JOIV : Int. J. Inform. Visualization, 5(4) - December 2021 380-387

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Combining Deep Learning Models for Enhancing the Detection of Botnet Attacks in Multiple Sensors Internet of Things Networks

Abdulkareem A. Hezam [a], Salama A. Mostafa [a,1], Zirawani Baharum [b,2], Alde Alanda [c], Mohd Zaki Salikon [d]

[a] *Center of Intelligent and Autonomous Systems (CIAS), Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Parit Raja, Johor, Malaysia.*

[b] *Malaysian Institute of Industrial Technology, Universiti Kuala Lumpur, Persiaran Sinaran Ilmu, Bandar Seri Alam, Johor Bahru, Malaysia*

[c] *Department of Information Technology, Politeknik Negeri Padang, West Sumatera, Indonesia*

[d] *Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Parit Raja 86400, Johor, Malaysia*

Corresponding author: [1]salama@uthm.edu.my; [2]zirawani@unikl.edu.my

Abstract*— Distributed-Denial-of-Service impacts are undeniably significant, and because of the development of IoT devices, they are expected to continue to rise in the future. Even though many solutions have been developed to identify and prevent this assault, which is mainly targeted at IoT devices, the danger continues to exist and is now larger than ever. It is common practice to launch denial of service attacks in order to prevent legitimate requests from being completed. This is accomplished by swamping the targeted machines or resources with false requests in an attempt to overpower systems and prevent many or all legitimate requests from being completed. There have been many efforts to use machine learning to tackle puzzle-like middle-box problems and other Artificial Intelligence (AI) problems in the last few years. The modern botnets are so sophisticated that they may evolve daily, as in the case of the Mirai botnet, for example. This research presents a deep learning method based on a real-world dataset gathered by infecting nine Internet of Things devices with two of the most destructive DDoS botnets, Mirai and Bashlite, and then analyzing the results. This paper proposes the BiLSTM-CNN model that combines Bidirectional Long-Short Term Memory Recurrent Neural Network and Convolutional Neural Network (CNN). This model employs CNN for data processing and feature optimization, and the BiLSTM is used for classification. This model is evaluated by comparing its results with three standard deep learning models of CNN, Recurrent Neural Network (RNN), and long-Short Term Memory Recurrent Neural Network (LSTM–RNN). There is a huge need for more realistic datasets to fully test such models' capabilities, and where N-BaIoT comes, it also includes multi-device IoT data. The N-BaIoT dataset contains DDoS attacks with the two of the most used types of botnets: Bashlite and Mirai. The 10-fold cross-validation technique tests the four models. The obtained results show that the BiLSTM-CNN outperforms all other individual classifiers in every aspect in which it achieves an accuracy of 89.79% and an error rate of 0.1546 with a very high precision of 93.92% with an f1-score and recall of 85.73% and 89.11%, respectively. The RNN achieves the highest accuracy among the three individual models, with an accuracy of 89.77%, followed by LSTM, which achieves the second-highest accuracy of 89.71%. CNN, on the other hand, achieves the lowest accuracy among all classifiers of 89.50%.

***Keywords*—DDoS; deep learning; classification; IoT; RNN; LSTM-RNN; BiLSTM-CNN.**

Manuscript received 13 Jun. 2021; revised 15 Oct. 2021; accepted 29 Nov. 2021. Date of publication 31 Dec. 2021.

I. INTRODUCTION

The huge growth in Internet usage has also led to some very dangerous cybercrimes such as DDoS and others of the same nature [1]. A massive scaled DDoS botnet attack in 2016 has compromised over 100,000 IoT devices targeting Dyn DNS infrastructure. Another one was used against Carphone Warehouse in 2015. It was utilized as a distraction which gave the hackers access to 2.4 million customers' personal information. When a hacker attempts to render a computer or network asset to anticipated customers by interrupting the services of a host connected to the Internet, this is known as DDoS. It is a targeted attack that earmarks on attacking websites with more traffic than the website or server can accommodate. It results in shutting down the website or server, and since that very device is being under attack. The CPU operates in maximum workload, which slows down the performance of the running applications, including antiviruses and Internet protection applications. The huge

capacity of the attack leaves the firewall paralyzed, which makes it easy for the hacker to get into the device aimed to hack. This host-based attack turns the hacked device into a zombie and takes advantage of it being a trusted source to the clout, and it eases for the hackers to make their way to hacking the cloud.

With the age of the Internet, Internet of Things (IoT) devices have entered our lives, and they are in use in so many forms (e.g., smartwatches, smart bills, webcams, smart houses, etc.). Those devices are fundamentally insecure, and that goes back to the massive scale and distributed nature of IoT networks [2]. IoT devices connect the virtual world with real things. Intelligent devices and equipment are linked to one another and the Internet. They collect information about their immediate surroundings using sensors and then evaluate it and connect it to a network. An IoT architecture at a high level has four components: applications and analytics, integration, security, and infrastructure [3], [4]. However, the integration of these four different components in one architecture is the same for all IoT systems. It has the form of layers of sensors/devices, connection network, data processing, and user interface. Fig. 1 shows a general IoT network architecture.

