennio version) device

(4) The system showed low performance in detecting Scan and TCP flood attacks; for Scan attacks: precision (75%) and recall and *F*1-score (0.00), whereas for TCP flood attacks: precision (53%) and *F*1-score (69%)

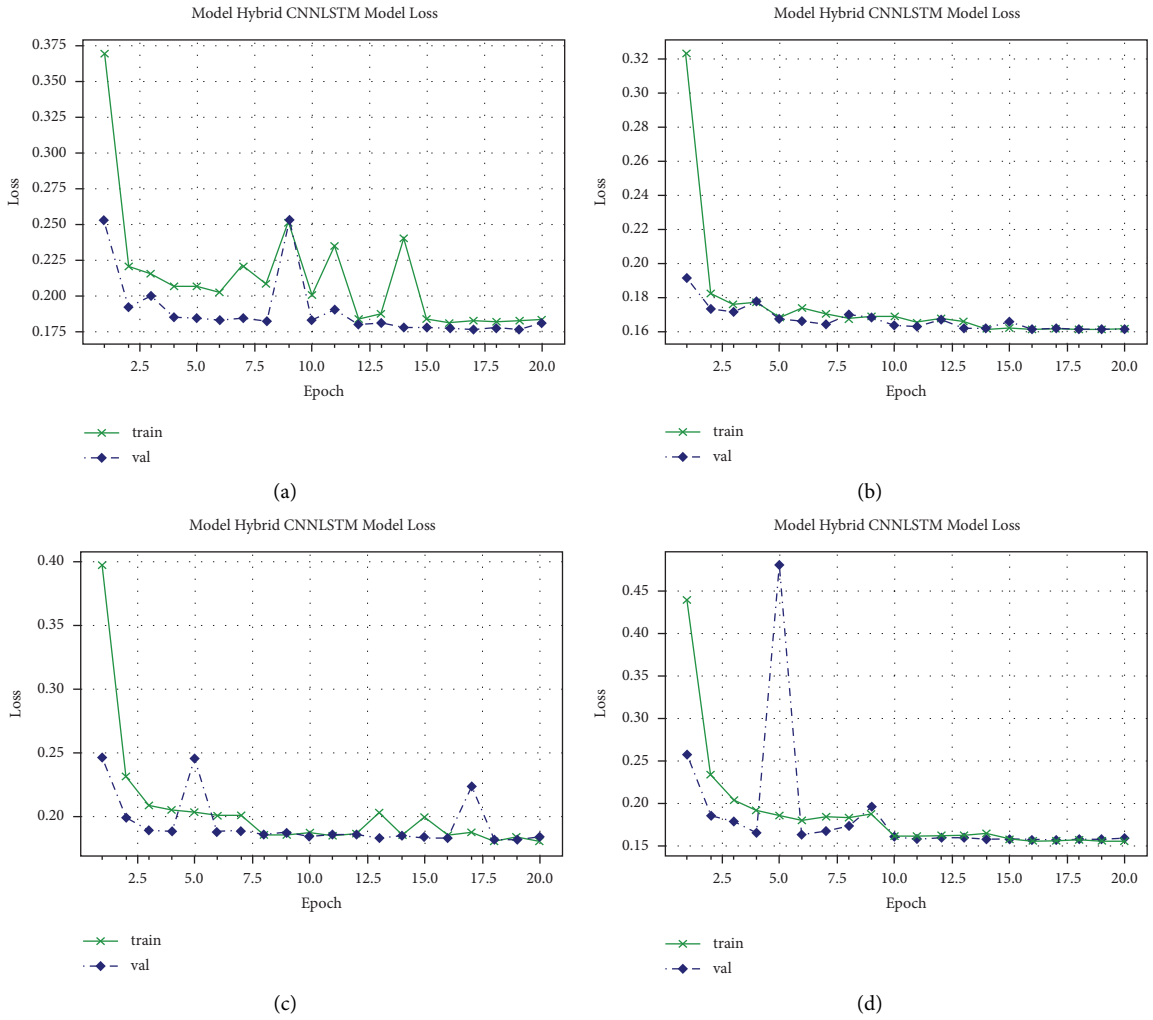


Figure 21: Training loss of the CNN-LSTM model in detecting N-BaIoT attacks from security camera devices: (a) Provision PT-737E, (b) Provision PT-838, (c) Simple Home XCS7-1002-W, and (d) Simple Home XCS7-1003-WHT.

