Hybrid Deep-Learning Model to Detect Botnet Attacks over Internet of Things Environments

Mohammed Y. Alzahrani (✉ msawileh@bu.edu.sa)

Al baha University <https://orcid.org/0000-0002-9726-6088>

Alwi M Bamhdi

Omm-Al-Qura University

Research Article

Keywords: Internet of things, Botnet attacks, Artificial intelligence, cybersecurity

Posted Date: November 23rd, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-1049378/v1>

License: This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Version of Record: A version of this preprint was published at Soft Computing on February 14th, 2022.
See the published version at <https://doi.org/10.1007/s00500-022-06750-4>.

Hybrid Deep-Learning Model to Detect Botnet Attacks over Internet of Things Environments

Mohammed Y. Alzahrani¹ and Alwi M Bamhdi²

¹Department of Computer Sciences and Information Technology, Al Baha University, Saudi Arabia,
msawileh@bu.edu.sa

²Alwi M Bamhdi, the Department of Computer Sciences, College of computing (Al Qunfudhah),
Umm Al-Qura University, Saudi Arabia. ambamhdi@uqu.edu.sa

In recent years, the use of the internet of things (IoT) has increased dramatically, and cybersecurity concerns have grown in tandem. Cybersecurity has become a major challenge for institutions and companies of all sizes, with the spread of threats growing in number and developing at a rapid pace. Artificial intelligence (AI) in cybersecurity can to a large extent help face the challenge, since it provides a powerful framework and coordinates that allow organisations to stay one step ahead of sophisticated cyber threats. AI provides real-time feedback, helping rollover daily alerts to be investigated and analysed, effective decisions to be made and enabling quick responses. AI-based capabilities make attack detection, security and mitigation more accurate for intelligence gathering and analysis, and they enable proactive protective countermeasures to be taken to overwhelm attacks. In this study, we propose a robust system specifically to help detect botnet attacks of IoT devices. This was done by innovatively combining the model of a convolutional neural network with a long short-term memory algorithm mechanism to detect two common and serious IoT attacks (BASHLITE and Mirai) on four types of security camera. The data sets, which contained normal malicious network packets, were collected from real-time lab-connected camera devices in IoT environments. The results of the experiment showed that the proposed system achieved optimal performance, according to evaluation metrics. The proposed system gave the following weighted average results for detecting the botnet on the Provision PT-737E camera: camera precision: 88%, recall: 87% and F1 score: 83%. The results of system for classifying botnet attacks and normal packets on the Provision PT-838 camera were 89% for recall, 85% for F1 score and 94%, precision. The intelligent security system using the advanced deep learning model was successful for detecting botnet attacks that infected camera devices connected to IoT applications.

Keywords: Internet of things, Botnet attacks, Artificial intelligence, cybersecurity,

1. Introduction

Due to an increase in cybercrime, researchers have been focusing on identifying intrusions in networks [1]. Previously, traditional computer networks and personal computers were the focus of cyberattacks, but now communication infrastructure encompasses the internet of things (IoT) [3], the internet of connected vehicles [5], the internet of medical things [4] and 5G [7], these have become the targets of many cyberattacks. Older strategies, such as firewalls and antivirus software, are unable to offer solutions to complex cyberattacks [10]. Machine learning and deep learning algorithms are now used to detect intrusion in networks [12–14]. Algorithms can also be developed to perform both in group strategies and in combination. Grouping strategies is a machine learning technique to solve

intrusion detection and prediction problems. At the present time, big data produced by computers and IoT devices is causing a threat to data traffic in networks and increases the malware that threatens the integrity of data; however, these issues cannot be dealt with by current strategies [16]. A bot in a network is a personal computer containing malware that allows attackers to hack it. A botnet is a spider computer network made up of many hosts, each running independent programs. It runs a bot on a large number of devices connected to the internet to form a botnet network operated by a malicious group [19]. Botnets pose a threat to network security because they are used in cybercrime techniques such as distributed denial of service (DDoS). Machine learning algorithms are used to track such attacks in IoT [20–22]. A botnet can be used to refuse services directed and distributed to any system on the internet so that it cannot properly serve its legitimate customers. Currently, DDoS attacks are performed from a botnet platform; despite their simplicity, they are very effective due to the bandwidth of the bots. An intruder can gain illegal access to user data through a large number of attacks. Networks are exposed to a wide variety of attacks, including probing, denial of service, user-to-root and port scanning. To execute these attacks, transport or protocols such as internet control message protocol, user datagram protocol (UDP), transmission control protocol (TCP) and file transfer protocol may be used. Network-based intrusion detection systems (NIDS) are among the best means to scan networks and identify attacks [23,24]. Many machine learning algorithms have been used to detect cyberattacks, but these fail the sanctity of big data traffic. Also, they lack the required optimisation. Researchers are now able to develop systems to detect and verify botnet data [25]. However, complex data traffic does not guarantee high accuracy, as the techniques for detecting botnet attacks are constantly changing. Therefore, there is a need for classification techniques based on neural development that can identify the number of layers and neurons when detecting attacks [26]. When training the network and learning weights, hyperparameters must be set. Signature-based and anomaly-based detection are two of the most important methods of intrusion detection. The former uses signature-based detection of a known attack pattern, while anomaly-based detection is used for both known and unknown attack patterns [31]. Traffic is identified by NIDS, which means extracting the most important features from traffic records in order to classify them as malicious or normal by machine learning algorithms [32]. Network- and host-based systems are two means to classify network traffic records. In the NIDS network, all log traffic is monitored for intrusions such as DoS [33]. When using vast databases for anomaly detection, high detection accuracy and low rate of alerts can be obtained. Training and testing phases are implemented to develop databases for anomaly-based systems; patterns are identified in the training phase and compared during the testing phase. Signature and heuristic methods cannot detect malware or provide an adequate level of detection against new and unknown variants, so machine learning algorithms are used to solve this problem. Machine learning and deep learning techniques detect attacks without requiring advanced security knowledge [34,35]. The efficiency of intrusion detection systems (IDSs) can be improved through nature-inspired, meta-heuristic data mining, reinforcement learning, grammar-based machine learning and artificial intelligence [37–39]. IDS performance can also be improved through the artificial bee colony (ABC) [40], grey wolf optimisation [42] and artificial fish swarm (AFS) algorithms [43]. Rajagopal et al. proposed a group model using a meta-classification method based on stacked generalisation and used two data sets, UGR'16 and UNSW NB-15, which were collected from real and emulated network traffic. The proposed system achieved accuracy of 97%, as well as 94% with emulated data sets [44]. Elmasry et al. proposed particle swarm optimisation (PSO) for selecting hyperparameters and their most important features in a single process. Classification was done by three algorithms, namely deep belief networks, deep neural networks and long short-term memory recurrent neural networks (LSTM-RNN) [45]. Tripathi et al. proposed an identification grasshopper optimisation method for distinguishing between normal and malicious traffic for the CIC-IDS 2017 and KDD Cup 99 data sets. The machine learning algorithms they used determined the type of attack [46]. Suhaimi et al. proposed a genetic algorithm, with an immune algorithm to improve attack detection accuracy, and they performed a number of simulations to check the performance of the method. The immune genetic algorithm performed well for predicting network penetration [47]. Zhou et al. proposed the M-AdaBoost-A algorithm for predicting network intrusion, and they used methods