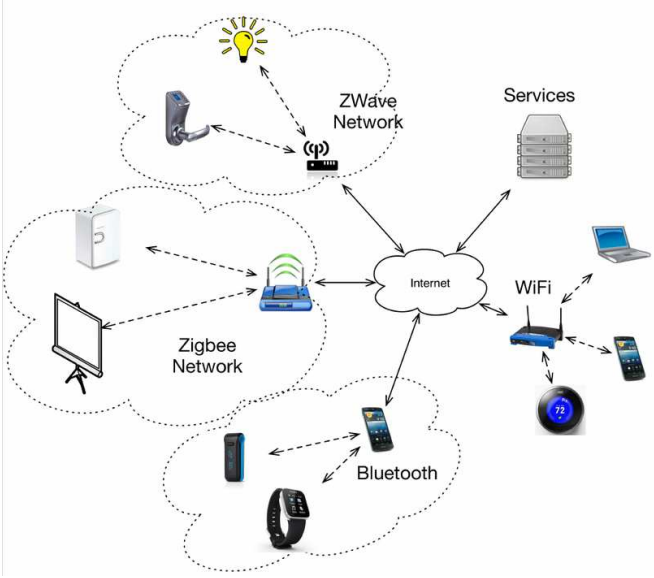

Fig. 1 An IoT network architecture [5]

Subsequently, IoT devices are projected to grow to 13.6 percent per year through 2022 as they will reach 43 billion devices by 2023 [6]. However, this huge growth for IoT devices comes with it cyber attack threats. Through time DDoS has gotten more powerful and more sophisticated. The newest example is Mirai malware. It is a worm mutant that goes far beyond severity. The Mirai botnet is built through a series of different operating stages, including propagation, infection, command, and control (C&C) communication, and execution of an attack [8]. This Mirai-based attack against Dyn DNS infrastructure has led to unavailable major internet platforms [7]. At the beginning of 2017, the source code of the Mirai botnet was released to the public led to a huge increase of DDoS attacks using Mirai-derived IoT botnets [8].

To create a DDoS attack on a system, two major stages are required. The first stage includes the use of malicious packets to disrupt the protocols or running programs by an attacker to victims' computers. Second, to initiate flood assaults by depleting a server or network resources like bandwidth, storage, router's processing capabilities, disk/database, an attacker uses these zombies [9].

IoT devices are projected to grow to 13.6 percent per year through 2022 as they will reach 43 billion devices by the year 2023 [6]. This huge growth for IoT devices comes with its threats. Through time DDoS has gotten more powerful and more sophisticated. The newest example is Mirai malware. It is a worm mutant that goes far beyond severity. Several different operational stages are involved in the construction of the Mirai botnet, including sc. propagation of the botnet, infection, C&C communication, and execution of an attack [10].

In October 2016, Mirai botnet commanded more than 100,000 IoT devices to perform a DDoS attack against (Dyn DNS infrastructure) [11]. This cyberattack has led to major internet platforms being unavailable. At the beginning of 2017, the source code of the Mirai botnet was released to the public led to a huge increase of DDoS attacks using Mirai-derived IoT botnets [12]. The severity of such an attack can impact on a huge scale. The massive growth of IoT devices and the ignorance of some manufacturers on the security of these devices could result in a tremendous issue regarding trusting new technology whose primary job is to make our lives easier.

There has been significant work regarding detecting such attacks, such as the one provided in [13]. The problem is that Distributed-Denial-of-Service "DDoS impacts are doubtlessly major, and it will continue to grow along with the growth of Internet-of-Things "IoT" devices. So many solutions have been contributed to detecting and mitigating this attack, specifically in IoT devices, yet the threat still exists and is more substantial than ever. One of which is using a machine learning pipeline that captures the traffic and identifies whether it is benign or there is an attack [14].

Creating a machine learning pipeline to execute, as described in many papers, such as in the paper titled "data collecting, feature extraction, and binary classification for IoT traffic DDoS detection," is one of the popular and effective approaches. The features are intended to use IoT-specific network behaviors while also using network flow parameters, including packet length, inter-packet intervals, and protocols. [11]. Yet, this method is more likely to be outdated where it focuses on the early steps of the infection. New IoT devices (e.g., smartwatches) connect to some public free Wi-Fi. The malware will be installed in the device at the moment of pairing. Mirai worm is a mutant, so it cannot be detected by the machine learning pipeline method as that method is in the first layer of defense. A new way has been proposed which focuses on the later steps of infection [15]. The solution is to add last layer security using a deep autoencoder. This will take snapshots at the step of lurching the attack. It instantly detects the compromised device and makes an alert recommending disconnecting the infected device until being sanitized.

In this paper, we propose a combination of RNNs with CNNs through BiLSTM and CONVNET-1D. In such a manner, the model shall use a CNN model for feature extraction and an RNN layer for interpreting the features across time steps.

This section will review the most recently released methods of detecting DDoS botnets. So much effort and so many people have devoted great work to mitigating such threats, especially in an IoT environment. The threat still exists now more than ever with the help of insecure IoT devices. Many methods are created daily to mitigate such threats, like adding more bandwidth so your server or website does not get overwhelmed or building redundancy into your infrastructure yet, the development of DDoS is way more powerful than such prevention methods. However, Machine learning has been the best hope to detect and mitigate DDoS attacks. There have been so many published papers on solving the mystery of DDoS attacks in IoT environments using ML. State-of-art solutions have been increasing rapidly, as the studies have shown in [10], [16]–[19], but DDoS methods of attacking are changing and coping with any new defense systems. They may be described as not primitive anymore where the new botnets have become so complex and sophisticated to be detected.