(5) Overall, our proposed system provided good performance in detecting botnet attacks from Danminin and Ennio doorbell devices

In the second experiment, the CNN-LSTM model was used to detect botnet attacks from thermostat IoT devices, as shown in Table 7. We obtained the following points:

(1) The CNN-LSTM model has achieved good performance in detecting all the attacks except TCP flood attacks; the performance of the proposed system was 100% with respect to precision, recall, and *F*1-score metrics

(2) The proposed system achieved low performance, with precision 52% and *F*1-score 69%, in the detection of TCP flood attacks

In the third experiment, we used a baby monitor (Philips B120N/10) IoT device dataset to examine the proposed system, as summarized in Table 8. We observed the following points:

(1) The proposed system attained 100% performance in classifying most botnet attacks in terms of precision, recall, and *F*1-score metrics

(2) The CNN-LSTM model demonstrated low performance in detection of Scan attacks precision (67%) and recall and *F*1-score (0.00), and TCP flood precision (54%)

In the fourth experiment, we applied the CNN-LSTM system to detect botnet attacks from security camera devices (Provision PT-737E, Provision PT-838, Simple Home XCS7-1002-WHT, and Simple Home XCS7-1003-WHT), as shown in Table 9. We arrived at the following points:

(1) The CNN-LSTM system obtained optimal results in detecting most of the attacks, with 100% precision, recall, and *F*1-score metrics

(2) The proposed system shows low performance in detecting Scan and TCP flood attacks from security camera IoT devices

Table 9: . Performances of the CNN-LSTM model in detecting attacks from security camera IoT devices.

Attacks	Precision	Recall	*F*1-score
	Provision PT-737E		
Benign	99	100	100
mirai_udp	98	100	99
COMBO	100	95	97
Junk	100	100	100
Scan	50	100	0.67
TCP	41	0.00	0.00
UDP	100	100	100
ACK	100	100	100
Mirai-Scan	100	100	100
Mirai-SYN	100	100	100
mirai_udpplain	100	100	100
Accuracy		87.19	
Weighted average	86	87	83
Loss		0.18	
	Provision PT-838		
Benign	99	100	100
mirai_udp	98	100	100
COMBO	100	100	100
Junk	100	100	100
Scan	93	0.00	0.00
TCP	54	0.00	70
UDP	100	100	100
ACK	100	100	100
Mirai-Scan	100	100	100
Mirai-SYN	100	100	100
mirai_udpplain	100	100	100
Accuracy	89.23		
Weighted average	94	89	85
Loss	0.16		
	Simple Home XCS7-1002-W		
Benign	100	100	100
mirai_udp	100	100	100
COMBO	100	100	100
Junk	100	100	100
Scan	90	0.00	0.00
TCP	53	100	69
UDP	100	100	100
ACK	100	100	100
Mirai-Scan	100	100	100
Mirai-SYN	100	100	100
mirai_udpplain	100	100	100
Accuracy	87.76		
Weighted average	92	88	84
Loss	0.18		
	Simple Home XCS7-1003-WHT		
Benign	100	100	100
mirai_udp	100	100	100
COMBO	100	100	100
Junk	100	100	100
Scan	90	0.00	0.00
TCP	54	100	70
UDP	100	100	100
ACK	100	100	100
Mirai-Scan	100	100	100
Mirai-SYN	100	100	100
mirai_udpplain	100	100	100
Accuracy	**89.64**		
Weighted average	**93**	**90**	**86**
Loss	**0.15**		

