2023 IEEE International Conference on Edge Computing and Communications (EDGE) | 979-8-3503-0483-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/EDGE60047.2023.00062

DDoS-FOCUS: A Distributed DoS Attacks Mitigation using Deep Learning Approach for a Secure IoT Network

Mohammed Al-khafajiy
School of Computer Science
University of Lincoln
Lincoln, UK
malkhafajiy@lincoln.ac.uk

Ghaith Al-Tameemi
Faculty of Art, Science, and Technologies
University of Northampton
Northampton, UK
ghaith.al-tameemi@northampton.ac.uk

Thar Baker
School of Architecture, Technology and Engineering
University of Brighton
Brighton, UK
t.shamsa@brighton.ac.uk

***Abstract*—The fast growth of the Internet of Things devices and communication protocols poses equal opportunities for lifestyle-boosting services and pools for cyber attacks. Usually, IoT network attackers gain access to a large number of IoT (e.g., things and fog nodes) by exploiting their vulnerabilities to set up attack armies, then attacking other devices/nodes in the IoT network. The Distributed Denial of Service (DDoS) flooding-attacks are prominent attacks on IoT. DDoS concerns security professionals due to its nature in forming sophisticated attacks that can be bandwidth-busting. DDoS can cause unplanned IoT-services outages, hence requiring prompt and efficient DDoS mitigation. In this paper, we propose a DDoS-FOCUS; a solution to mitigate DDoS attacks on fog nodes. The solution encompasses a machine learning model implanted at fog nodes to detect DDoS attackers. A hybrid deep learning model was developed using Conventional Neural Network and Bidirectional LSTM (CNN-BiLSTM) to mitigate future DDoS attacks. A preliminary test of the proposed model produced an accuracy of 99.8% in detecting DDoS attacks.**

***Index Terms*—DDoS, IoT, CNN-BiLSTM, Distributed fog/edge**

I. Introduction

The cyber-physical attacks are security breach in cyberspace that impacts the computing/network medium. A malicious attack targets networked nodes and takes control over the computing and/or communication components. The attacks can have different forms and may cause serious damage; one form of attack can be forging node identity and fabricating falsified data to delude other legitimate nodes in the network and possibly terminate desired IoT applications/services [1]. Distributed Denial of Service (DDoS) is another form of attack where the attacker enlists the help of several interconnected nodes to each breed a small number of services-requests that, when added together, overload the target victim node(s). These interconnected nodes may either be willing accomplices (such as attacks initiated by Hacktivists groups) or unwitting victims nodes whose have been infected with malware. These nodes burden/encumbrance the network by unduly consuming network bandwidth, computational power and storage by operating fake services and fabricating massive amounts of falsified data.

With the emergence of the Internet of Things (IoT) and the fast development of the Internet protocols and the 5G, a large number of nodes/devices/things (*will stick to nodes*) attached to the Internet. Given the nodes limited maturity and capabilities, most of them lack the support of classical security schemes, such as Public Key Infrastructure (PKI), as a minimum security requirement [2], makes these nodes vulnerable to various attacks, one of which is the DDoS attacks. The consequences of developing these insecure IoT nodes can lead to insecure networks and the Internet as a whole as they can easily be compromised to become bad bots and exploited by attackers to launch malicious activities. According to Imperva 2022 report [3] on bad bot traffic, the bad bot traffic accounted for a record-setting 27.7% of all global traffic in 2021, up from 25.6% in 2020. Combined with good bot traffic, 42.3% of Internet traffic this past year was not human, compared to 40.8% in 2020. Human traffic decreased by 2.5% to 57.7% of all traffic [3]. This rapid increase in bad-bot-nodes traffic and the rising number of compromised-nodes has conspicuously raised the size of malicious-nodes and their traffic over the Internet, thus threatening the digital infrastructure massively [2], [4].