The work of [11] has created a machine learning pipeline for detecting DDoS in IoT traffic that collects data, extracts features, and classifies binary data. Network flow characteristics such as packet length, interpacket interval, and protocol are used to take advantage of IoT network behavior. The outcomes were outstanding where it can identify attacks with accuracy higher than 0.999. However, such a method focused mainly on the early steps of propagation and communication with the command-and-control server nonetheless gives the time for the botnet to continue growing. Such malware like "Mirai "is so sophisticated and can mutate daily, making it hard to spot [20]. As proposed, the main reason for the insecurity of IoT devices is that they do not possess enough memory [21].

The work of [22] proposed a system using a sequential architecture framework to detect DDoS attacks. The system showed quite a remarkable result and performance, achieving 99% for botnet detection using three ML algorithms of Artificial Neural network, J48 decision tree, and Naive Bayes classifiers [22]. The system is divided into two phases. The first is "Model Builder," where it conducts 1) data collection, 2) data organization, 3) model training, and 4) feature selection. The second phase is the "Attack detector," where it detects the attack sequentially. As the data gets to the pre-processing phase, it runs into two steps. The first is to encode the packets sent, and the second one is to extract the features of the packet, then it shall detect the attack based on the information given by the feature extraction step.

With that being said, The work of [23] proposes a deep learning model to detect the attacks coming from compromised IoT devices within the network. The proposed system will use conventional machine learning models to collect the network flows and convert them into connection records, then use a deep learning model to detect which device the attack is coming from.

Using a new method of network-based anomaly detection proposed and empirically tested by [15], a new approach to detecting anomalous network traffic from compromised IoT devices is proposed. This method collects network activity snapshots and uses deep autoencoders to identify abnormal network traffic. As part of the research, they infected nine commercial Internet of Things (IoT) devices in their lab with Mirai and BASHLITE, two well-known IoT-based botnets.

The results of the tests revealed that the suggested method was capable of detecting attacks launched by hacked IoT devices that were part of a botnet reliably and promptly. Instead of focusing on the initial phase of the botnet operation as described in [24] and [25], they focused on the latter stages of the operation. Evidently, some IoT devices connect automatically to free WI-FI, such as smartwatches with small memory. It needs to sync the data so it connects to public-free WI-FI, which puts it at risk of being infected. Such a method adds the last line of defense in terms of security. The proposed system will detect which device within the network is compromised and instantly send an alert to inform the monitor to isolate the device and sanitize it.

The work of [26], this paper claims to overcome the glitches found on middle-box's [27] high cost and software-based [28] high-performance overhead by proposing a programmable switches defense layer that will address the above limitations. These switches have the features of being reconfigured within the field without the need for additional hardware upgrades. The users of this system can designate their defense strategies in a modular fashion. The system then draws the boundaries of the defense to run on the programmable switches to encounter the new attack patterns. Once the attack changes, the system can reconfigure the patterns to respond to the new attack. The evaluation using the prototype shows that the system can effectively defend against high volume attacks with new features apart from the known systems "middle-box and software-based "such as:

- It easily supports the customization of defense strategies.
- Adapt to dynamic attacks with low overheads

This section will cover and discuss the related work based on the algorithms used, machine learning techniques, and the results of each model, as shown in Table I. As for the first paper, they manually generated their dataset and used a machine learning pipeline recruiting five ML techniques, namely, 1) KNN "KDTree" algorithm, 2) LSVM with linear kernel, 3) DT using Gini impurity scores, 4) RF using Gini impurity scores, and 5) NN. Except for the LVSM classifier, the models employed exhibited an accuracy of 99.9%, indicating that the data were not linearly separable.

As for the second paper, where they used a sequential detection scheme with the help of the N-BaIoT dataset, in Fig. 2, we see the techniques used as they ran through different sequences of testing. In such a study, they tested the models through a connection-orientated "TCP" and a connectionless "UDP" protocols where you see the hybrid classifier outperformed the rest.

The third paper utilized distributed deep learning and picked two models. CNN, where they created the dataset for it manually, which showed 0.943 accuracies, and RNN-LSTM, which they assigned the N-BaIoT dataset to it, RNN-LSTM has outperformed CNN as the accuracy for it was 0.948.

The work of [15] has proposed a new approach to detecting DDoS attacks in the IoT environment "deep autoencoder" as newly introduced. Alongside it used three other machine learning techniques, to be specific, the Local Outlier Factor,

the Isolation Forest, and the support vector machine. Using the N-BaIoT dataset as an autoencoder obtained an FPR of zero.

TABLE I.
COMPARISON ACCURACY RATE BETWEEN VARIOUS ARTICLES

Ref.	Model	Accuracy	Datasets
[11]	KN	99.90	Generated their own
	LVSM	99.10	
	DT	99.90	
	DF	99.90	
	NN	99.90	
	NB	80.01	
[22]	Hybrid	99.02	N-BaIoT
	J48	99.04	
[29]	ANN	99.11	Their own N-BaIoT
	CNN	94.30	
	RNN-LSTM	94.80	
[15]	Isolation Forest	-	N-BaIoT
	SVM	-	
	Autoencoder	-	

II. MATERIALS AND METHOD

In this research, CHRISP-DM was used. The data mining research methodology is mainly used for achieving the research objectives [30]. However, it will be very beneficial to be used in any machine learning project with some very logical steps that could cover and assist almost any project without any regard to its nature. Data mining research methodology is the abbreviation of the Cross-industry standard process for data mining which refers to the process model that gives a system to the carrying-out data mining project. The data mining research methodology is meant to do large mining projects, more reliable, less expensive, more repeatable, quicker, and more achievable. The development model that is utilized for this research is the data mining research methodology reference model. This model was picked in light of the fact that it outlines the project life cycle. Fig. 2 shows the research methodology, consisting of four main steps: collection, preparation, feeding the data to the classifier, and evaluating the results.