such as POS to group several classifiers according to M-AdaBoost-A. Their method has achieved good performance in detecting intrusions in both wireless and traditional networks [48].

Wu et al. presented the SRDLM method, based on deep learning and semantic re-encoding, which took advantage of deep learning methods to enhance classification capabilities. It achieved 99% accuracy in detecting a web injection network attack [49].

Injadat et al. suggested a multi-stage machine learning-based NIDS method for assessing the UNSWNB 2015 and CICIDS 2017 data sets. The method worked to detect intrusion by reducing the complexity of the calculations [50]. Almomani et al. proposed genetic algorithm, grey wolf optimiser, firefly optimisation and PSO to evaluate the UNSW-NB15 data set through J48 ML and SVM classifiers [51]. Ahmad et al. suggested designing NIDS based on machine learning and deep learning techniques and studied the pros and cons of the proposed method [52]. Li et al. proposed a NIDS design linear nearest neighbour lasso step strategy to identify the most important advantages of intrusion detection [53]. Selvakumar et al. proposed a firefly algorithm to extract the most important features based on filter and wrapper methods. The KDD CUP 99 data set was evaluated by Bayesian networks and C4.5 classifiers, and the authors proved that 10 features were sufficient for detecting intrusions [54]. Dong et al. proposed an improved convolutional neural network for intrusion detection that processed and characterised the data in wireless networks. The system achieved accurate results and low false positive rates [55]. Sakr et al. suggested a method for intrusion detection by monitoring the movement of network data flow towards the cloud system. They used the SVM classifier to evaluate network traffic, and the SVM parameters were tuned by SPSO. They also used binary-based particle swarm optimisation to extract the most important features from the NSL-KDD data set [56]. Salo et al. presented an information gain and principal component analysis to detect network intrusion. The performance of their method was evaluated by using MLP, instance-based learning and SVM classifiers, and it achieved superior results [57]. Deng et al. studied problems of network security, intrusion detection, attacks, access control, key management and fault tolerance, as well as the importance of detecting intrusion and its connection with IoT [58]. Devan et al. presented the XGBoost method of feature extraction through a deep learning model, and the system was evaluated by the NSL-KDD test data set [59]. Magán-Carrión et al. presented a study to analyse and compare machine learning algorithms based on NIDSs while using the UGR'16 data set for intrusion detection [60]. Hajisalem et al. presented a hybrid classification method between AFS and ABC algorithms, and they also used a feature selection algorithm to show the correlation of each feature with others. They applied fuzzy c-means clustering to remove unnecessary features, determine the necessary features and, finally, evaluate the proposed system on the UNSW-NB15 and NSL-KDD data sets [61].

The main contribution of this research is to develop an intelligent security system by using advanced deep learning algorithms that detect and classify one of the serious intrusions that threaten IoT platforms. This study presents the convolutional neural network and long short-term memory (CNN-LSTM) model to detect and classify BASHLITE and Mirai attacks of security camera devices connected to IoT applications. It investigates whether the proposed system achieves superior accuracy compared to existing systems.

2. Methodology

Developing an intrusion detection system to identify and classify botnet attacks of IoT architecture is presented. The system works as an alternative method based on artificial intelligence to secure IoT devices from botnet attacks. The hybrid CNN-LSTM model is proposed to identify botnet attacks of security cameras. The system was examined using *real* network traffic extracted from four different security cameras that were connected to IoT devices while under attack from Mirai and BASHLITE botnets. The main objective was to develop a system for designing a database that could help to detect zero-day attacks while matching with botnet attack patterns available in the database. A generic architecture of the proposed system is shown in [Figure 1](#). The main mechanisms of our security system are described in the next section.

