International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Botnet Attack Detection in IoT Network Using CNN-LSTM

Rituparna Borah[1], Satyajit Sarmah[1*], Chandan Kalita[1], Mirzanur Rahman[1], Vaskar Deka[1]

Submitted:13/03/2024 **Revised**: 28/04/2024 **Accepted**: 05/05/2024

Abstract: The fast rise of the IoT (Internet of Things) has led to a growth in cyber assaults, especially on IoT devices. Malicious assaults in the IoT ecosystem must be detected in order to decrease safety issues. Botnet attacks, like Bashlite and Mirai, pose a significant threat to IoT devices. Static IoT devices, with the shortage of adequate memory and resources for computation, are particularly more vulnerable. To overcome this problem, we employed the CNN-LSTM model to identify various botnet assaults on IoT devices. In this work, we utilized an actual N-BaIoT dataset with mostly benign and malignant behaviours. According to the outcomes, the CNN-LSTM model works remarkably well. It notices botnet assaults on doorbell IoT devices (such as Danminin and Ennio) with 90.88% and 88.77% accuracy, respectively. The system also achieves 88.53% accuracy while identifying botnet attacks on thermostat device. By observing the results, we can mention that the Convolution Neural Network-Long Short Term Memory (CNN-LSTM) algorithm successfully detects botnet attacks on several IoT devices with high accuracy, providing an efficacious method for enhancing IoT security

Keywords: Convolutional Neural Network, Internet of Things, Network Security, Cyber Security, Botnet attack, NBaIot dataset

1. Introduction

The term IoT coined by Kevin Ashton[1] is a buzzword in the Information Technology today. IoT is an integrated system of interrelated devices where devices are connected to the internet so that they can communicate and share data from one to the other. The main aim is to collect data from its neighboring area using sensors and actuators and then analyse and process the data received from different kinds of IoT devices [2]. The whole of IoT devices deployed in 2021 was 35.82 billion, and it is expected to rise to 75.44 billion by 2025 [3]. The amount of active IoT devices is estimated to exceed 25.4 billion by 2030, up from only 10 billion in 2021. Every 127th new gadget is linked to the internet in 2019 [4]. The swift advancement of IoT infrastructure comes with the trade-off of facing numerous attacks and intensified security issues. According to Symantec's report, IoT devices are under attack every two minutes [5]. There are many types of malwares and many cyber-attacks use a combination of several types to achieve their goals.

[1]*Research scholar, Department of Information Technology Gauhati University, Jalukbari, Guwahati, Assam 781014, India Orcid id: 0009-0009-6453-9367*
[1*]*Assistant Professor, Department of Information Technology Gauhati University, Jalukbari, Guwahati, Assam 781014, India Orcid id: 0000-0002-4032-5278*
[1]*Assistant Professor, Department of Information Technology Gauhati University, Jalukbari, Guwahati, Assam 781014, India Orcid id: 0000-0002-5842-221X*
[1]*Assistant Professor, Department of Information Technology Gauhati University, Jalukbari, Guwahati, Assam 781014, India Orcid id: 0000-0003-0356-1279*
[1]*Assistant Professor, Department of Information Technology Gauhati University, Jalukbari, Guwahati, Assam 781014, India Orcid id: 0000-0001-6361-442X*
**Corresponding Author mail id: satyajitnov2@gmail.com*

Botnet represents one example of such malware. Owari, Mirai and Bashlite are the very popular botnet attacks now days [6], [7], [8].

A botnet is a group of computers which are associated and work under the instruction of a master called as the Bot-Master or Bot-Herder in order to accomplish something [9], [10], [11]. Most botnets that we found till now required a common centralized architecture. That means, bots in the botnet connect directly to some special device called server, which is termed as control servers. These control servers accept instructions from the botmaster and forward these to the other bots present in the network [9]. Previous technologies for securing IoT devices from botnet attacks have significant deficiencies and lack effective mechanism. One potential remedy for combating botnet attacks is the implementation of an intrusion detection system (IDS).