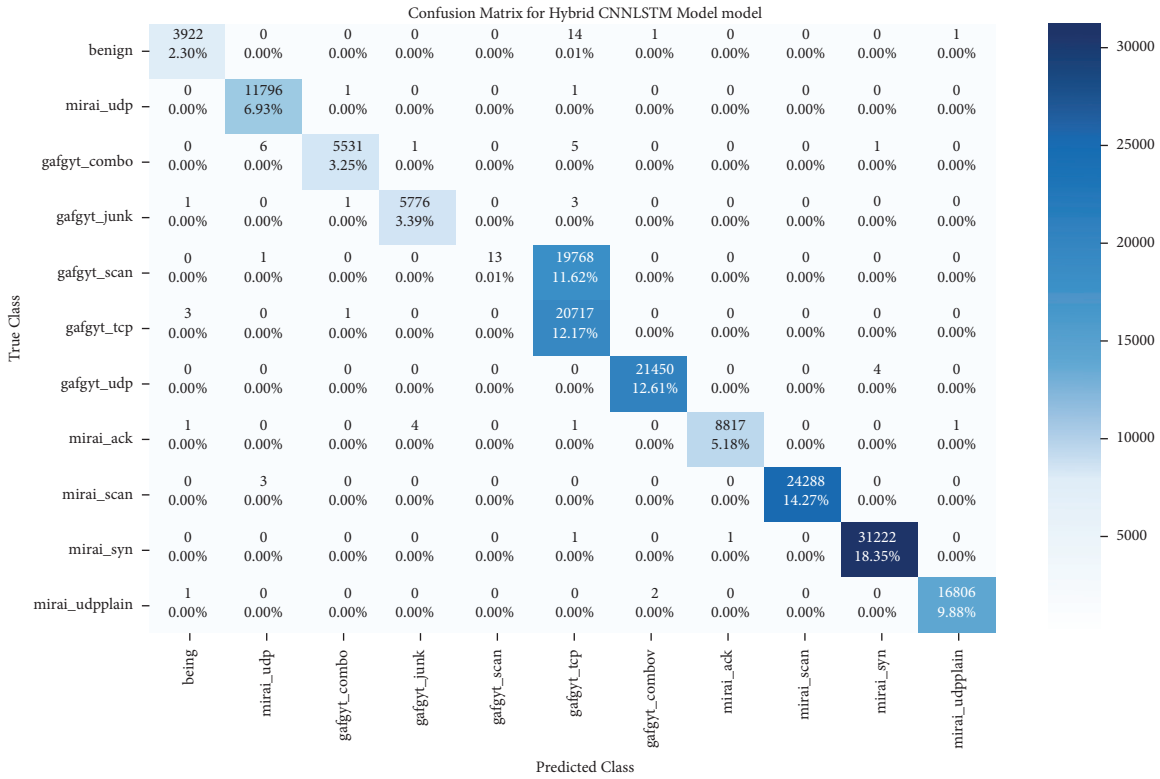
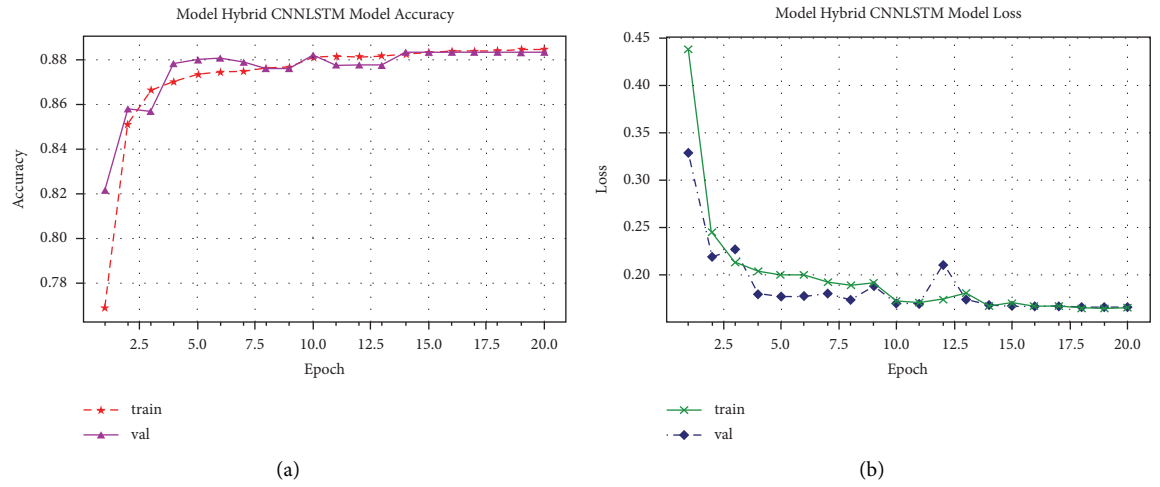


Figure 22: Confusion metrics of CNN-LSTN to detect botnet attacks from webcam device.