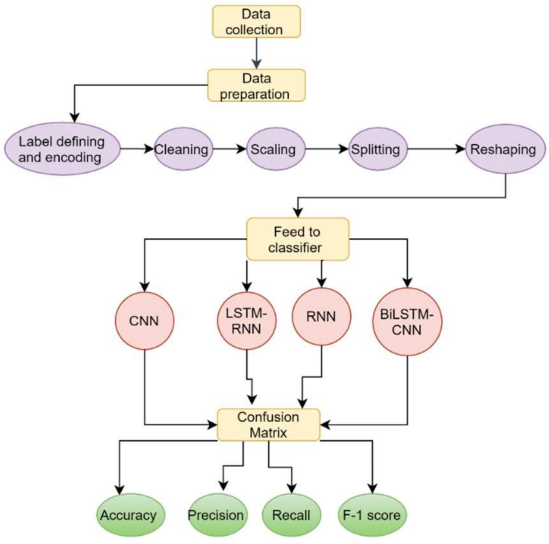

Fig. 2 Research Methodology

Each step consists of some other steps. For instance, the data should be first cleaned (dealing with missing values) in preparing the dataset and then define and encode the labels. Then a feature scaling stage takes place before splitting the data into testing and training. In our research, we used 10-folds cross-validation. At the very last step of preparing the data, we reshaped the data to the shape expected by the network. In that manner, a third primary step is to feed them to the model and start the training stage. Eventually, the final sage evaluates the result and calculates the confusion matrix as comparison data.

A. Dataset

The work of [17] infected nine commercial IoT devices (i.e., doorbell, thermostat, baby monitor, security camera, and webcam) with the most recent DDoS malware like Mirai and Bashlite to better test and study DDoS attacks using real traffic data. The deep autoencoder used in their research was trained on benign examples of normal behavior so that it could be taught to replicate the inputs. Reconstruction of normal senses becomes easier with this method. The problem was that it didn't work when trying to recreate unusual observations (unknown behavior). Anomalies were defined as those observations that arose as a result of the reconstruction error. The framework was assessed on one dataset and had the option to distinguish the unusual traffic adequately. The researchers made the trace traffic of the dataset accessible on the University of California Irvine online repository and are available at the DATASET website. We have used a subset of the N-BaIoT dataset in this project, namely:

- Ecobee-Thermostat,
- Philips-B120N10-Baby-Monitor.
- Provision-PT-838-Security-Camera,
- SimpleHome-XCS7-1002-WHT-Security-Camera.

Fig. 3 shows the general structure of the dataset.

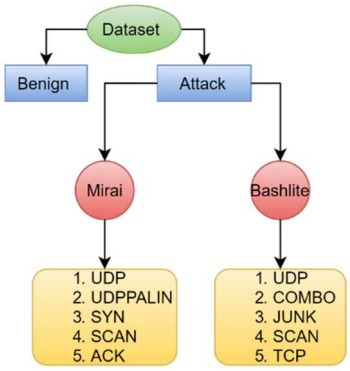

Fig. 3 Dataset Architecture [17] .

Table II shows the names of the used datasets that correspond to four things. These things are operating in the IoT environment and exposing them to attacks.

The main objective in this project is to use a neural network to classify the requests sent using these devices and see how well each classifier would classify them into malicious "and which type" and normal.

TABLE II
THE USED DATASETS

No.	Dataset name	Symbol
1.	Ecobee-Thermostat	A
2.	Philips-B120N10-Baby-Monitor	B
3.	Provision-PT-838-Security-Camera	C
4.	SimpleHome-XCS7-1002-WHT-Security-Camera	D

B. Deep Learning Models

In this work, we are investigating the suitability of the DL algorithms in detecting DDoS attacks in IoT environments. Fig 4 shows the architecture of the three classifiers they were first chosen to be compared in this research.

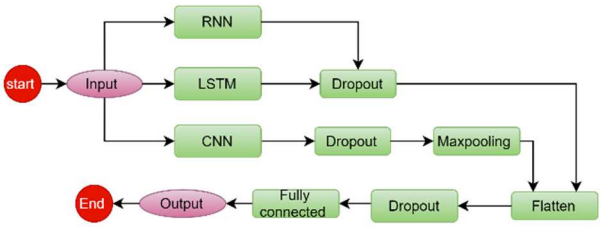

Fig. 4 CNN, RNN and, LSTM architectures

1) Recurrent Neural Network: As the output from previous stages is fed into the current step, it resembles a neural network. If you use a conventional neural network, neither inputs nor outputs depend on each other. However, there are times when anticipating what a sentence will say next necessitates remembering what came before. As a result, RNN was created, which used a hidden layer to resolve the problem. The hidden state is the most important part of an RNN since it stores data about a sequence. Consider a network with four layers: an input layer, a hidden layer, and an output layer. This is called a deep network. When that happens, each hidden layer will have its own weights and biases, such as (w1, b1) for the first hidden layer, (w2, b2) for the second hidden layer, and (w3, b3) for the third hidden layer, just as in a normal neural network. In other words, each of these layers is distinct from the others. As an example, they aren't aware of past results [31].