A deep autoencoder model proposed in [12] effectively detects botnet attacks in IoT environments, outperforming SVM and Decision Tree algorithms on the N-BaIoT dataset, promising improved security for IoT systems.In [13] the authors applied traditional ML techniques to extract significant features and improved the system a bit. Specifically, the researchers used a method called Linear Nearest Neighbor lasso step (LNNLS-KH). This LNNLS-KH method was employed to update the positions of a krill herd, aiming to achieve optimal global solution. In [14] the authors created an LSTM classifier for IoT applications based on Advanced RISC Machines (ARM). The LSTM classifier was evaluated using 100 samples of malware data that weren't employed in the model training. The proposed approach performed well in identifying and categorizing malware in ARM-based IoT systems, with an average accuracy of 97%.The study in [15] proposes deep learning techniques, specifically Bidirectional Long Short Term

Memory (BiLSTM), for detecting botnet attacks, particularly focusing on the Mirai botnet, achieving an accuracy of 99.98% with the BLSTM model, indicating the effectiveness of DL methods in securing computer systems against such threats. In [16] the researchers used ML approaches to identify several botnet assaults. They trained and evaluated four distinct ML systems depend on four classifiers: Nave Bayes, K-Nearest Neighbors, Support Vector Machines (SVM), and Decision Trees utilizing two datasets, Bot-IoT and University of New South Wales (UNSW). The study's main findings indicate that the Decision Trees model performed exceptionally well in detecting botnet attacks. In [17] the authors demonstrated the process of collecting data from smart cities, sensors, and human inputs. Their recommendation involved utilizing anomaly detection model-based machine learning techniques with annual power data, loops, and land sensor data. They employed long short-term memory-neural network (LSTM-NN) and MLP models for this purpose, with LSTM-G-NB achieving the highest accuracy. In [18] the authors employed packet-captured data via port mirroring within a network that encompassed various IoT devices. These devices, comprising nine types such as a baby monitor, motion sensor, refrigerator, security camera, smoke detector, socket, thermostat, TV, and a watch, were investigated. The study introduced a random forest learning approach based on an unauthorized IoT device classifier model. The model exhibited an average accuracy of 94%.

From the above we can conlude that

a) We can consider more efficient techniques for feature extraction and selection in traditional machine learning techniques to enhance the performance of botnet attack detection systems.

b) Address scalability and efficiency challenges in LSTM-based classifiers and deep learning methods to enable their deployment in large-scale IoT networks without compromising performance.

c) Explore methods for integrating LSTM-based classifiers and deep learning models with existing IoT security frameworks and infrastructures, facilitating seamless deployment and management.

d) Develop defenses against adversarial attacks targeting LSTM-based classifiers and deep learning models, ensuring their resilience to manipulation attempts aimed at evading detection.

e) Consider techniques for adapting LSTM-based classifiers and deep learning models to handle concept drift and evolving attack strategies in IoT environments, ensuring their effectiveness over time.

Botnet attacks, exemplified by BASHLITE and Mirai, pose a significant threat to IoT devices.Static IoT devices, characterized by limited memory and computational resources, are particularly vulnerable to these attacks. The objective is to develop a method for detecting botnet assaults on IoT devices effectively. The factors include the proliferation of IoT devices with inadequate security measures and sophisticated techniques employed by cybercriminals to exploit vulnerabilities in IoT devices. Existing solutions for IoT security often lack effectiveness; especially against evolving botnet attacks.Traditional intrusion detection systems may not be suitable for the dynamic nature of IoT environments. In this paper we mainly focus on detectition of botnet attack on IoT devices using CNN- LSTM Model. CNN-LSTM model could identify botnet attack with better accuracy. It has a chance to improve IoT device and network safety against botnet attacks. Implementing effective detection methods is essential for enhancing IoT security and mitigating the risks associated with cyber assaults.

2. Method

In this section, we outline the framework we developed for constructing an IoT botnet detection model, encompassing the entire workflow from dataset definition to botnet detection.

2.1. N-BaIoT dataset

The N-BaIoT datset was gathered from an ML repository and focuses on network data in IoT contexts. It has 155 characteristics that were obtained through switch port mirroring. The collection contains genuine network traffic from a variety of sources, including nine commercial IoT gadgets. The gadgets type and and the gadget name used in NBaIot dataset is shown in Table 1. In this case, 23 primary characteristics were retrieved at various times, including 100ms, 500ms, 10s, 10 min, and 1min. Table 2 shows the primary characteristics of the dataset. The collection includes nine commercial gadgets that were employed to capture network traffic, such as botnet assaults. The two main types of attacks present in the dataset are named Mirai and Bashlite.