In this research, we utilise IoT fog nodes and their essential role and position in the IoT network to mitigate DDoS attacks. In brief, a fog computing network can be described as a network paradigm that utilises edge nodes to handle essential computation, communication and storage locally at the network edge and routes over the backbone of the Internet. The services fog nodes provide are similar to the cloud but at a smaller scale and proximity. Fog nodes are positioned between things and clouds at the network edge “closer” to where the data is generated. Fog nodes do not substitute the cloud but complement its features, especially for time-sensitive services/applications where processing and response must be ultra-fast. The main contribution of this research is the utilization of fog nodes to host a hybrid deep learning model, specifically a CNN-BiLSTM, for mitigating DDoS attacks. The CNN-BiLSTM model combines a Convolutional Neural Network (CNN) with a Bidirectional Long-Short-Term

2767-9918/23/$31.00 ©2023 IEEE
DOI 10.1109/EDGE60047.2023.00062

Memory (BiLSTM) to effectively detect spatial inputs and capture temporal features, enabling accurate prediction of DDoS attacks in IoT environments. By leveraging fog nodes, this approach enhances the overall efficiency and effectiveness of DDoS mitigation/detection in IoT networks.

The rest of the text is sectioned as follows: Background on the security threats/attacks on IoT networks discussed in Section II. The proposed DDoS-FOCUS framework is presented in Section III. The results and evaluation are reported in Section IV. Section VI concludes the paper.

II. Background

The security threats/attacks on IoT networks are highlighted in this section. It is worth mentioning that the IoT architecture adopted is inspired by our previous research [1], [5].

A. IoT network architecture with fog nodes

The decentralised IoT network architecture which compares fog nodes is comparable to other distributed computing networks, such as cloudlet and mist paradigms. Recall from [1], [5] that fog nodes are distributed over the network edge where data are generated, and the fog nodes can be clustered, forming a fog domain. In general, the IoT network architectures are either application-specific or application-agnostic. This research adopts IoT application agnosticism architecture (presented in Figure 1) that is inline with [1], [5]–[10]. It is essential to understand the IoT-fog-based architecture to get insight into how DDoS attacks can be formed and impact the fog nodes. The IoT topology adopted encompasses the $things$, $fogs$, and $cloud$ layers as per Figure 1.

Things Layer: what so-called $perception$ layer, is where data is generated. It contains various devices and equipment in the form of things (e.g., ambient sensors) that are equipped (for example, communication protocols like MQTT and Zigbee) to transmit data over the IoT network. The heterogeneity of these nodes and their support of different networking protocols increase the IoT network's vulnerabilities.

Fog Layer: it is formed from distributed nodes that span over the edge of the network. The fog nodes are geo-distributed to support large IoT networks' scalability, extensibility and availability. Fog nodes are meant to handle the primary data processing from the things layer. Due to their low capabilities (compared to clouds), they can be at more risk from cyber-attacks, specifically DDoS, to disturb their functionality and employ them for the attacker's benefit.

Cloud Layer: also known as $data-centres$ layer enabling convenient network infrastructure settings to access shared resources (e.g., computation and storage) via the Internet. The cloud is ideal to perform "large-services" of data mining that fog nodes cannot execute (e.g., big query with Hadoop/Spark).

B. Threats on IoT network

This section sheds light on how the malicious IoT nodes (things or fogs nodes) may use a variety of techniques/attacks to interfere with network functioning.