(a) (b)

Figure 23: Performance of the CNN-LSTM model in detecting botnet attacks from webcam devices: (a) accuracy and (b) training loss.

In the fifth experiments, we used dataset extract from webcam (Samsung SNH1011N) IoT devices to test our system; it is noted that the system obtained low performance in detecting scan attack with F1-score (0.00).

Figure 24 displays receiver operating characteristic (ROC) curves for simulation results of the CNN-LSTM model for detecting botnet attacks. The ROC is used to measure the validation of the proposed system to detect botnets from IoT devices. The graphical representation (y-axis) is the recall metric for detecting ten attacks and benign traffic in nine different commercial devices; x-axis represents the specificity metric for detecting all botnet attacks.

Overall, the CNN-LSTM model has the ability to detect botnet attacks from different IoT devices with optimal performance. The proposed system showed low performance in detecting Scan and TCP flood attacks. Curve and confusion metrics are presented and proved the effectiveness and efficiency of the system to detect botnet attacks from nine commercial IoT devices, including the ten most serious attacks that infect the IoT environment.

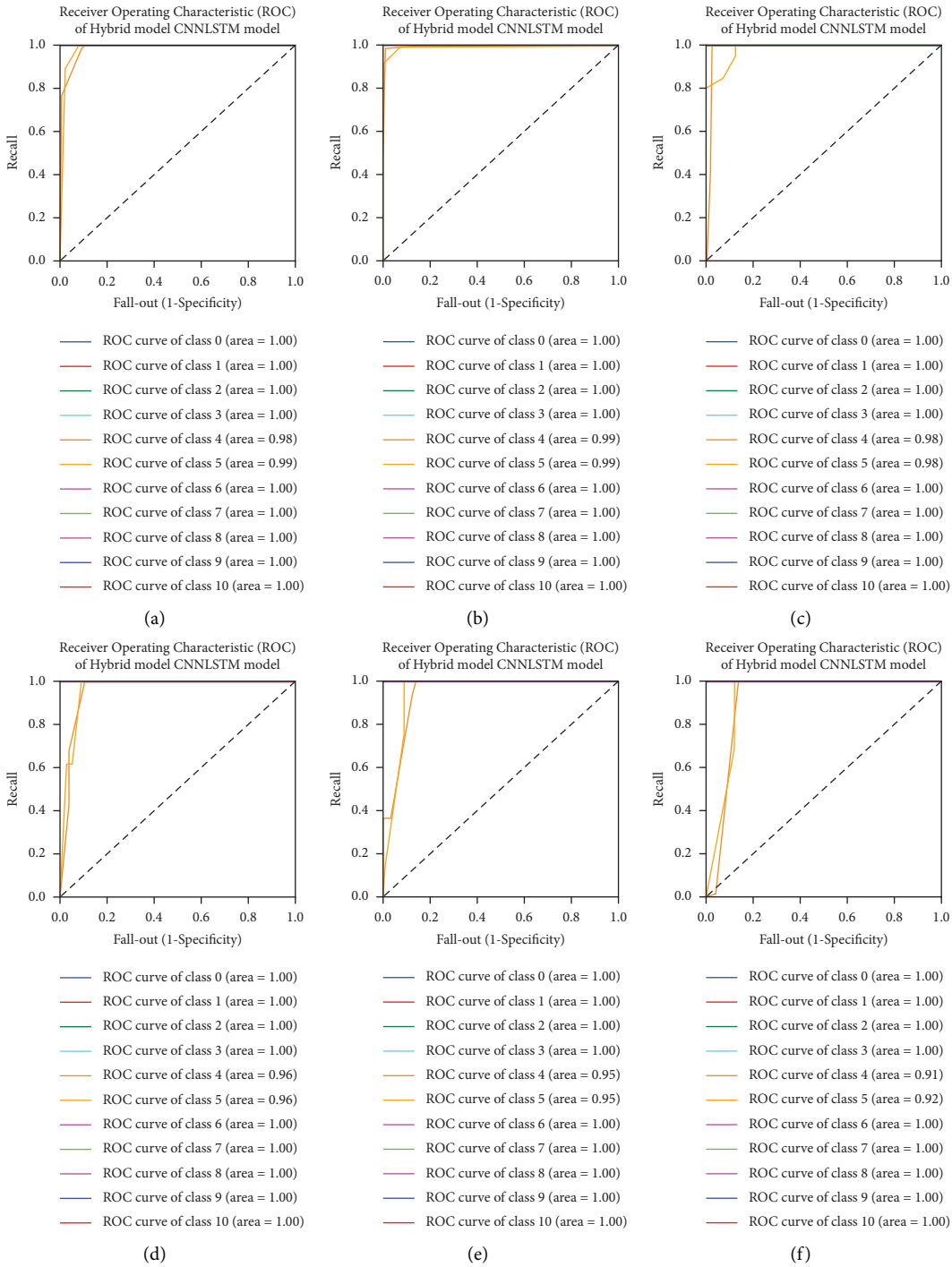


Figure 24: Continued.

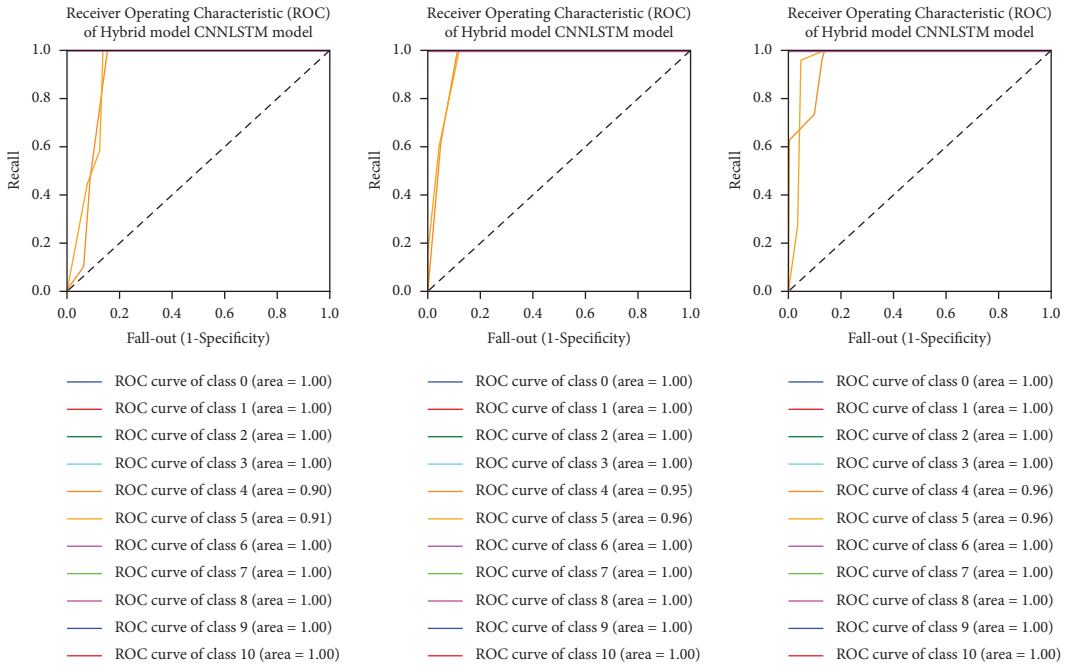


Figure 24: ROC curves of the CNN-LSTM model for detecting botnet attacks: (a) Danmini, (b) Ennio, (c) thermostat, (d) Phillips B120N/10, (e) Provision PT-737E, (f) Provision PT-838, (g) Simple Home XCS7-1002-WHT, (h) Simple Home XCS7-1003-WHT, and (i) Samsung SNH1011N.

Table 10: Performances of the CNN-LSTM model in detecting attacks from webcam (Samsung SNH1011N) IoT devices.

Attacks	Precision	Recall	*F*1-score
Benign	100	100	100
mirai_udp	100	100	100
COMBO	100	100	100
Junk	100	100	100
Scan	100	0.00	0.00
TCP	51	100	68
UDP	100	100	100
ACK	100	100	100
Mirai-Scan	100	100	100
Mirai-SYN	100	100	100
mirai_udpplain	100	100	100
Accuracy		88.35	
Weighted average	94	88	84
Loss		0.16	

Developing security system to detect the intrusion from IoT environment has played a pivotal role in protecting the IoT network. The CNN-LSTM deep learning algorithm was used to detect the botnet attacks. Table 11 summarizes CNN-LSTM model results against existing systems. There are few studies that have used some datasets to detect the botnet attack from IoT network. During research, we have found one study that used some attacks but different dataset.