Figure 1 illustrates the laboratory setup employed to gather botnet attcaks from IoT devices. Multiple access point objects were used to link these IoT devices to Wi-Fi. Port mirroring was enabled on the switch devices to collect and evaluate actual network activity. The datasets containing the network traffic were capturted and recorded using Wireshark software, a widely used network protocol analyser.Table 3 summarizes the attack patterns seen in the dataset, with an emphasis on two prevalent botnet assaults, BASHLITE and Mirai. BASHLITE assaults are DDoS attacks that were developed in C programming to infiltrate Linux computers. These attacks are very common in IoT gadgets such as cameras. Mirai botnet assaults, on the other hand, that were found in 2016, use malware built to operate on ARC processors and target large-scale IoT networks.

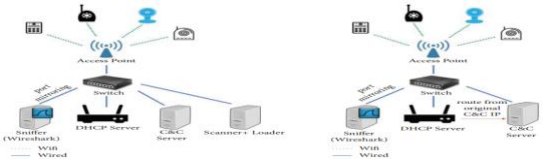

Fig 1. Different Botnet Attack on IoT devices in lab environment [12]

Table 1. The gadget type and the gadget model name used in NBaIot dataset

Gadget type	Gadget model name
Dorbell	Daminin Emnio
Thermostat	Ecobee
Baby Monitor	Philips B120 N/10
Security Camera	Provision PT-737E Provision PT-838 Simple Home XCS7-1002-WHT Simple Home XCS7-1003-WHT
Webcam	Samsung SNH101N

2.2 Proposed Methology

Figure 2 depicts the system architecture of the developing system. Our framework encompasses three main components: a botnet dataset, botnet training models, and botnet detection models. The botnet dataset comprises four subdatasets extracted from N-BaIoT. We specifically selected devices that encompass all ten attack samples detailed in Table 3 of the N-BaIoT dataset. These devices include a doorbell (Ennio), a baby monitor (Philips B120N/10) and thermostat.

We used the CNN-LSTM as the botnet training model. We have used multiclass classification botnet detection model.

2.2.1. CNN-LSTM

CNNs, specialized for image tasks, employ convolution layers where neurons are organized in 3D (width, height, depth), linked to a small receptive field of the preceding layer, incorporating convolution, activation, pooling, and fully connected layers as core components [12], [19]. The Convolution layer in CNN extracts features from input data by sliding multiple filters across it, multiplying filter elements with local receptive field elements to create weighted summations representing specific patterns or features.

The convolutional operation in a neural network incorporates Stride, filter size, and zero padding to control the filter's movement, dimensions, and maintain spatial information respectively.It enables convolution operations to preserve spatial dimensions of the input, particularly at the edges and aids in the avoidance of information loss owing to boundary factors [20], [21].

Table 2. Primary Characteristics of NBaIot dataset

Aggregated by	Value	Statistics	Total no. of features
Source IP	Packet size (only outbound) Packet Count	Mean, Variance	3
Source MAC_IP	Packet size (only outbound) Packet Count	Integer Mean, Variance Integer	3
Channel	Packet size (only outbound) Packet Count Amount of time between packet arrivals Packet size (Both inbound and outbound)	Mean, Variance integer Mean, Variance, Integer Magnitude, radius, covariance, correlation , coefficient	10
Socket	Packet size (only outbound) Packet count Packet size (both inbound and outbound)	Mean, Variance Integer	7
	Total		23

Table 3. Different BoTNET Attack

Major attacks	Subattacks	Description
Bashlite	Junk	By sending spam data
	TCP Flood	Sends flood of request

	UDP Flood	Sends flood of request
	Scan	Scan the network for victim devices
	COMBO	Opens connection IP address and network port by sending spam data
Mirai	ACK	Sends flood of acknowledgement
	SYN	Sends synchronize-packet-flood
	Plain UDP	UDP flood by optimizing seeding packet per second scans the network for victim devices
	Scan	

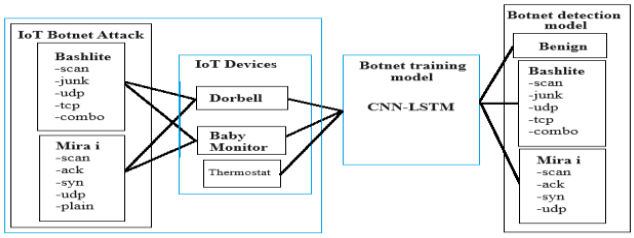

Fig 2. System Architecture

Rectified linear activation function (ReLU) is a non-linear activation function utilized to perform element-wise activation on feature maps produced by convolutional layers. It operates in this manner: it returns 0 for negative input values and the same value (x) for positive input values. Its range spans from 0 to infinity, which means it only modifies negative values, leaving positive values unchanged. This function is critical in bringing non-linearity to the network, allowing it to record complicated correlations and improve CNN's learning ability.