Figure 1. Proposed system

2.1 Network Botnet IoT Data Set

Network botnet IoT (N-BaIoT) network traffic data sets were collected from a machine-learning repository that contained 155 features extracted from switch ports connected to IoT environments. The data sets were gathered from four security camera devices, which are shown in Table 1, containing two botnet attacks, namely Mirai and BASHLITE. The data sets are available at this link:

(https://archive.ics.uci.edu/ml/datasets/detection_of_IoT_botnet_attacks_N_BaIoT).

Table 1. IoT devices used for developing data sets

Device type	Devices used in the model
Security camera	Provision PT-737E
Security camera	Provision PT-838
Security camera	Simple Home XCS7-1.003-WHT
Security camera	Samsung SNH 1011N

Figure 2. Lab setup design to extract botnet attack from security camera devices during IoT connection [42]

The lab setup to extract N-botnet attacks from four IoT-connected security cameras using wifi through access point devices is presented in [Figure 2](#). The port mirroring was configured on the port of switch devices to analyse and gather network packets. The extracted data were recoded using the Wireshark tool. BASHLITE and Mirai IoT attacks were injected into the data sets for extracting sub-attacks from the security cameras. These attacks are summarised in Table 2. They used various environments, including C programming and Linux systems, for collecting significant intrusion data sets.

Table 2. Data set attacks

Botnet attacks	Type of botnet attacks	Description
BASHLITE	Junk	Sending spam data
	TCP flood	Sending a flood of requests
	UDP flood	Sending a flood of requests
	Scan	Scanning the network for attacking systems
	COMBO	Sending spam data to networks
Mirai	ACK	Sending a flood of acknowledgments
	SYN	Sending a flood of synchronisations
	Plain UDP	UDP flooded by optimising seeding packet per second
	UDP flood	Scanning the network for victim devices
	Scan	

2.2 Deep Learning Algorithms

Convolutional neural network (CNN) is deep learning technique that is well known for capturing specific data in images. In the CNN model, there are two important parts: convolutional and pooling layers. A convolutional layer is a mathematical operation that can pass over an input matrix using filters for extracting local features from given input data, while the task of pooling layers is to reduce the number features of the tensor by decreasing its size. This can help the CNN technique to obtain less time complexity and lower computation cost. [Figure 3](#) displays a graphical representation of the process of convolutional and max-pooling layers in the CNN technique.

Figure 3. Graphical representation of the convolutional and max-pooling process.

The preceding layer feature map engages with the convolutional kernel using filters in the convolution operation, resulting in the convolutional layer's yielded feature map j . Each outcome feature map j may comprise a convolution involving numerous input feature maps. The convolution layer's formulas are as follows:

$$y_j^l = (\sum_{i \in c_i} t_i^{l-1} \oplus w_{ij}^{(l)} + b_j^{(l)}, \quad (1)$$

where c_i represents a set of input feature maps, $b_j^{(l)}$ indicates the bias, y_j^l is the production of the convolution and $w_{ij}^{(l)}$ is symbolic of the convolution kernel;

$$t_j^l = f(y_j^l); \quad (2)$$

where t_j^l is the feature map of the convolution layer l ; and

$$f(x) = \max(0, x) \quad (3)$$

where f is known as the activation function, which is a rectified linear unit (Relu).

2.2.1 Long Short-Term Memory

The LSTM algorithm is a deep learning model designed for a number of real-life applications in classification and prediction domains. Figure 5 outlines the architecture of the LSTM model, which is one type of recurrent neural network. The main contribution of LSTM is to process the sequence data that have a form of connection unlike feedforward neural networks (FFNN). The LSTM model is unlike common FFNN, since it has response connections and can not only process single data points and not entire sequences of data. Three significant gates, namely the input, forget and output gate are presented. The input gate is used to manage and store training data in long-term memory (LTM), while the LTM state is initialised from the existing input training data and short-term memory is initialised from the preceding state time step. The input gate employs filters for extracting significant information and abandoning non-useful information. The sigma function is used to transfer the information from input gate into the output gate and has two indicator values: 0 and 1. The significant values are indicated by the 1 value, whereas unimportant information is represented by the 0 value. The output data are stored in LTM. The forget gate is a significant gate in the LSTM model and used to selected useful information

and remove unnecessary information to enhance the capability of the model. It does so by multiplying the values of input gates and forget vector values. This information is passed from the forget gate to the output gate during the development of the new LSTM model to solve real-life problems.

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f), \quad (4)$$

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i), \quad (5)$$

$$\tilde{C}_t = \tanh(w_C[h_{t-1}, x_t] + b_C), \quad (6)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t, \quad (7)$$

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o), \quad (8)$$

$$h_t = o_t * \tanh(C_t), \quad (9)$$

where i_t refers to training data, W is weight value, B is bias, σ is activation function, f_t is the forget gate, O_t is the output gate, c_t is the cellular cell, x_t is input information and h_t output information.

Figure 5. Structure of the LSTM model

2.2.2 CNN-LSTM model

This model has two parts. One of these can be made up of the CNN technique, which involves an input layer that receives sensor parameters such as inputs, an output layer that can be used to extract features to LSTMs and numerous hidden layers. The hidden layers are utilised to contain a convolution layer, ReLU, which is the activation function that is located between layers of the CNN-LSTM network. The CNN extracts the local features and the LSTM temporal part. Through this structure, the neural network 'learns' a weight for each input that decides a precise output. The hybrid of CNN-LSTM is presented in [Figure 6](#).

Figure 6. Architecture of the hybrid CNN-LSTM model

The significant values for developing the proposed system are presented in Table 3. We considered the size of the kernel convolution to be 5, whereas that of the epoch system was 15. A graphical representation of the proposed system is presented in Figure 7, while a snapshot of it is shown in Figure 8.