2) Convolutional Neural Network: A ConvNet is a deep learning algorithm. Although it is mainly used for image classification, it can be used for time series datasets using (ConvNet 1D) which will give input and output of 2 dimensional where the first dimension is the timesteps, and the other one is the value of acceleration [32]. The architecture of a ConvNet is inspired by the layout of the Visual Cortex and is comparable to the linking network of neurons in the human brain. The Receptive Field is a small portion of the visual field in which individual neurons are responsive to stimuli. The whole visual field is covered by a collection of identical fields that are all overlapping one another. [33]. For instance, the argument input shape (115, 3) indicates 115-time steps with three data points in each time step. These three data points represent acceleration along the x, y, and z axes. Kernel size is set to 5, signifying the width of the kernel, and kernel height is set to the number of data points in each time step[33].

Convolution is a mathematical procedure that reduces a tensor, matrix, or vector to a smaller size, as described above. An input matrix with only one dimension may be summarized along that axis, whereas one with several dimensions can be summarized along all of them at the same time. One-dimensional convolution (Conv1D) and two-dimensional convolution (Conv2D).

$$bi = \sum_{j=m-1}^{0} \blacksquare ai + j * wj \quad (1)$$

where $i = [1, n - m + 1]$

3) Long-Short Term Memory Recurrent Neural Network (LSTM-RNN): Input, recurrent LSTM layer, and output layers are all included in the typical LSTM-RNN architecture. It's important to mention that the LSTM input layer is linked to the output LSTM input layer. An input gate is an input device that receives data from another input device, such as a transistor or a transistor-based device. Additional connections are made between the output units and the output layer of the network through the cell. When biases are ignored, the total number of parameters in a typical LSTM network with one cell per memory block is calculated as below.

$$N = n_c \times n_c \times 4 + n_i \times n_c \times 4 + n_c \times n_o + n_c \times 3 \quad (2)$$

where n_c is the number of memory cells, n_i is the number of input units, and n_o is the number of output units [35]. The Long Short Term Perspectives Error analysis flow in existing RNNs prompted the memory design since lengthy delays were far from current structures due to back propagated error, which bursts or rots significantly in both directions. Layers of LSTM are made up of memory blocks, which are a collection of recurrently linked blocks. [36]. We can think of these squares as a differentiable variant of the memory chips of an advanced PC. For every recurrently linked memory cell, three multiplicative units (the input, output, and forget gates) provide constant analogies of write, read and reset operations for the cells. The net and cells can only work together while the gates are open [35].

4) Bidirectional Long-Short Term Memory Recurrent Neural Network and CNN: In this model, we combined both CNN and BiLSTM as a newly introduced method. Replicating the first recurrent layer is part of BiLSTM. When the input sequence is sent to the first layer as-is, a reversed copy of it is fed to the second layer. There are now two layers next to each other [36], [37]. These two models are strong, which works in our favor.

Evidently, CNNs and RNNs are strong models as individuals. Ergo, combining such models shall yield great results. Fig. 5 shows the building structure of BiLSTM-CNN. In this experiment, we used three CNN layers. A max-pooling layer follows each to avoid overfitting and one layer of BiLSTM with 124 units.

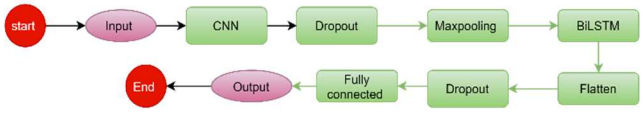

Fig. 5 BiLSTM-CNN architecture

C. Evaluation metrics

Classifier performance is assessed using a confusion matrix. The confusion matrix is defined as a table that is used to describe the performance of the classifiers [38], [39]. A relative explanation of the evaluation metric steps is carried out as below [9], [33].

1) Accuracy: It indicates how many predictions were correct. Where TN = True Negative as described by the formula

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

2) Recall: Recall is the percentage of the total number of relevant occurrences found. Because of this, it significantly relies on an understanding and ability to assess.

$$recall = \frac{TP}{TP + FN} \quad (4)$$

3) Precision: The percentage of anticipated positive instances may be calculated using a formula where P is the parameter. True Positivity is represented by TP. FP = False Positive. And it shares some common features with recall as it is too based on an understanding and measure of relevance.

$$precision, p = \frac{TP}{TP + FP} \quad (5)$$

4) F-1 score: It is the relation between precision and recall

$$F1 = 2 \times \frac{Precision \times recall}{Precision + recall} \quad (6)$$

III. Results and Discussion

Every classifier has a specific setting to its parameters. Some parameters of different classifiers have the same setting. Table III does include some of the parameters that are common across the implemented models.