$$ReLU = \begin{cases} 0, if\ X < 0, \\ x, if\ x \geq 0. \end{cases} \quad (1)$$

The pooling layer in neural networks, notably max pooling, reduces input dimensions by selecting the maximum value within a pooling filter, enhancing efficiency and effectiveness.It contributes to downsizing the input size by 75%, leading to significant outcomes [22], [23]. This process aids in decreasing computational complexity while retaining essential features, making max pooling a prevalent and promising technique in neural network [24] .

The fully connected layer, or dense layer, in CNNs connects every node to those in the preceding and next layers, collecting and integrating learned characteristics from previous layers to generate final predictions, crucial for tasks such as image identification and object recognition.

The RNN is a sophisticated DL model that is utilized in several real-world applications. The LSTM model is a type of RNN that is especially built to analyze input in a sequential manner with feedback links.

The sructural design of the LSTM design is shown in Figure 3, with "x" representing input data, and "y" representing classification output. The LSTM method contains mainly three gates. Those are input gate, forget gate, and output gate.

The input gate is in charge of storing pertinent training data in long-term memory. It begins with current input information and initializes short-term memory with last time step. The input gate employs filters to retrieve essential details while discarding unhelpful data. The required data is sent via the sigmoid function that provides a value of 1 to relevant data and a value of 0 to unimportant data. The input layer's output is stored in long-term memory.

The forget gate, that is also a substantial part of every LSTM network. It decides what data to maintain or discard by multiplying the forget vector values by the current input gate. The forget gate output is then transferred to the next cell, allowing the long-term memory to be refreshed.

In this study, we have combined the CNN and LSTM models to detect botnet attacks across different IoT device types. In our study, the hybrid CNN-LSTM model's generic structure is depicted in Figure 4.The primary components of the proposed system for detecting botnet attacks from IoT devices are outlined in Table 4. Specifically, we set the size of the convolution kernel to 5 and the number of epochs to 20. The ReLU activation function was employed throughout the system. Figure 5 illustrates a snapshot of the CNN-LSTM model.

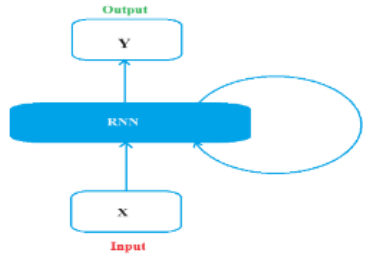

Fig 3. Structure of LSTM [6]

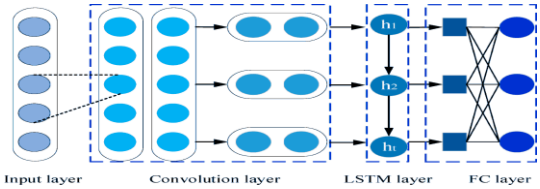

Fig 4. Structure of CNN-LSTM layer

Table 4. Parameters of CNN-LSTM model

Parameters	Value
Convolution Filters	100
Kernel size of filter	5
Fully connected layer	256
Activation function	ReLu
Classification function	Softmax
Optimizer	VGG16
Epochs	20

```
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, 115, 64)           384
conv1d_1 (Conv1D)            (None, 115, 32)           10272
lstm (LSTM)                  (None, 115, 32)           8320
lstm_1 (LSTM)                (None, 115, 16)           3136
flatten (Flatten)            (None, 1840)              0
dense_5 (Dense)              (None, 128)               235648
dense_6 (Dense)              (None, 64)                8256
dense_7 (Dense)              (None, 11)                715
=================================================================
Total params: 266,731
Trainable params: 266,731
Non-trainable params: 0
_________________________________________________________________
```

Fig 5. Snapshot of CNN-LSTM Model

3. Results and discussion

In this section, we have included the summaery of the results obtained.

3.1. Environment Setup

The proposed system was developed using the software and hardware environment as described in Table 5.