1) Forgery: IoT nodes that are malicious may have fraudulent data and false identities in order to deceive other nodes in the network and possibly terminating the desired IoT service. These kinds of nodes overkill the network's resources by creating a lot of bogus data, burdening the network links by consuming high bandwidth and unduly consuming the node's computational power and storage.
2) Tamper: packet tampering can be done by malicious nodes in the network. This type of attacks can be formed by manipulating packets transmitted over the network or even dropping or delaying transmitted packets. It is challenging to identify such rogue nodes since transmission failure or delay may result from various causes, including unstable network-channels conditions or a poor network signal, in addition to tampered IoT nodes.
3) Impersonation: in order to provide an IoT service, malicious/impersonation nodes will pose as honest/genuine nodes and then provide phishing data or services to compromise network security and end users' privacy.
4) Distributed Denial of Service (DDoS): DDoS attacks are hostile attempts to stop legitimate users/nodes from using IoT services by disrupting them. These attacks are created by sending a large number of unnecessary or redundant service requests to the target IoT nodes in order to stop genuine users/nodes from receiving services, and also this assault uses up network resources. DDoS attacks utilize multiple IoT devices to form attacks on a specifically targeted node. The infected IoT nodes are controlled remotely by malware injected by a DDoS attacker. These individual nodes can be referred as bots, and a group of bots is called a botnet. Once a botnet has been created, the attacker can control an attack by giving each bot remote commands. Hence, the victim's IoT nodes that are targeted by these botnet receive a requests sent from botnet's bot, potentially causing the victim's IoT node or full network to become overwhelmed that can results in getting the network down.

III. Proposed DDoS-FOCUS

The proposed DDoS-FOCUS consists of a network data pre-processing deep learning model for the DDoS classification task. Data are analyzed and understood during pre-processing step by performing exploratory data analysis. Next, data cleaning is deployed to improve the quality of the data. Then, feature scaling and label encoding are employed. Finally, we deploy a hybrid deep learning model to predict future network attacks.

A. Dataset

The dataset used to develop and assess the proposed DDoS-FOCUS framework and train the CNN-BiLSTM deep learning model is the CSE-CIC-IDS201 dataset, it is publicly available[1]. It has been generated from three well-known

[1] DDoS balanced and unbalanced datasets can be accessed at https://www.kaggle.com/datasets/devendra416/ddos-datasets