TABLE III
SHARED PARAMETERS

Parameter	Value or state
Batch size	1024
Input	115,1
Learning rate	0.001
Epsilon	1e-07
Optimizer	Adam
Epochs	100
Verbose	1
Activation function	SoftMax, ReLU
Output	11

This research initially compared three deep learning classifiers: CNN, LSTM-RNN, and RNN, and studied DDoS attacks using the N-BaIoT dataset. However, the dataset was not used fully as we just used a subset of it utilizing just four IoT devices from the nine devices that were used in [17]. The results of RNN, CNN, LSTM-RNN, and BILSTM-CNN are shown in Table IV. The results came out a fine success given the fact that every classifier's accuracy did not go below 89%. However, by the look at Table IV, we see each dataset has achieved different results in each classifier yet, the baby monitor dataset has achieved the highest, and that is for a very obvious reason: the size of the dataset itself. Each model has achieved high accuracy.

TABLE IV
THE RNN, CNN, LSTM-RNN, AND BILSTM-CNN RESULTS

Dataset	Accuracy	Error	precision	F1 score	Recall
RNN					
A	0.8802	0.1950	0.9420	0.8694	0.8965
B	0.9221	0.1200	0.9559	0.8803	0.9078
C	0.8897	0.1810	0.9470	0.8717	0.9001
D	0.8897	0.1810	0.9470	0.8717	0.9001
ABCD	0.8954	0.1690	0.9479	0.8732	0.9011
CNN					
A	0.8790	0.1910	0.9383	0.8705	0.8995
B	0.9173	0.1340	0.9283	0.8682	0.8948
C	0.8931	0.1630	0.9362	0.8777	0.9047
D	0.8897	0.1810	0.947	0.8717	0.9001
ABCD	0.8947	0.1672	0.9374	0.8720	0.8997
LSTM-RNN					
A	0.8822	0.3510	0.8608	0.7566	0.7906
B	0.9214	0.1470	0.8843	0.7362	0.8040
C	0.8897	0.1680	0.9242	0.8446	0.8783
D	0.8930	0.1780	0.8905	0.7903	0.8422
ABCD	0.8965	0.2110	0.8899	0.7819	0.8287
BiLSTM-CNN					
A	0.8833	0.1710	0.9551	0.8790	0.9072
B	0.9226	0.1170	0.9573	0.8813	0.9083
C	0.8894	0.1640	0.9540	0.8786	0.9070
D	0.8930	0.1780	0.8905	0.7903	0.8422
ABCD	0.8970	0.1575	0.9392	0.8573	0.8911

When compared to the three individual classifiers RNN, LSTM, and CNN, we observe that RNN has achieved the highest accuracy and lowest error rate, yet it has fallen to CNN regarding precision. RNN accuracy and error rate are great, but when compared with the others regarding other aspects such as sensitivity, recall, and f-1 score, it goes all to CNN as the highest and followed by LSTM. Fig. 6 shows the overall results of the four DL classifiers.

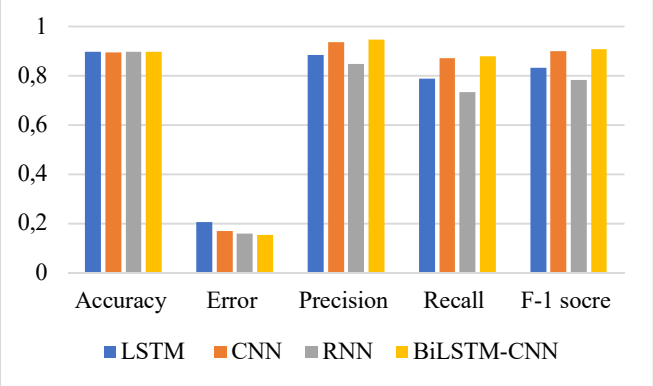

Fig. 6 Overall results

That being said, the combination between LSTM and CNN has surpassed all throughout all the experiments and all evaluation metrics. It has outstandingly proven to be a fine combination given the fact that it achieved an accuracy of 0.8979 with a precision of almost 95%. Fig.s 7 and 8 show the

confusion matrix for the combined model BiLSTM-CNN and RNN.

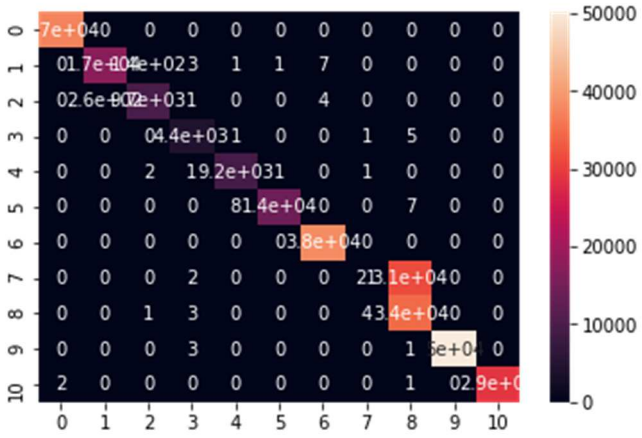

Fig. 7 BiLSTM-CNN confusion matrix

The confusion matrix above in Fig. 7 and 8 show the overall performance of the RNN and BiLSTM-CNN classifiers. The colored blocks indicate the number of classes used for this experiment which are eleven, where each square shows the correlation between the variables on each axis.