Table 5. Software and hardware environment

Hardware/Software	Requirement
Operating System	Windows 10
CPU	2.40 GHz
Memory	8GB
Development environment	Jupyter Python 3.9.12
Matplotlib	Version 3.5.1
NumPy	Version 1.21.5
Pandas	Version 1.4.2
Scikit-learn	Version 1.0.2
Keras	Version 2.9.0
Tensorflow	Version 2.9.11

3.2 Evaluation Metric

The metric used to test the system for detection of botnet attack are Accuracy, Precision, Recall and F1-Score. All of them are defined using the following equation:

$$Accuracy = (TP + TN)/(FP + FN + TP + TN)$$

$$Precision = (TP + FP)/TP$$

$$Recall = TP/(TP + FN)$$

$$F1 - Score = 2 * (Precision * Recall)/(Precision + Recall)$$

Where TP denotes True Positive, TN denotes True Negative, FP denotes False Positive and FN denotes False Negative.

3.3 Results

The experiment involved conducting three trials on different IoT environments to assess a proposed system's effectiveness. CNN-LSTM was used and implemented to identify botnets within the IoT networks. The goal was likely to evaluate the system's performance in detecting and countering botnet attacks in the context of IoT environments. We utilize Scikit-learn's dataset split function to randomly divide the samples from the N-BaIoT dataset into training and testing sets, with a ratio of 70% for training and 30% for testing. The expreminent details of the three devices is presented in the next section

3.3.1. Experiment 1 for Dorbell Devices

Utilizing network data obtained from doorbell devices, notably Danminin and Ennio, the combined CNN-LSTM network is used to identify various anomalies. Tables 6(a) and 6(b) shows the outcomes of hybrid model. Weighted averages of 90.88% accuracy, 90% recall, and 87% F1-score define the proposed method's performance in spotting attack

anomalies from Danminin doorbell. Similarly, weighted averages of 88.87% accuracy, 88% recall, and 85% F1-score reflect the proposed system's efficacy in identifying intrusions from the Ennio doorbell.

Table 6(a). Detection of different attacks from doorbell (Danminin) device through CNN-LSTM model

Dorbell Danminin			
Attacks	Precision	Recall	F1-Score
Benign	100	100	100
mirai_udp	100	95	97
COMBO	100	95	97
Junk	100	100	100
Scan	100	0	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	95	90	87
Loss		0.12	

Table 6(b). Detection of different attacks from doorbell (Ennio) device through CNN-LSTM model

Dorbell Ennio			
Attacks	Precision	Recall	F1-Score
Benign	100	100	100
mirai_udp	98	93	95
COMBO	94	99	96
Junk	100	100	100
Scan	75	0.00	0.00
TCP	53	100	69
UDP	90	97	94
ACK	100	100	100
Mirai-Scan	100	100	100
Mirai-SYN	95	83	89
Mirai_udpplain	99	97	98
Accuracy	88.61		
Weighted average	93	89	85
Loss	0.19		

The graphs in Figures 6 and 7 depict the training model's confusion metrics, which capture its ability to distinguish patterns of novel botnet assaults emanating from both Danminin and Ennio devices. Figure 8 displays how effectively CNN-LSTM algorithm recognizes incursions from both Danminin and Ennio devices.

3.3.2 Experiment 2 for Thermostat Device

CNN-LSTM is used to identify intrusions based on data obtained from a thermostat device. Table 7 shows the model's performance in identifying botnet assaults. The assessment measures' weighted average findings were as follows: accuracy - 94%, recall - 89%, and F1-score - 85%. These metrics demonstrate how well the CNN-LSTM approach identified and classified botnet assaults in the provided dataset.

Figure 9 depicts the CNN-LSTM model's confusion metrics when botnet assaults were identified utilizing network data from thermostat devices. According to the data, the system efficiently recognized a substantial number of botnet assaults, demonstrating its excellent identification capacity.

Figure 10 illustrates the CNN-LSTM model's performance analysis, which demonstrates its capacity to identify various botnet assaults inside an IoT context using data from thermostat devices. The accuracy of the CNN-LSTM method is displayed in Figure 10 (a), with an improvement from 80% to 88.53% when the model is trained across 20

epochs. Figure 10 (b) depicts the equivalent training loss trend, which shows a decrease from an initial value of 20.0 to a minimal value of 0.16. These statistics highlight the method's improved precision and efficacy in detecting botnet assaults during the training phase.