Table 3. Significant parameters of the proposed CNN-LSTM system

Significant parameters	Values
Convolution layer	200
Kernel size	5
Max pooling size	5
Drop-out	0.5
Fully connected layer fitters	256
Activation function	ReLU
Epoch	20
Fitters	128

Model: "sequential"

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 115, 128)	768
conv1d_1 (Conv1D)	(None, 115, 128)	82048
lstm (LSTM)	(None, 115, 64)	49408
lstm_1 (LSTM)	(None, 115, 64)	33024
flatten (Flatten)	(None, 7360)	0
dense (Dense)	(None, 128)	942208
dense_1 (Dense)	(None, 256)	33024
dense_2 (Dense)	(None, 11)	2827
Total params: 1,143,307		
Trainable params: 1,143,307		
Non-trainable params: 0

Figure 7. Proposed structure of CNN-LSTM and implementation screen

3. Results

The empirical results of the intelligent system are presented to detect and classify botnet attacks from security camera devices.

3.1 Environmental Setup of the Proposed System

The proposed system was implemented by using different environments. Table 3 demonstrates the proposed system requirements of the design algorithm for protecting cameras from intrusion. These items of hardware and software are most appropriate for developing our intelligent security system by using advanced deep learning approaches.

Table 3. Environmental setup of the proposed system

Hardware	Software
	Matplotlib
	Numpy
Operating system (Windows 10)	Pandas
CPU (I70)	Scikit-learn
Memory (16.8 GB)	Keras
	Tensorflow
	Jupyter Python

3.2 Evaluation Metrics

In order to examine and evaluate the proposed system for detecting botnet attacks, Precision, F1 score, Accuracy and Recall metrics were used. The equations are defined as follows:

$$\text{Accuracy} = \frac{TP+TN}{FP+FN+TP+TN} \times 1.00\%, \quad (10)$$

$$\text{Precision} = \frac{TP}{TP+FP} \times 1.00\%, \quad (11)$$

$$F1 - \text{score} = 2 * \frac{\text{precision} \times \text{sensitivity}}{\text{precision} + \text{sensitivity}} \times 1.00\%, \quad (12)$$

$$\text{Sensitivity} = \frac{TP}{TP+FN} \times 1.00\%, \quad (13)$$

$$\text{Recall} = \frac{TP}{TP+FN} \times 1.00\%, \quad (14)$$

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

3.3 Results and Discussion

In order to evaluate and test the system under development, we used four data sets generated from different security camera devices and conducted four experiments on different IoT platforms. The CNN-LSTM model was implemented to detect botnet attacks by using a network data set extracted from an IoT setup. The data sets were divided into 30% testing data and 70% training data for validation of the proposed system. Table 4 shows simple input data for the security cameras.

Table 4. Input samples for commercial security cameras

Major attacks	Sub-attacks	Training data, security camera	Training data, webcam
BASHLITE	Junk attack	8788	8248
	TCP flood attack	26770	29217
	UDP flood attack	31371	30852
	Scan attack	8329	8624
	COMBO attack	17352	17923
Mirai	ACK attack	17449	32505
	Syn attack	18653	36507
	Plain UDP attack	16051	25083
	UDP flood attack	47514	47273
	Scan attack	29123	13164

In this section, the optimal results obtained from the deep learning model to detect and classify intrusion from security cameras are shown. In this experiment, we used data sets that extracted from four camera devices (Provision PT-737E, Provision PT-838, Simple Home XCS7-1.002-WHT and Simple Home XCS7-1.003-WH and Samsung SNH1011N) to develop a security system for identifying botnet attack patterns. Table 5 shows the results of the hybrid CNN-LSTM model for detecting botnet attacks of security cameras.

Table 5. Results of proposed system for detecting botnet attacks of IoT security cameras

Provision PT-737E				
Attacks	Precision	Recall	F1 score	Support
Benign	1.00	1.00	1.00	12511
Mirai_udp attack	1.00	1.00	99	12278
COMBO attack	1.00	95	97	6257
Junk attack	1.00	1.00	1.00	5906
Scan attack	56	0.00	0.00	20690
TCP attack	50	1.00	1.00	20875
UDP attack	1.00	1.00	1.00	11992
ACK attack	1.00	1.00	1.00	19455
Mirai-Scan attack	1.00	1.00	1.00	13051
Mirai-SYN attack	1.00	1.00	1.00	31329
Mirai_udpplain attack	1.00	1.00	1.00	11308
Accuracy (%)	87			
Weighted average (%)	88	91	91	
Macro average (%)	88	87	83	
Loss	0.17			
Provision PT-838				
Attacks	Precision	Recall	F1 score	Support
Benign	1.00	1.00	1.00	19747
Mirai_udp attack	1.00	1.00	1.00	11532
COMBO attack	1.00	1.00	1.00	5872

Junk attack	1.00	1.00	1.00	5777
Scan attack	85	0.00	0.00	17863
TCP attack	54	1.00	70	20766
UDP attack	1.00	1.00	1.00	11601
ACK attack	1.00	1.00	1.00	19447
Mirai-Scan attack	1.00	1.00	1.00	12256
Mirai-SYN attack	1.00	1.00	1.00	31792
Mirai_udpplain attack	1.00	1.00	1.00	10726
Accuracy (%)	89.23			
Weighted average (%)	94	89	85	
Macro average (%)	94	91	88	
Loss	0.160			