Soe et al. [45] applied three machine learning algorithms, namely, naïve Bayes, J48, and artificial neural network (ANN), to detect botnet attacks from IoT devices; the dataset was different but attacks are similar. This study has used

Table 11: Comparison of the proposed system with existing models [45].

Attacks	Models	Accuracy
mirai_udp	Naïve Bayes	82.49
COMBO		77.33
Junk		61.52
Scan		76.11
TCP		85.81
UDP		82.41
ACK		85.81
mirai_udp	J48	99.09
COMBO		99.01
Junk		99.08
Scan		99.05
TCP		99.05
UDP		98.09
ACK		99.01
mirai_udp	ANN	99
COMBO		99
Junk		98.98
Scan		98.98
TCP		98.95
UDP		98.97
ACK		99.01
mirai_udp	Proposed model	100
COMBO		100
Junk		100
Scan		100
TCP		100
UDP		100
ACK		100
Mirai-Scan		100

feature selection to improve the accuracy of the machine learning. In this study, we have examined the proposed system with IoT device datasets extracted from nine devices. We have considered the highest accuracy results to compare against existing systems. Overall, we observed that the proposed system shows better performance.

6. Conclusion

We developed a system based on a deep learning algorithm to reduce the risks that IoT devices face from DDoS attacks. Identification of DDoS attacks in the early stages can help network security by speeding up operations to disconnect most of the IoT devices from Internet connections for preventing and stopping botnet attacks from accelerating. In this research, we have used the N-BaIoT dataset generated from nine commercial IoT devices, namely, Danmini, Ennio, Ecobee, Phillips B120N/10, Provision PT-737E, Provision PT-838, Simple Home XCS7-1002-WHT, Simple Home XCS7-1003-WHT, and Samsung SNH1011N, injected by two major IoT attacks, namely, BASHLITE and Mirai botnet attacks. The BASHLITE attack has subattacks that are Junk, TCP flood, UDP flood, Scan, and COMBO, whereas the Mirai attack is categorized into ACK, SYN, Plain UDP, UDP flood, and Scan. The main findings of this research are as follows:

(i) The hybrid CNN-LSTM model has successfully achieved good results compared with existing studies.

(ii) The proposed system has achieved low accuracy in detecting Scan and TCP flood attacks in terms of evaluation metrics.

(iii) The experimental results proved that the detection of botnet attacks depends on numerous training models rather than the type of IoT device. We believe that the proposed system based on CNN-LSTM can effectively enhance security in various types of IoT platforms by detecting botnet attacks.

(iv) The CNN-LSTM model has accomplished high results in detecting most botnet attacks.

(v) The main contribution of this study is to develop a framework by using advanced artificial intelligence to identify various unknown patterns from IoT devices to detect botnet attacks from various types of IoT devices effectively and efficiently. In the future, we will try to find ways to improve the detection of Scan and TCP flood attacks.

Data Availability

The public data are available at https://archive.ics.uci.edu/ml/datasets/detection_of_IoT_botnet_attacks_N_BaIoT.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

The authors extend their appreciation to the Deanship of Scientific Research at King Faisal University for the financial support under grant no. 216013.

References

[1] World Economic Forum, "Fourth Industrial Revolution," 2020, https://www.weforum.org/focus/fourth-industrial-revolution.

[2] K. Ashton, "That 'internet of things' thing," *RFiD J*, vol. 22, pp. 97–114, 2009, https://www.rfidjournal.com/articles/view?4986.

[3] Gartner, "Top 10 Strategic Technology Trends for 2020," 2020, https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2020.

[4] Gartner, https://www.gartner.com/en/newsroom/press-releases/2019-08-29-gartner-says-5-8-billionenterprise-and-automotive-io, 2020.

[5] Cisco, *Cisco Visual networking Index (VNI) global Mobile data traffic Forecast update, 2017–2022*, Cisco Systems Inc., San Jose, CA, USA, 2019.

[6] Broadcom, "Symantec Internet Security Threat Report 2019," vol. 24, 2020, https://docs.broadcom.com/doc/istr-24-2019-en.

[7] M. Kuzin, Y. Shmelev, and V. Kuskov, "New Trends in the World of IoT Threats—Securelist Kaspersky Lab," 2018, https://securelist.com/new-trends-in-the-world-of-iot-threats/87991/.