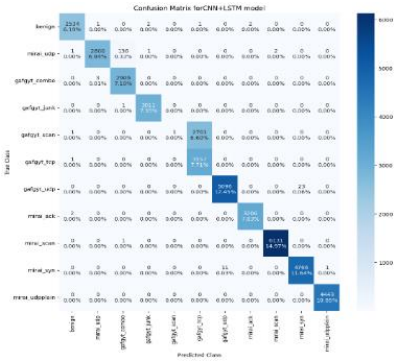

Fig 6. Confusion metrics of CNN-LSTM for (a) Danmini

3.3.3 Experiment 3 for Baby Monitor Device

We have used CNN-LSTM approach for detection of intrusions from IoT devices, specifically baby monitors. The outcomes of the experiment are outlined in Table 8. To efficiently address botnet assaults, the proposed approach displayed good performance, attaining high accuracy in finding unfamiliar patterns within datasets. The system's robustness was demonstrated by weighted averages of important assessment criteria of 93% for accuracy, 92% for recall, and 89% for the F1-score. The outcomes are further illustrated through confusion metrics in Figure 11, indicating that the model successfully trained to recognize all botnet assaults.

3.4 Discussion

In this work, we utilized the CNN-LSTM model to identify botnets attack. The system was evaluated using NBaIot dataset. From the results of the first experiment for doorbell devices Danminin and Ennio as shown in table 6(a) and 6(b) it is found that CNN-LSTM model showed an accuiracy of 90.88 for Danminin devices and 88.61 for Ennio devices. The model achieved 100% performance in most of the attacks except for the Scan attack the performance was low in case of Danminin. In case of Ennio also the performance was high for most of the attacks but it was low for Scan and TCP attack. The oveall performance was good for both the device. From the results of the second experiment for Thermostat devices as shown in Table 7 it is found that CNN-LSTM model showed an accuracy of 88.53. The overall performance was good except for TCP attack which showed a low performance i.e. 52% precision and 69% f1-score.

From the results of the third experiment for Baby Monitor devices as shown in Table 8 it is found that the device showed an accuracy of 91.5. The overall performance was good except for Scan attack it was 67% .and for TCP attack 54% precision and 70% F1-Score.

The development of a security system to identify intrusions within the IoT environment has been crucial in safeguarding the IoT network. The CNN-LSTM deep learning algorithm was employed for detecting botnet attacks. Table 9 presents a summary of the CNN-LSTM model's results against F1-Score compared to existing systems. In [25] the authors have used the same dataset and used the deep learning algorithms CNN, RNN and LSTM for multiclass classification.

In this research, we implemented the proposed system using datasets extracted from nine IoT devices. We evaluated the F1-Score and compared them with those of existing systems. Our findings indicate that the proposed system demonstrates superior performance in all aspects.

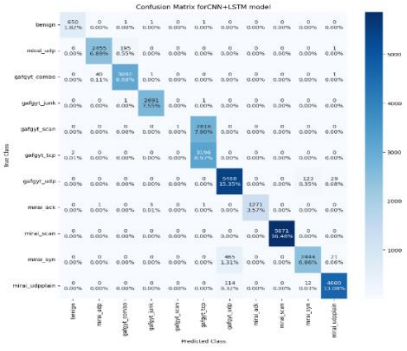

Fig 7. Confusion metrics of CNN-LSTM for (b) Ennio

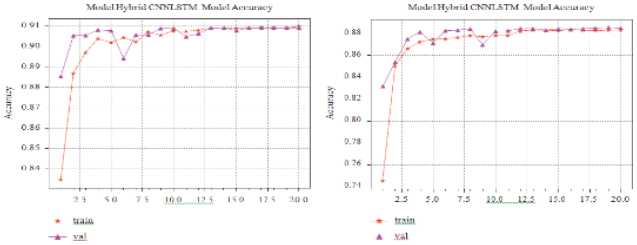

Fig 8. Detection of Botnet attack from doorbell devices using CNN LSTM (a) Danminin (b) Ennio

Table 7. Detection of different attacks from doorbell thermostat IoT device through CNN-LSTM model

Attacks	Precision	Recall	F1-Score
Benign	99	100	100
mirai_udp	99	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		

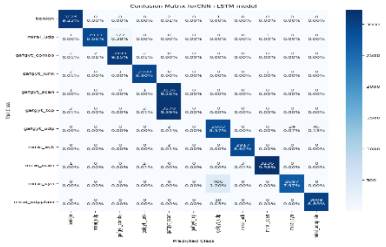

Fig 9. Confusion metrics of CNN-LSTM for Thermostat Device

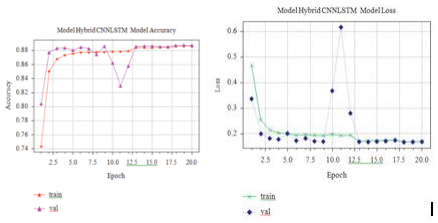

Fig 10. Detection of Botnet attack from thermostat Iot devices using CNN LSTM (a) Training accuracy (b) Training Loss.