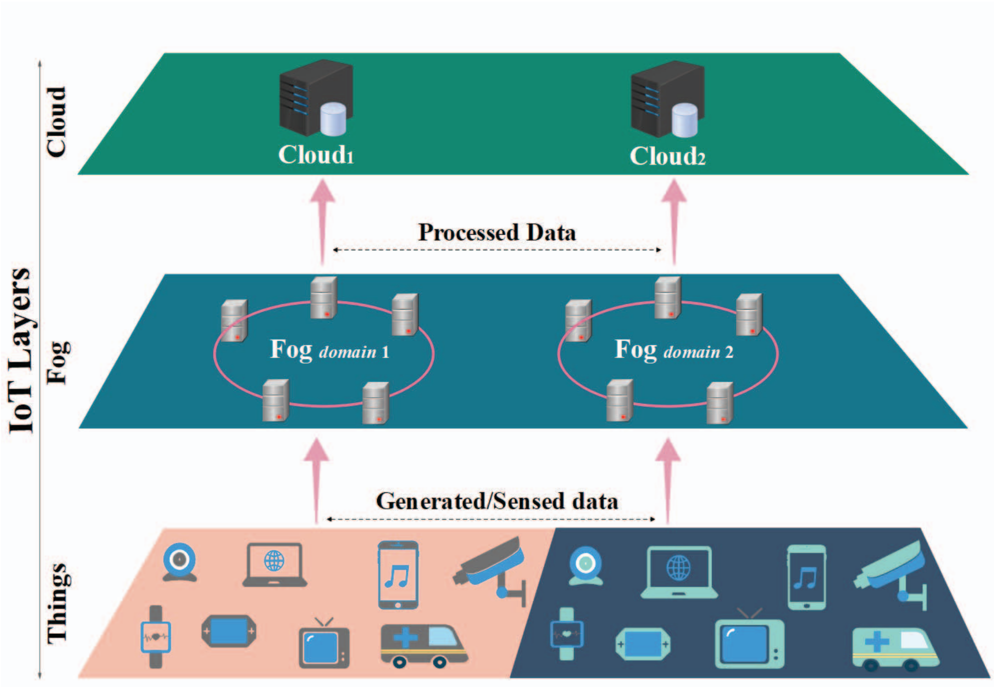


Fig. 1. IoT network layers

datasets (CIC-IDS2017, CSE-CIC-IDS2018, CIC DoS) from CIC Canda [11]. The dataset contains DoS and DDoS traffic flows labelled as ”DoS” combined with ”Benign” flows which both are extracted separately from the same base datasets and made into a single large dataset. It is worth mentioning that the network topology in which the data is captured comprises 50 attacking machines (i.e., malicious nodes), and the victim network has 5 divisions comprising 420 devices and 30 servers. The data captured from the network topology includes traffic packets and system logs of each machine; having a total of 84 network traffic attributes that were extracted from over 7.5 million flows captured over three different years (2016, 2017 and 2018). By extracting transverse values from the datasets used to detect and characterise DDoS attacks, the dataset became useful for training the model.

B. Data Pre-processing

1) Due to the large amount of socket information contained in the dataset in CSV file format, we condensed it for simpler training purposes. The ordinal encoding method was used to convert non-numeric elements to numeric data in order to conform to the framework’s numeric composition.
2) To ensure a randomized sample, records were omitted from the dataset at random times during import. In addition, we removed the rows with NaN values as part of data preparation.
3) BEGNIGN classes were marked 0 while DDoS classes were marked 1. We also scale our data to avoid undue impact on training due to the dataset’s quantities.

C. Deep Learning Models

Deep learning is gaining popularity among researchers in various tasks aimed at detecting computer network attacks and anomalies. Deep learning can identify patterns in complex data; therefore, it is useful to solve many tasks such as image segmentation, pattern recognition, time series classification, etc. In deep learning methods, there is no need for manual feature engineering, which is a significant benefit. Therefore, these methods do not require feature selection before training as they can detect patterns among massive datasets automatically. Weight matrices are used, for example, to emphasize the features that have the greatest impact on classification. It is possible to extract elaborate attack patterns from IP flow protocol datasets, which are sometimes less visible to the human eye, which improves classification accuracy.

This paper proposes a hybrid deep learning model of CNN-BiLSTM for DDoS attack mitigation. In addition to our proposed model, other deep learning techniques such as GRU, CNN, and BiLSTM are fully described below for conducting a comparative analysis with our proposed model:

1) Gated Recurrent Network (GRU): A GRU works by updating and resetting its gates. As part of the first gate, the information regarding a new entry is defined and what new information will be added is defined as well. As opposed to the first gate, the second one describes how much information will be lost in the long run. The hidden state h_t in GRU is calculated as shown in equations (1) to (4):

$$z_t = \sigma(x_t \eta^z + h_t - 1W^z) \tag{1}$$

$$r_t = \sigma(x_t \eta^r + h_t - 1W^r) \tag{2}$$

$$\tilde{h_t} = tanh(x_t \eta^h + (r_t * h_t - 1)W^h) \tag{3}$$

$$h_t = (1 - z_t) * h_t - 1 + z_t * \tilde{h_t} \tag{4}$$

Where η^z, W^z, η^r, W^t, η^h, and W^h refer to the weights associated with the GRU. z_t denotes the update gate, r_t denotes the reset-gate, $\tilde{h_t}$ represents a candidate hidden-state, and σ is the component wise logistic sigmoid function.

2) Proposed CNN-BiLSTM: a convolutional neural network in conjunction with a $bidirectional$ long-short-term-memory (CNN-BiLSTM) was used for detecting the future DDoS attack. Using our proposed model, we detect and categorize traffic into two groups: normal traffic, called ”Bengin”, and abnormal traffic, called ”DDoS”. The proposed approach's structure is shown in Figure 2. A combined model is usually more effective if both networks are combined, according to literature in [12]. This approach is particularly useful because it has the ability to detect both spatial and temporal information. The model architecture is described in the following subsections.

a) 1D CNN: CNN became widely used due to its ability to automatically detect contaminants within objects. CNNs are composed of a set of layers. Each layer in the network performs a specific functionality. The outputs of several filters are extracted simultaneously and represented as activations by convolutions. A feature vector can be created by applying multiple convolutions to the input.