Simple Home XCS7- 1.003-WH

Attacks	Precision	Recall	F1 score	Support
Benign	1.00	1.00	1.00	9455
Mirai_udp attack	1.00	1.00	1.00	10939
COMBO attack	1.00	1.00	1.00	5728
Junk attack	1.00	1.00	1.00	5598
Scan attack	73	0.00	0.00	17790
TCP attack	54	1.00	70	20665
UDP attack	1.00	1.00	1.00	22422
ACK attack	1.00	1.00	1.00	9145
Mirai-Scan attack	1.00	1.00	1.00	25005
Mirai-SYN attack	1.00	1.00	1.00	25005
Mirai_udpplain attack	1.00	1.00	1.00	30171
Accuracy (%)	89.67			
Weighted average (%)	92	90	86	
Macro average (%)	93	91	88	
Loss	0.15			

Samsung SNH1011N IoT devices

Attacks	Precision	Recall	F1 score	Support
Benign	1.00	1.00	1.00	3957
Mirai_udp attack	97	1.00	98	11818
COMBO attack	1.00	94	97	5492
Junk attack	1.00	1.00	1.00	5728
Scan attack	62	0.00	0.00	19532
TCP attack	51	1.00	68	20576
UDP attack	1.00	1.00	1.00	21382
ACK attack	1.00	1.00	1.00	8857
Mirai-Scan attack	1.00	1.00	1.00	24348
Mirai-SYN attack	1.00	1.00	1.00	31675
Mirai_udpplain attack	1.00	1.00	1.00	16801
Accuracy (%)	88.28			
Weighted average (%)	93	90	86	
Macro average (%)	92	90	87	
Loss	0.15			

The false negative, true negative and false positive parameters were used to create confusion metrics for the proposed system to find the accuracy to detect botnet attacks of security camera devices. Figure 9 illustrates the confusion metrics of the training model for extracting the patterns from real network traffic from Simple Home XCS7-1.002-W and Simple Home XCS7-1.003-WHT security cameras. The confusion metrics show the superior accuracy of the proposed system.

Figure 8. Confusion metrics of the proposed system for a) Provision PT-737E device, b) Provision PT-838, c) Simple Home XCS7-1.002-W and d) Simple Home XCS7-1.003 devices

Figure 9 shows the performance of the CNN-LSTM model for identifying botnet attacks of the Provision PT-737E device. The accuracy increased from 80% to 87% and the accuracy loss decreased from 0.2 to 0.7; therefore, the performance of the proposed system increased.

Figure 9. Accuracy and loss of the proposed system for detecting attacks on the PT-737E security camera

The performance of CNN-LSTM model for detecting Mirai and BASHLITE attacks on the Provision PT-838 security camera is presented in Figure 10. The accuracy is shown to have increased from 82% to around 90%, and the validation loss decreased from 0.35 to 0.17.

Figure 10. Accuracy and loss of the proposed system for detecting attacks on security camera Provision PT-838

Whereas the accuracy of the Simple Home XCS7-1.002-W device increased from 82% to 89%, it is shown that the accuracy whenever the epoch increased. Figure 10 shows the accuracy and loss of the proposed system for detecting and classifying attacks. The cross-entropy losses the training loss reduced from 0.35 to 0.17 for extracting unknown attacks from the Simple Home XCS7-1.002-W device. The training loss of the

Figure 11. Accuracy and loss of the proposed system for detecting attacks on the Simple Home XCS7-1.002-W security camera

Figure 12 shows the performance of the proposed system for detecting attacks on the Simple Home XCS7-1.003-WHT device. The accuracy increased from 78% to 88% with 15 epochs. The actual loss reduced from 0.40 to 0.20, and the performance of the proposed system increased with the addition of more epochs.

Figure 12. Training loss of the CNN-LSTM model for detecting N-BaIoT attacks on the Simple Home XCS7-1.003-WHT security camera

5. Discussion

Botnet represents a robotic software program that scans for susceptible network devices and, once it locates one, adapts it into a tradition bot. Botnet attacks are common malware that comprise a connected network for hacking the victim's computers to helps to control a centralised computer by hackers, who can then simply install cyberattacks across whole networks. The botnet is a form of DDoS attack that allows cyber criminals to access a system or network that is connected to a wi-fi connection. Therefore, developing a deep learning algorithm such as the CNN-LSTM model to detect botnet attacks can protect many companies and enterprises against this type of attack. The deep learning model is used to design a database that contains numbers of botnet attacks and is installed in network system of companies. The system can help to detected botnets while extracting various unknown patterns from

the signature database. In this study, the proposed system was tested with real data extracted from real IoT camera applications.

Figure 13. ROC curves in terms of recall metrics for validation and examination of each attack on the data sets

To validate the proposed CNN-LSTM system, the receiver operating characteristic (ROC) is presented in Figure 13, where the y-axis represents the recall metric for detecting each botnet attack and normal packets. The x-axis is the specificity metric for detecting all botnet attacks on four security camera devices connected to an IoT system.

The empirical results of the hybrid CNN-LSTM model for classifying botnet attacks show that the system has the capability to identify any packet has stamp of botnet from during finding similarity new network packets with botnet attacks on signature databases. The simulation results show that our system has achieved optimal performance against the existing system, as show in Table 6. Overall, the graphics representation and confusion metrics confirm the effectiveness and efficiency of the proposed system for protecting IoT devices from botnet attacks. We have compared the proposed results with the data sets used by Soe et al., who used three machine learning algorithms (ANN, naïve bays and J48) [61]. We have noted the proposed model has achieved better supervisor accuracy.