Table 8. Detection of attacks from Baby Monitor IoT device using CNN LSTM MODEL

Attacks	Precision	Recall	F1-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	

4. Conclusion

We created a framework to identify IoT botnet attacks using CNN-LSTM, and subsequently utilized it to uncover instances of botnet attacks directed at a range of IoT devices. Our framework comprises a botnet dataset, a training model, and a detection model dedicated to botnet activity[26].

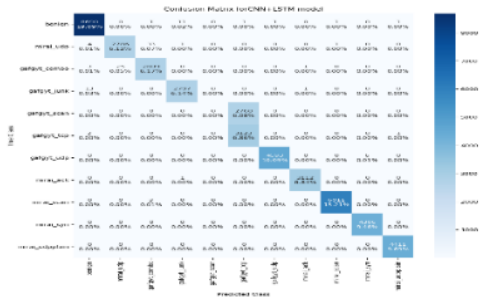

Fig 11. Confusion metrics of CNN-LSTM for Baby Monitor devices

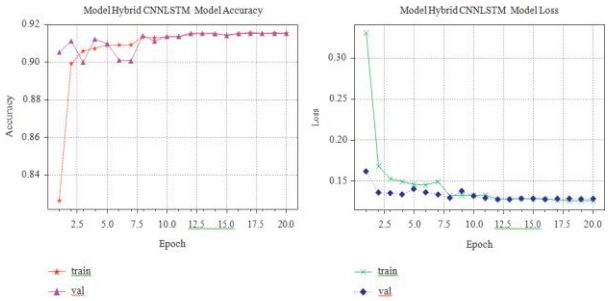

Fig 12. Detection of botnet attacks from baby monitor device using CNN LSTM Model

Table 9. Comparision of the F1 score of the proposed model with the existing one[25]

Model	Dorbell	Baby Monitor	Thermostat
CNN	0.91	0.91	-
RNN	0.41	0.44	-
LSTM	0.62	0.54	-
CNN-LSTM(Proposed)	0.86	0.89	0.85

We employed the N-BaIoT dataset in this study that was obtained via various nine commercial IoT gadgets, and mostly doorbell Danmini, Ennio, and thermostat devices were chosen for implementation. These IoT devices were primarily targeted by two kinds of attacks: bashlite and Mirai. Junk, Scan, combo, TCP flood, and UDP flood were all part of the bashlite assault. On the other hand, the Mirai attack was classified into subattacks including Syn, Ack, Scan, Plain UDP and UDP flood attacks. This work mainly focuses on leveraging this dataset and attack scenarios to identify DDoS attacks in their early stages, thereby contributing to the enhancement of network security measures using CNN-LSTM model. The model has demonstrated notable success. This success highlights the model's strong capability in accurately identifying and classifying such attacks. We can conclude that the CNN LSTM model can perform well in terms botnet attack detection in IoT environment. In future we will try to improve the accuracy to 100 percent. Different dataset also can be used for the same model and a comparison also can be made to evaluate the performance.

References

[1] Radouan Ait Mouha, Internet of Things," *Journal of Data Analysis and Information Processing*, Vol.9 No.2, May 2021.

[2] Ashton, Kevin, "That 'internet of things' thing," *RFID journal* vol. 22, no. 7, pp. 97-114, 2009.

[3] Steward, Jack, "The ultimate list of internet of things statistics for 2022," URL https://findstack.com/internet-of-things-statistics (2022).'.

[4] Jovanovic, Bojan, "Internet of things statistics for 2022-taking things apart," Data Prot (2022).'.

[5] Pepper, Robert, "Cisco visual networking index (VNI) global mobile data traffic forecast update." *In Mobile World Congress*, 2013.