In order to improve the performance of feature extraction, this network utilizes three convolutional layers connected by another layer. Initially, the network has a 1D input-layer accepting inputs dimensioned with [4874564,1]. First layer is *Conv1D* with a kernel size of 1 and a total of 64 filters. In the obtained feature map, a batch normalization layer (*Batch_N*) and a rectified linear unit layer (*ReLU*) were added after the convolution, respectively, to normalize the inputs across filters. Max-pooling (*MaxPooling1D*) layer also included for dimensionality reduction for the feature vectors. Dropout layer was used to mitigate the overfitting issue. These layers (*Conv1D, Batch_N, ReLU, MaxPooling1D, Dropout*) were double replicated to extract deeper/intense features from input using filter sizes 32 and 16.

b) BiLSTM: the LSTM formed of various number of gates; which are *(i)* to represent the input, *(o)* is the output, and *(f)* for forget gates. The input and output gate acts as a memory block for a main functional cell, storing input events and calculating the data to be stored in the network's memory. The cell is additionally connected to its associated gate through a peephole connection to facilitate feedback. At any time *t*, the primary cell output *Z* is specified as,

$$Z_t = f_t Z_{t-1} + i_t z_t \tag{5}$$

where f_t denotes the forget gate activation, i_t for the input gate activation, and z_t the input to the main cell. In this network, hidden units are activated by using a sigmoid function σ provided by Equation (6).

$$h_t = o_t tanh(c_t) \tag{6}$$

o_t denotes the output gate activation. Additionally, each gate can be declared using the following equations.

$$i_t = \sigma(W_{xi}X_t + W_{hi}h_{t-1} + W_{zi}z_{t-1} + b_i) \tag{7}$$

$$f_t = \sigma(W_{xf}X_t + W_{hf}h_{t-1} + W_{zf}z_{t-1} + b_f) \tag{8}$$

$$o_t = \sigma(W_{xo}X_t + W_{ho}h_{t-1} + W_{zo}z_{t-1} + b_o) \tag{9}$$

$$z_t = tanh(W_{xz}X_t + W_{hz}h_{t-1} + b_z) \tag{10}$$

W_{x*} denotes the input-to-gate weight, W_{h*} denotes the hidden-to-hidden weight, and W_{z*} denotes the peephole weights. While the training task is being performed, BiLSTM can process data both forward and backwards and can be defined as follow:

$$y_t = W \underset{hy}{\rightarrow} h^{\overrightarrow{N}} + W \underset{hy}{\leftarrow} h^{\overleftarrow{N}} + by \tag{11}$$

where $h^{\overrightarrow{N}}$ and $h^{\overleftarrow{N}}$ are the hidden layers in the forward and backward directions, respectively. Total of 50 units are selected through the BiLSTM, resulting in a total of 100 units (50 backward and 50 forward).

IV. Training and Classification Performance

We train our data using the proposed CNN-BiLSTM model and each network individually (CNN and BiLSTM). We also compared our proposed model with another deep learning technique which is GRU. An analysis of the ability of deep learning to identify DDoS attacks was conducted using five scenarios. The training was carried out using the adaptive moment estimation (Adam) algorithm. The reason for choosing Adam is its suitability for data with noisy and sparse gradients.

The original dataset was divided into three subsets using train_test_split. We first ordered our instances based on the timestamp and assigned the first 80% of this data for training and the last 20% for testing. As a result, our model can effectively detect upcoming attacks over time. Table I summarises the proposed model properties and experiment environment.