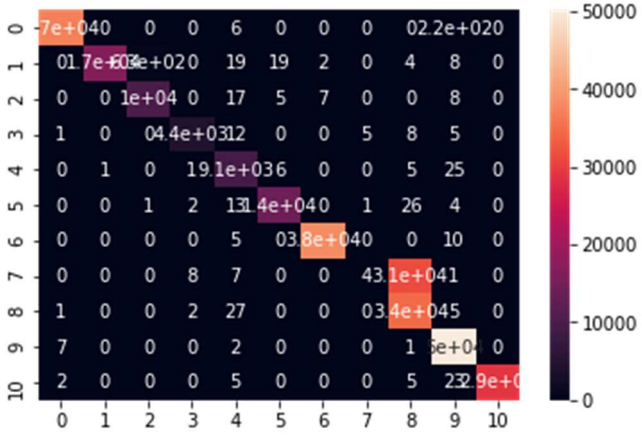

Fig. 8 RNN confusion matrix

IV. Conclusion

As a result of the attackers' use of spoofing methods, DDoS assaults pose a serious danger to everyone utilizing the Internet. Based on DDoS history, we observe that "Mirai" and "Bashlite" will not be the last and most powerful botnets as they evolved and bypassed the old methods for mitigation like adding more bandwidth or doing traffic extraction. Cloud computing is still the future. IoT-based DDoS attacks are threatening it. As IoT devices are fundamentally insecure yet, some of them show high resistance, but the rest do not. So, we cannot just leave for the goodwill of the manufacturer. The deep learning approach we chose has proven very worthy, giving an average accuracy of 0.896975, which indicates that it is a solid way to deal with DDoS attacks no matter what way they use. As observed from fig 5, BiLSTM-CNN has proven to be a great combination acquiring the highest accuracy, recall, precision, and F-1 score, and the lowest error rate followed by RNN has achieved an accuracy of 0.8977, and the error rate amounted to 0.1576. LSTM follows it with an accuracy of 0.8971. However, LSTM has achieved the highest error rate compared to the other classifiers. Although CNN has achieved the lowest accuracy rate of 0.895, it did achieve the second-lowest error rate of 0.1685. The usage of such a huge dataset and the excellent outcomes that we have obtained is an obviously great demonstration of the capacity to eliminate DDoS threats using deep learning. With more work and more innovative methods to use AI, we believe that we could have a solid opportunity to halt DDoS attacks once and for all.

Acknowledgment

This paper is funded by the Research and Innovation, Universiti Kuala Lumpur. This paper is supported by the Center of Intelligent and Autonomous Systems (CIAS), Universiti Tun Hussein Onn Malaysia (UTHM).

References

[1] B. A. Khalaf, S. A. Mostafa, A. Mustapha, and N. Abdullah, "An Adaptive Model for Detection and Prevention of DDoS and Flash Crowd Flooding Attacks," *Int. Symp. Agents, Multi-Agent Syst. Robot. 2018, ISAMSR 2018*, no. march, pp. 1–6, 2018.

[2] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2017*, 2017, pp. 618–623.

[3] K. Alieyan, A. Almomani, R. Abdullah, B. Almutairi, and M. Alauthman, "Botnet and Internet of Things (IoTs)," no. February, pp. 304–316, 2019.

[4] R. Want, B. N. Schilit, and S. Jenson, "Enabling the internet of things," *Computer (Long. Beach. Calif).*, vol. 48, no. 1, pp. 28–35, 2015.

[5] P. Desai, A. Sheth, and P. Anantharam, "Semantic Gateway as a Service Architecture for IoT Interoperability," *Proc. - 2015 IEEE 3rd Int. Conf. Mob. Serv. MS 2015*, pp. 313–319, 2015.

[6] F. Dahlqvist, M. Patel, A. Rajko, and J. Shulman, "Growing opportunities in the Internet of Things (IoT)," *McKinsey & Company*, 2019.

[7] S. Hilton, " 'oracle,'" *[Online]. Available: https://dyn.com/blog/.*

[8] C. Seaman, (2016). Threat advisory: Mirai botnet. *Akamai Threat Advisory*.

[9] B. A. Khalaf, S. A. Mostafa, A. Mustapha, M. A. Mohammed, and W. M. Abduallah, "Comprehensive review of artificial intelligence and statistical approaches in distributed denial of service attack and defense methods," *IEEE Access*, vol. 7, pp. 51691–51713, 2019.

[10] M. Ejaz Ahmed and H. Kim, "DDoS attack mitigation in internet of things using software defined networking," *Proc. - 3rd IEEE Int. Conf. Big Data Comput. Serv. Appl. BigDataService 2017*, pp. 271–276, 2017.

[11] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning DDoS detection for consumer internet of things devices," in *Proceedings - 2018 IEEE Symposium on Security and Privacy Workshops, SPW 2018*, Aug. 2018, pp. 29–35.

[12] H. Griffioen and C. Doerr, "Examining Mirai's Battle over the Internet of Things," *Proc. ACM Conf. Comput. Commun. Secur.*, no. October 2020, pp. 743–755, 202.

[13] C. F. M. Maseer, Z. K., Yusof, R., Bahaman, N., Mostafa, S. A., & Foozy, "Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset," *IEEE Access*, vol. 9, pp. 22351–22370, 2021.

[14] N. A. and N. F. R. Doshi, "Machine Learning DDoS Detection for Consumer Internet of Things Devices," *IEEE*, 2018.