Table 6. Comparison of proposed system (CNN-LSTM) with existing approaches [61]

Authors/ references	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	99.00
	COMBO		99.00
	Junk		98.98
	Scan		98.98
	TCP		98.95
	UDP		98.97
	ACK		99.01
	Mirai_udp	Our proposed CNN-LSTM system model	1.00
	COMBO		1.00
	Junk		1.00
	Scan		1.00
	TCP		1.00
	UDP		1.00
	ACK		1.00
	Mirai- Scan		1.00

6. Conclusion

Intelligent security systems with advanced AI for detecting serious attacks have played an essential role in protecting IoT environments from botnet attacks. The CNN-LSTM algorithm is employed to detect and classify such intrusions. Early detection of a DDoS intrusion can help network administrators to stop network operations by disconnecting all the victim IoT devices from wifi connections to prevent the spread of the botnet attack across the network.

In this study, the proposed system was evaluated and tested by employing four security cameras (Provision PT-838, Simple Home XCS7-1.002-WHT and XCS7-1.003-WHT and Samsung SNH 1011N) to contain two major IoT attacks (BASHLITE and Mirai). Whereas the BASHLITE attacks were divided into sub-attacks (Scan, COMBO, Junk, TCP flood and UDP flood) the Mirai botnet

attacks were categorised into Scan, ACK, Syn, Plain UDP and UDP flood. The CNN-LSTM was shown to effectively improve security on camera devices while they were connected to the IoT. The main objective of developing this system was to intelligently detect a serious IoT platform threat. The system can help to detect unknown botnet attacks that threaten IoT networking.

Acknowledgments: No

Funding: No fund

Conflict of Interest

The authors declare that they have no conflict of interest.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Vasilomanolakis, E.; Karuppayah, S.; Mühlhäuser, M.; Fischer, M. Taxonomy and Survey of Collaborative Intrusion Detection. *ACM Comput. Surv.* 2015, 47, 1–33. [CrossRef]
2. Hussain, B.; Du, Q.; Sun, B.; Han, Z. Deep Learning-Based DDoS-Attack Detection for Cyber-Physical System over 5G Network. *IEEE Trans. Ind. Inform.* 2021, 17, 860–870. [CrossRef]
3. de Assis, M.V.O.; Carvalho, L.F.; Rodrigues, J.J.P.C.; Lloret, J.; Proença, M.L., Jr. Near real-time security system applied to SDN environments in IoT networks using convolutional neural network. *Comput. Electr. Eng.* 2020, 86, 106738. [CrossRef]
4. Manimurugan, S.; Al-Mutairi, S.; Aborokbah, M.M.; Chilamkurti, N.; Ganesan, S.; Patan, R. Effective attack detection in internet of medical things smart environment using a deep belief neural network. *IEEE Access* 2020, 8, 77396–77404. [CrossRef]
5. Rehman Javed, A.; Jalil, Z.; Atif Moqurrah, S.; Abbas, S.; Liu, X. Ensemble Adaboost classifier for accurate and fast detection of botnet attacks in connected vehicles. *Trans. Emerg. Telecommun. Technol.* 2020. [CrossRef]
6. Lee, S.; Abdullah, A.; Jhanjhi, N.; Kok, S. Classification of botnet attacks in IoT smart factory using honeypot combined with machine learning. *PeerJ Comput. Sci.* 2021, 7, 1–23. [CrossRef]
7. Perez, M.G.; Celdran, A.H.; Ippoliti, F.; Giardina, P.G.; Bernini, G.; Alaez, R.M.; Chirivella-Perez, E.; Clemente, F.J.G.; Perez, G.M.; Kraja, E.; et al. Dynamic Reconfiguration in 5G Mobile Networks to Proactively Detect and Mitigate Botnets. *IEEE Internet Comput.* 2017, 21, 28–36. [CrossRef]
8. Wei, W.; Woźniak, M.; Damaševičius, R.; Fan, X.; Li, Y. Algorithm Research of Known-plaintext Attack on Double Random Phase Mask Based on WSNs. *J. Internet Technol.* 2019, 20, 39–48. *Electronics* 2021, 10, 1341 22 of 24 9.
9. Yong, B.; Wei, W.; Li, K.; Shen, J.; Zhou, Q.; Wozniak, M.; Połap, D.; Damaševičius, R. Ensemble machine learning approaches for webshell detection in Internet of things environments. *Trans. Emerg. Telecommun. Technol.* 2020. [CrossRef]
10. Chung, Y.Y.; Wahid, N. A hybrid network intrusion detection system using simplified swarm optimization (SSO). *Appl. Soft Comput.* 2012, 12, 3014–3022. [CrossRef]
11. Ganapathy, S.; Kulothungan, K.; Muthurajkumar, S.; Vijayalakshmi, M.; Yogesh, P.; Kannan, A. Intelligent feature selection and classification techniques for intrusion detection in networks: A survey. *EURASIP J. Wirel. Commun. Netw.* 2013, 2013, 1–16. [CrossRef]
12. Aburomman, A.A.; Reaz, M.B.I. Review of IDS development methods in machine learning. *Int. J. Electr. Comput. Eng. (IJECE)* 2016, 6, 2432–2436. [CrossRef]
13. Bijalwan, A. Botnet Forensic Analysis Using Machine Learning. *Secur. Commun. Netw.* 2020, 2020, 9302318. [CrossRef]