Our proposed model was evaluated using Equations (12) to (15). A number of performance metrics resulting from our study were considered, including: accuracy, sensitivity (also known as recall), precision, and F1-score. All of these performance metrics are represented using these formulae:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

$$sensitivity = \frac{TP}{TP + FN} \tag{13}$$

$$precision = \frac{TP}{TP + FP} \tag{14}$$

$$F1 - score = \frac{2TP}{2TP + FP + FN} \tag{15}$$

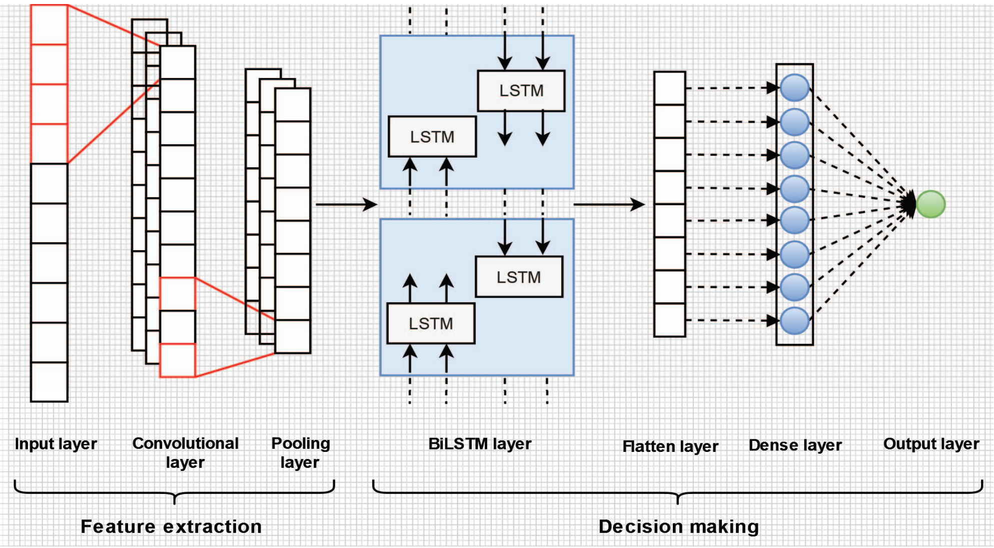


Fig. 2. The proposed CNN-BiLSTM architecture

TABLE I
DETAILS OF CNN-BILSTM

Parameter	Value
Input dimensions	76 features
Filter size	64, 32, 16
Kernel size	1
Input activation	ReLU
Dropout layer	50%
BiLSTM layer units	50
Dense layer units	100
Output dimension	1
Output activation	sigmoid
Error function	Binary cross-entropy
Training split	80%
Testing split	20%
Validation split	20% of the training size
Learning optimizer	adam
Epochs	50
Batch size	256

It is obvious that TP denotes to true-positive, whereas TN indicates true-negative, the false-positive is FP, and false-negative is FN.

V. RESULTS

A. Classification results

In Figure 3, we provide an illustrative example of the comparison between the training loss and accuracy of our CNN-BiLSTM model and standalone CNN and BiLSTM models. The results demonstrate that the CNN-BiLSTM model outperforms the other algorithms without experiencing overfitting. We evaluate the performance of our proposed CNN-BiLSTM model by classifying unseen traffic flows and predicting whether or not they are DDoS attacks. Table II reveals the obtained accuracy, precision, recall, and F1 on the test dataset using GRU, BiLSTM, CNN, and CNN-BiLSTM. It was observed that the GRU network had the lowest accuracy, with a score of 0.97. In BiLSTM network, performance improved by approximately two percent, and accuracy increased to 0.993. The best performance was achieved using proposed hybrid approach of CNN-BiLSTM, with an average accuracy of 0.998.