[15] Y. Meidan *et al.*, "N-BaIoT: Network-based Detection of IoT Botnet Attacks Using Deep Autoencoders," *IEEE Pervasive Comput.*, vol. 17, no. 3, pp. 12–22, May 2018.

[16] Z. Al-Othman, M. Alkasassbeh, and S. A.-H. Baddar, "A State-of-the-Art Review on IoT botnet Attack Detection," 2020.

[17] Y. Meidan *et al.*, "N-BaIoT: Network-based Detection of IoT Botnet Attacks Using Deep Autoencoders," May 2018.

[18] O. F. Rashid, Z. A. Othman, and S. Zainudin, "A novel DNA sequence approach for network intrusion detection system based on

cryptography encoding method." *International Journal on Advanced Science, Engineering and Information Technology*, *7*(1), 183-189, 2017.

[19] H. Suo, J. Wan, C. Zou, and J. Liu, "Security in the internet of things: A review," *Proc. - 2012 Int. Conf. Comput. Sci. Electron. Eng. ICCSEE 2012*, vol. 3, pp. 648–651, 2012.

[20] C. Kolias, G. Kambourakis, A. Stavrou, J. Voas, and I. Fellow, "DDoS in the IoT," *Computer (Long. Beach. Calif).*, vol. 50, no. 7, pp. 80–84, 201.

[21] Y. Jia, F. Zhong, A. Alrawais, B. Gong, and X. Cheng, "FlowGuard: An Intelligent Edge Defense Mechanism against IoT DDoS Attacks," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9552–9562, 2020.

[22] Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto, and K. Sakurai, "Machine learning-based IoT-botnet attack detection with sequential architecture," *Sensors (Switzerland)*, vol. 20, no. 16, pp. 1–15, 2020.

[23] S. Sriram, R. Vinayakumar, M. Alazab, and K. P. Soman, "Network flow based IoT botnet attack detection using deep learning," *IEEE INFOCOM 2020 - IEEE Conf. Comput. Commun. Work. INFOCOM WKSHPS 2020*, pp. 189–194, 2020.

[24] M. Ozcelik, N. Chalabianloo, and G. Gur, "Software-Defined Edge Defense Against IoT-Based DDoS," *IEEE CIT 2017 - 17th IEEE Int. Conf. Comput. Inf. Technol.*, pp. 308–313, 2017.

[25] D. H. Summerville, K. M. Zach, and Y. Chen, "Ultra-lightweight deep packet anomaly detection for Internet of Things devices," *2015 IEEE 34th Int. Perform. Comput. Commun. Conf. IPCCC 2015*, 2016.

[26] M. Zhang *et al.*, "Poseidon: Mitigating Volumetric DDoS Attacks with Programmable Switches," no. February, 2020.

[27] E. van der Velden, "Master Thesis," *arXiv*, no. October, 2018.

[28] A. Wani and S. Revathi, "DDoS Detection and Alleviation in IoT using SDN (SDIoT-DDoS-DA)," *J. Inst. Eng. Ser. B*, vol. 101, no. 2, pp. 117–128, 2020.

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

[30] R. Wirth and J. Hipp, "CRISP-DM: towards a standard process model for data mining. Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining, 29-39," no. 24959, 2000.

[31] M. F. Ab Aziz, S. A. Mostafa, C. F. M. Foozy, M. A. Mohammed, M. Elhoseny, & A. Abualkishik, (2021). Integrating Elman Recurrent Neural Network with Particle Swarm Optimization Algorithms for an Improved Hybrid Training of Multidisciplinary Datasets. *Expert Systems with Applications*, 115441.

[32] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D convolutional neural networks and applications: A survey," *Mech. Syst. Signal Process.*, vol. 151, p. 107398, 202.

[33] S. A. Kashinath, S. A. Mostafa, A. Mustapha, H. Mahdin, D. Lim, M. A. Mahmoud, ... and T. J. Yang, (2021). Review of Data Fusion Methods for Real-Time and Multi-Sensor Traffic Flow Analysis. *IEEE Access*.

[34] P. K. Bediako, "Long Short-Term Memory Recurrent Neural Network for detecting DDoS flooding attacks within TensorFlow Implementation framework," p. 31, 2017.

[35] A. Azzouni, and G. Pujolle, "A long short-term memory recurrent neural network framework for network traffic matrix prediction," *arXiv preprint arXiv:1705.05690*, 2017.

[36] X. Liang, and T. Znati, "A long short-term memory enabled framework for DDoS detection," In *2019 IEEE Global Communications Conference (GLOBECOM)* (pp. 1-6). *IEEE*, December 2019.

[37] L. Shang, W. Zhao, J. Zhang, Q. Fu, Q., Zhao, and Y. Yang, "Network Security Situation Prediction Based on Long Short-Term Memory Network," In *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)* (pp. 1-4). IEEE, September 2019.

[38] B. A. Azizan, A. H., Mostafa, S. A., Mustapha, A., Foozy, C. F. M., Abd Wahab, M. H., Mohammed, M. A., & Khalaf, "A Machine Learning Approach for Improving the Performance of Network Intrusion Detection Systems," *Ann. Emerg. Technol. Comput.*, vol. 5, no. 5, 2021.

[39] A. M. Kadhum, and M. K. Hasan, "Assessing the determinants of cloud computing services for utilizing health information systems: A case study. *International Journal on Advanced Science, Engineering and Information Technology*, *7*(2), 503-510, 2017.