[6] Alkahtani, Hasan, and Theyazn HH Aldhyani, "Botnet attack detection by using CNN-LSTM model for Internet of Things applications," *Security and Communication Networks*, pp. 1-23, 2021,

[7] Shazly, Khadija, Dina A. Salem, Nacereddine Hammami, and Ahmed IB ElSeddawy, "A Review on Distributed Denial of Service Detection in Software Defined Network."'.

[8] Kalidindi, Archana, and Mahesh Babu Arrama. "A tab transformer based model for detecting botnet-attacks on internet of things using deep learning," *Journal of Theoretical and Applied Information Technology*, vol. 101, no. 13, 2023.

[9] Kaur, Navdeep, and Maninder Singh, "Botnet and botnet detection techniques in cyber realm," In 2016 international conference on inventive computation technologies (ICICT), vol. 3, pp. 1-7. IEEE, 2016.

[10] B. Saha and A, Gairola, "Botnet: An overview," CERT-In White PaperCIWP-2005-05, 2005.'.

[11] M. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "A multifaceted approach to understanding the botnet phenomenon," in Proc. 6th ACM SIGCOMM Conference on Internet Measurement (IMC'06), 2006, pp.41-52',

[12] Meidan, Yair, Michael Bohadana, Yael Mathov, Yisroel Mirsky, Asaf Shabtai, Dominik Breitenbacher, and Yuval Elovici, "N-baiot—network-based detection of iot botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol.17, no. 3, pp. 12-22, 2018.

[13] Li, Xin, Peng Yi, Wei Wei, Yiming Jiang, and Le Tian, "LNNLS-KH: a feature selection method for network intrusion detection," *Security and Communication Networks*, pp. 1-22, 2021.

[14] HaddadPajouh, Hamed, Ali Dehghantanha, Raouf Khayami, and Kim-Kwang Raymond 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.

[15] McDermott, Christopher D., Farzan Majdani, and Andrei V. Petrovski, "Botnet detection in the internet of things using deep learning approaches," In 2018 international joint conference on neural networks (IJCNN), pp. 1-8. IEEE, 2018.

[16] Alshamkhany, Mustafa, Wisam Alshamkhany, Mohamed Mansour, Mueez Khan, Salam Dhou, and Fadi Aloul, "Botnet attack detection using machine learning," In 2020 14th International Conference on Innovations in Information Technology (IIT), pp. 203-208. IEEE, 2020.

[17] X. Xie, D. Wu, S. Liu, and R. Li, "IoT data analytics using deep learning," in IEEE Access, vol. 6, pp. 16793-16802, 2018, doi: 10.1109/ACCESS.2018.2815839.'.

[18] Y. Meidan, M. Bohadana, A. Shabtai, M. Ochoa, N. O. Tippenhauer, J. D. Guarnizo, and Y. Elovici, "Detection of unauthorized IoT devices using machine learning techniques," arXiv, 2017, arXiv:1709.04647 [cs.CR].'.

[19] Jo, Wooyeon, Sungjin Kim, Changhoon Lee, and Taeshik Shon, "Packet preprocessing in CNN-based network intrusion detection system," *Electronics*, vol. 9, no. 7,pp. 1151, 2020.

[20] Guo, Yanming, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S. Lew, "Deep learning for visual understanding: A review." *Neurocomputing*, vol. 187, pp. 27-48, 2016.

[21] Wu, Jianxin, "Introduction to convolutional neural networks," National Key Lab for Novel Software Technology. Nanjing University. China 5, no. 23,pp. 495, 2017.

[22] Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1-9. 2015.,

[23] Han, Kun, Dong Yu, and Ivan Tashev, "Speech emotion recognition using deep neural network and extreme learning machine," In Interspeech 2014. 2014.'.

[24] Ahmad, Zeeshan, Adnan Shahid Khan, Cheah Wai Shiang, Johari Abdullah, and Farhan Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1 (2021): e4150.'.

[25] Kim, Jiyeon, Minsun Shim, Seungah Hong, Yulim Shin, and Eunjung Choi, "Intelligent detection of iot botnets using machine learning and deep learning," *Applied Sciences*, vol. 10, no. 19 pp.7009, 2020.

[26] Versaci M, Angiulli G, Crucitti P, De Carlo D, Laganà F, Pellicanò D, Palumbo A, "A Fuzzy Similarity Based Approach to Classify Numerically Simulated and Experimentally Detected Carbon Fiber-Reinforced Polymer Plate Defects," *Sensors*, 2022, vol. 22, pp. 4232. https://doi.org/10.3390/s22114232.