14. Allothman, Z.; Alkasassbeh, M.; Al-Haj Baddar, S. An efficient approach to detect IoT botnet attacks using machine learning. *J. High Speed Netw.* 2020, 26, 241–254. [CrossRef]
15. Damasevicius, R.; Venckauskas, A.; Grigaliunas, S.; Toldinas, J.; Morkevicius, N.; Aleliunas, T.; Smuikys, P. Litnet-2020: An annotated real-world network flow dataset for network intrusion detection. *Electronics* 2020, 9, 800. [CrossRef]
16. Mahmood, T.; Afzal, U. Security analytics: Big data analytics for cybersecurity: A review of trends, techniques and tools. In *Proceedings of the 2013 2nd National Conference on Information Assurance (NCIA)*, Rawalpindi, Pakistan, 11–12 December 2013; pp. 129–134.
17. Cozzi, E.; Vervier, P.A.; Dell'Amico, M.; Shen, Y.; Bilge, L.; Balzarotti, D. The Tangled Genealogy of IoT Malware. In *Proceedings of the Annual Computer Security Applications Conference*, Austin, TX, USA, 7–11 December 2020. [CrossRef]
18. Kolias, C.; Kambourakis, G.; Stavrou, A.; Voas, J. DDoS in the IoT: Mirai and Other Botnets. *Computer* 2017, 50, 80–84. [CrossRef]
19. Hoque, N.; Bhattacharyya, D.K.; Kalita, J.K. Botnet in DDoS Attacks: Trends and Challenges. *IEEE Commun. Surv. Tutor.* 2015, 17, 2242–2270. [CrossRef]
20. McDermott, C.D.; Majdani, F.; Petrovski, A.V. Botnet Detection in the Internet of Things using Deep Learning Approaches. In *Proceedings of the International Joint Conference on Neural Networks*, Rio de Janeiro, Brazil, 8–13 July 2018; Volume 2018.
21. Koroniotis, N.; Moustafa, N.; Sitnikova, E. Forensics and Deep Learning Mechanisms for Botnets in Internet of Things: A Survey of Challenges and Solutions. *IEEE Access* 2019, 7, 61764–61785. [CrossRef]
22. Yerima, S.Y.; Alzaylaee, M.K.; Shajan, A.; Vinod, P. Deep learning techniques for android botnet detection. *Electronics* 2021, 10, 519. [CrossRef]
23. Marir, N.; Wang, H.; Feng, G.; Li, B.; Jia, M. Distributed abnormal behavior detection approach based on deep belief network and ensemble svm using spark. *IEEE Access* 2018, 6, 59657–59671. [CrossRef]
24. Azeez, N.A.; Ayemobola, T.J.; Misra, S.; Maskeliunas, R.; Damaševičius, R. Network intrusion detection with a hashing based τ -apriori algorithm using Hadoop MapReduce. *Computers* 2019, 8, 86. [CrossRef]
25. Tuan, T.A.; Long, H.V.; Son, L.H.; Kumar, R.; Priyadarshini, I.; Son, N.T.K. Performance evaluation of Botnet DDoS attack detection using machine learning. *Evol. Intell.* 2020, 13, 283–294. [CrossRef]
26. Kebande, V.R.; Venter, H.S. A cognitive approach for botnet detection using Artificial Immune System in the cloud. In *Proceedings of the 2014 Third International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec)*, Beirut, Lebanon, 29 April–1 May 2014; pp. 52–57.
27. Da, K. A method for stochastic optimization. *arXiv* 2014, arXiv:1412.6980.
28. Zeiler, M.D. Adadelta: An adaptive learning rate method. *arXiv* 2012, arXiv:1212.5701.
29. Duchi, J.; Hazan, E.; Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* 2011, 12, 2121–2159.
30. Rauf, H.T.; Malik, S.; Shoaib, U.; Irfan, M.N.; Lali, M.I. Adaptive inertia weight Bat algorithm with Sugeno-Function fuzzy search. *Appl. Soft Comput.* 2020, 90, 106159. [CrossRef]
31. Ullah, I.; Mahmoud, Q.H. A two-level flow-based anomalous activity detection system for IoT networks. *Electronics* 2020, 9, 530. [CrossRef]
32. Dong, B.; Wang, X. Comparison deep learning method to traditional methods using for network intrusion detection. In *Proceedings of the 2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)*, Beijing, China, 4–6 June 2016; pp. 581–585.
33. Folorunso, O.; Ayo, F.E.; Babalola, Y. Ca-NIDS: A network intrusion detection system using combinatorial algorithm approach. *J. Inf. Priv. Secur.* 2016, 12, 181–196. [CrossRef]
34. Deng, L. A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Trans. Signal Inf. Process.* 2014, 3, e2. [CrossRef]