[8] A. Marzano, D. Alexander, O. Fonseca et al., "The Evolution of Bashlite and Mirai IoT botnets," in *Proceedings of the IEEE Symposium on Computers and Communications*, pp. 813–818, IEEE, Natal, Brazil, June 2018.

[9] Mohit kumar, "IoT botnets found using Default Credentials for C&C server Databases," 2020, https://thehackernews.com/2018/06/iot-botnet-password.html.

[10] Bankinfosecurity, "Massive botnet attack used more than 400,000 IoT devices," 2020, https://www.bankinfosecurity.com/massive-botnet-attack-used-more-than-400000-iotdevices-a-12841.

[11] Enigmasoftware, "BASHLITE Malware Hits Over One Million IoT Devices," 2020, https://www.enigmasoftware.com/bashlite-malware-hits-one-million-iot-devices/.

[12] Thingbots, "The Future of Botnets in the Internet of Things," 2020, https://securityintelligence.com/thingbots-the-future-of-botnets-in-the-internet-of-things/.

[13] Baker, A. R, Esler, and J. Snort Ids, *IPS Toolkit; 30 Corporate*, Elsevier Inc., Burlington, MA, USA, 2007.

[14] S. A. R. Shah and B. Issac, "Performance comparison of intrusion detection systems and application of machine learning to Snort system," *Future Generation Computer Systems*, vol. 80, pp. 157–170, 2018.

[15] Y. Meidan, M. Bohadana, A. Shabtai, J. D. Guarnizo, M. Ochoa, and N. O. P.IoT. Tippenhauer, "A machine learning approach for IoT device identification based on network traffic analysis," in *Proceedings of the Symposium on Applied Computing*, pp. 506–509, Association for Computing Machinery, Marrakech, Morocco, 4–6 April 2017.

[16] H. Alkahtani, T. Aldhyani, and M. Al-Yaari, "Adaptive anomaly detection framework model objects in cyberspace," *Applied Bionics and Biomechanics*, vol. 2020, Article ID 6660489, 14 pages, 2020.

[17] M. A. Ferrag and L. D.C. Maglaras, "A novel deep learning and Blockchain-based Energy Exchange framework for smart Grids," *IEEE Transactions on Engineering Management*, vol. 67, no. 4, 2019.

[18] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, Article ID 102419, 2020.

[19] O. Alkadi, N. Moustafa, B. Turnbull, and K. K. R. Choo, "A deep Blockchain framework-enabled Collaborative intrusion detection for protecting IoT and Cloud networks," vol. 8, no. 12 IEEE Internet Things J, 2020.

[20] M. A. Al-Garadi, A. Mohamed, A. Al-Ali, X. Du, I. Ali, and M. Guizani, "A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, 2018.

[21] X. Xie, D. Wu, S. Liu, and R. Li, "IoT Data Analytics Using Deep Learning," 2017, https://arxiv.org/abs/1708.03854.

[22] F. Alam, R. Mehmood, I. Katib, and A. Albeshri, "Analysis of eight data mining algorithms for smarter internet of things (IoT)," *Procedia Computer Science*, vol. 98, pp. 437–442, 2016.

[23] H. HaddadPajouh, A. Dehghantanha, R. Khayami, and K.-K. R. Choo, "A deep recurrent neural network based approach for internet of things malware threat hunting," *Future Generation Computer Systems*, vol. 85, pp. 88–96, 2018.

[24] Q.-L. Dong, S. N. He, and S. He, "Self-adaptive projection algorithms for solving the split equality problems," *Fixed Point Theory*, vol. 18, no. 1, pp. 191–202, 2017.

[25] C. D. McDermott, F. Majdani, and A. V. Petrovski, "Botnet detection in the internet of things using deep learning approaches," in *Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, Rio de Janeiro, Brazil, 8–13 July 2018.

[26] O. Brun, Y. Yin, and E. Gelenbe, "Deep learning with dense random neural network for detecting attacks against IoT-connected home environments," *Procedia Computer Science*, vol. 134, pp. 458–463, 2018.

[27] Y. Meidan, M. Bohadana, A. Shabtai et al., "Detection of Unauthorized IoT Devices Using Machine Learning Techniques," 2017, https://arxiv.org/abs/1709.04647.