35. Berman, D.S.; Buczak, A.L.; Chavis, J.S.; Corbett, C.L. A survey of deep learning methods for cyber security. *Information* 2019, 10, 122. [CrossRef]
36. Yilmaz, S.; Sen, S. Early Detection of Botnet Activities Using Grammatical Evolution. In *Applications of Evolutionary Computation*; Springer International Publishing: Berlin/Heidelberg, Germany, 2019; pp. 395–404. [CrossRef]
37. Yu, Y.; Long, J.; Liu, F.; Cai, Z. Machine learning combining with visualization for intrusion detection: A survey. In *Proceedings of the International Conference on Modeling Decisions for Artificial Intelligence*, Sant Julià de Lòria, Andorra, 19–21 September 2016; pp. 239–249. *Electronics* 2021, 10, 1341 23 of 24
38. Ahmed, A.A.; Jabbar, W.A.; Sadiq, A.S.; Patel, H. Deep learning-based classification model for botnet attack detection. *J. Ambient. Intell. Humaniz. Comput.* 2020. [CrossRef]
39. Alauthman, M.; Aslam, N.; Al-kasassbeh, M.; Khan, S.; Al-Qerem, A.; Raymond Choo, K. An efficient reinforcement learningbased Botnet detection approach. *J. Netw. Comput. Appl.* 2020, 150, 102479. [CrossRef]
40. Mazini, M.; Shirazi, B.; Mahdavi, I. Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms. *J. King Saud Univ. Comput. Inf. Sci.* 2019, 31, 541–553. [CrossRef]
41. Asadi, M.; Jabraeil Jamali, M.A.; Parsa, S.; Majidnezhad, V. Detecting botnet by using particle swarm optimization algorithm based on voting system. *Future Gener. Comput. Syst.* 2020, 107, 95–111. [CrossRef]
42. Al Shorman, A.; Faris, H.; Aljarah, I. Unsupervised intelligent system based on one class support vector machine and Grey Wolf optimization for IoT botnet detection. *J. Ambient Intell. Humaniz. Comput.* 2020, 11, 2809–2825. [CrossRef]
43. Lin, K.C.; Chen, S.Y.; Hung, J.C. Botnet Detection Using Support Vector Machines with Artificial Fish Swarm Algorithm. *J. Appl. Math.* 2014, 2014, 1–9. [CrossRef]
44. Rajagopal, S.; Kundapur, P.P.; Hareesha, K.S. A stacking ensemble for network intrusion detection using heterogeneous datasets. *Secur. Commun. Netw.* 2020, 2020, 4586875. [CrossRef]
45. Elmasry, W.; Akbulut, A.; Zaim, A.H. Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic. *Comput. Netw.* 2020, 168, 107042. [CrossRef]
46. Dwivedi, S.; Vardhan, M.; Tripathi, S. Defense against distributed DoS attack detection by using intelligent evolutionary algorithm. *Int. J. Comput. Appl.* 2020, 1–11. [CrossRef]
47. Suhaimi, H.; Suliman, S.I.; Musirin, I.; Harun, A.; Mohamad, R.; Kassim, M.; Shahbudin, S. Network intrusion detection system using immune-genetic algorithm (IGA). *Indones. J. Electr. Eng. Comput. Sci.* 2020, 17, 1059–1065. [CrossRef]
48. Zhou, Y.; Mazzuchi, T.A.; Sarkani, S. M-AdaBoost—A based ensemble system for network intrusion detection. *Expert Syst. Appl.* 2020, 162, 113864. [CrossRef]
49. Wu, Z.; Wang, J.; Hu, L.; Zhang, Z.; Wu, H. A network intrusion detection method based on semantic re-encoding and deep learning. *J. Netw. Comput. Appl.* 2020, 164, 102688. [CrossRef]
50. Injadat, M.; Moubayed, A.; Nassif, A.B.; Shami, A. Multi-Stage Optimized Machine Learning Framework for Network Intrusion Detection. *IEEE Trans. Netw. Serv. Manag.* 2020. [CrossRef]
51. Almomani, O. A Feature Selection Model for Network Intrusion Detection System Based on PSO, GWO, FFA and GA Algorithms. *Symmetry* 2020, 12, 1046. [CrossRef]
52. Ahmad, Z.; Khan, A.S.; Shiang, C.W.; Abdullah, J.; Ahmad, F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Trans. Emerg. Telecommun. Technol.* 2020, 32. [CrossRef]
53. Li, X.; Yi, P.; Wei, W.; Jiang, Y.; Tian, L. LNNLS-KH: A Feature Selection Method for Network Intrusion Detection. *Secur. Commun. Netw.* 2021, 2021, 8830431. [CrossRef]
54. Selvakumar, B.; Muneeswaran, K. Firefly algorithm based feature selection for network intrusion detection. *Comput. Secur.* 2019, 81, 148–155. [CrossRef]

55. Dong, Q.L.; He, S.N. Self-adaptive projection algorithms for solving the split equality problems. *Fixed Point Theory* 2017, 18, 191–202. [CrossRef]
56. Sakr, M.M.; Tawfeeq, M.A.; El-Sisi, A.B. Network Intrusion Detection System based PSO-SVM for Cloud Computing. *Int. J. Comput. Netw. Inf. Secur.* 2019, 11, 22–29. [CrossRef]
57. Salo, F.; Nassif, A.B.; Essex, A. Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection. *Comput. Netw.* 2019, 148, 164–175. [CrossRef]
58. Deng, L.; Li, D.; Yao, X.; Wang, H. RETRACTED ARTICLE: Mobile network intrusion detection for IoT system based on transfer learning algorithm. *Clust. Comput.* 2018, 22, 9889–9904. [CrossRef]
58. Devan, P.; Khare, N. An efficient XGBoost–DNN-based classification model for network intrusion detection system. *Neural Comput. Appl.* 2020, 32, 12499–12514. [CrossRef]
59. Magán-Carrión, R.; Urda, D.; Díaz-Cano, I.; Dorronsoro, B. Towards a Reliable Comparison and Evaluation of Network Intrusion Detection Systems Based on Machine Learning Approaches. *Appl. Sci.* 2020, 10, 1775. [CrossRef]
60. Hajisalem, V.; Babaie, S. A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection. *Comput. Netw.* 2018, 136, 37–50. [CrossRef]
61. Soe, Yan N., Yaokai Feng, Paulus I. Santosa, Rudy Hartanto, and Kouichi Sakurai. 2020. "Machine Learning-Based IoT-Botnet Attack Detection with Sequential Architecture" *Sensors* 20, no. 16: 4372. <https://doi.org/10.3390/s2016437>