[28] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning DDoS detection for consumer internet of things devices," in *Proceedings of the 2018 IEEE Security and Privacy Workshops*, pp. 29–35, SPW, San Francisco, CA, USA, 24 May 2018.

[29] E. Hodo, X. Bellekens, A. Hamilton et al., "Threat analysis of IoT networks using artificial neural network intrusion detection system," *Computers and Communications (ISNCC)*, in *Proceedings of the 2016 International Symposium on Networks*, pp. 1–6, Yasmine Hammamet, Tunisia, 11–13 May 2016.

[30] Y. Meidan, M. Bohadana, Y. Mathov et al., "N-BaIoT-Network-Based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.

[31] X. Li, P. Yi, W. Wei, Y. Jiang, Tian, and L. Lnnls-Kh, "A feature selection method for network intrusion detection," *Secur. Commun. Netw.*vol. 2021, Article ID 8830431, 22 pages, 2021.

[32] B. Selvakumar and K. Muneeswaran, "Firefly algorithm based feature selection for network intrusion detection," *Computers & Security*, vol. 81, pp. 148–155, 2019.

[33] S. Yilmaz and S. Sen, "Early detection of botnet Activities using Grammatical Evolution," in *Applications of Evolutionary Computation*, pp. 395–404, Springer International Publishing, Berlin/Heidelberg, Germany, 2019.

[34] M. Mazini, B. Shirazi, and I. Mahdavi, "Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms," *Journal of King Saud University - Computer and Information Sciences*, vol. 31, no. 4, pp. 541–553, 2019.

[35] A. Al Shorman, H. Faris, and I. Aljarah, "Unsupervised intelligent system based on one class support vector machine and Grey Wolf optimization for IoT botnet detection," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 7, pp. 2809–2825, 2020.

[36] K.-C. Lin, S.-Y. Chen, and J. C. Hung, "Botnet detection using support vector machines with artificial fish Swarm algorithm," *Journal of Applied Mathematics*, vol. 2014, pp. 1–9, 2014.

[37] Y. Yu, J. Long, F. Liu, and Z. Cai, "Machine learning combining with visualization for intrusion detection: a survey," in *Proceedings of the International Conference on Modeling Decisions for Artificial Intelligence*, pp. 239–249, Springer, Cham, Sant Julià de Lòria, Andorra, 19–21 September 2016.

[38] K. Shinan, K. Alsubhi, A. Alzahrani, and M. U. Ashraf, "Machine learning-based botnet detection in software-defined network: a systematic review," *Symmetry*, vol. 13, no. 5, p. 866, 2021.

[39] M. Alauthman, N. Aslam, M. Al-kasassbeh, S. Khan, A. Al-Qerem, and K.-K. Raymond Choo, "An efficient reinforcement learning-based Botnet detection approach," *Journal of Network and Computer Applications*, vol. 150, Article ID 102479, 2020.

[40] Y. Meidan, M. Bohadana, Y. Mathov et al., "Network-based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Computing, Special Issue - Securing the IoT*, vol. 17, no. 3, pp. 12–22, 2018.

[41] W. Jo, S. Kim, C. Lee, and T. Shon, "Packet Preprocessing in CNN-based network intrusion detection system," *Electronics*, vol. 9, no. 7, p. 1151, 2020.

[42] R. Yao, N. Wang, Z. Liu, P. Chen, and X. Sheng, "Intrusion detection system in the advanced Metering infrastructure: a cross-layer feature-Fusion CNN-LSTM-Based approach," *Sensors*, vol. 21, no. 2, p. 626, 2021.

[43] M. Ibrahim Ahmed Al-mashhadani, M. Hmoud Al-Adhaileh, A. M. Bamhdi, M. Y. Alzahrani, F. Waselallah Alsaade, and H. Alkahtani, "Human-animal affective robot touch classification using deep neural network," *Computer Systems Science and Engineering*, vol. 38, no. 1, pp. 25–37, 2021.

[44] Z. Ahmad, A. S. Khan, C. W. Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: a systematic study of machine learning and deep learning approaches," *Trans. Emerg. Telecommun. Technol.*vol. 32, no. 1, 2020.

[45] Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto, and K. Sakurai., "Machine learning-based IoT-botnet attack detection with Sequential architecture," *Sensors*, vol. 20, no. 16, p. 4372, 2020.