CNN-BiLSTM model was provided the highest F1-score in predicting the DDoS class with a score of 0.99, while GRU provided the lowest with 0.93 only. Its clear from the table II that all the models have performed well in predicting Benign class since the values of F1-score for CNN-BiLSTM, BiLSTM, CNN and were 1, 1, and 0.99, respectively. In comparison with the other classifiers, CNN-BiLSTM scored the highest in both precision and recall with a value of 0.998 and 0.994, respectively. The CNN classifier, on the other hand, provided the lowest precision and recall with scores of 0.992 and 0.964 respectively.

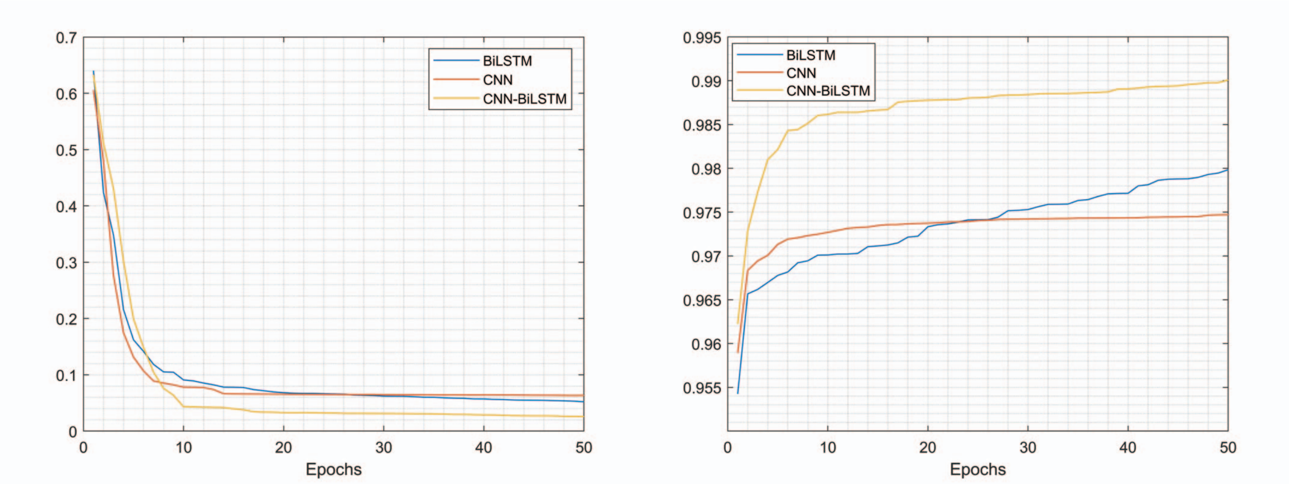
It has been shown that the proposed CNN-BiLSTM model outperforms the classical deep learning models. In the case of BiLSTM, due to its ineffectiveness in extracting features, BiLSTM is ranked low because it stores contextual information for both forward and backward directions. This research focuses on consolidating a BiLSTM with a CNN model. BiLSTM models can efficiently store two-directional context data forward (next) and backward (previous). CNN creates a more accurate representation of the data by combining information from the current and previous inputs. By maintaining the current context as well as the previous context, the BiLSTM model is able to predict court decisions more effectively than the CNN model. A high classification result is achieved because the input data is better represented.

B. State-of-the-Art comparison

Our proposed DDoS-FOCUS with CNN-BiLSTM will be compared to the state-of-the-art solutions to mitigate DDoS

TABLE II
PERFORMANCE OF PROPOSED CNN-BILSTM COMPARING TO THE OTHER DEEP LEARNING TECHNIQUES

Model	Class	Precision	Recall	F1-score	Accuracy
BiLSTM	DDoS	0.98	0.98	0.98	
	Benign	1.00	1.00	1.00	
	Averaged	0.987	0.989	0.988	0.993
CNN	DDoS	1.00	0.93	0.96	
	Benign	0.98	1.00	0.99	
	Averaged	0.992	0.964	0.977	0.98
GRU	DDoS	1.00	0.86	0.93	
	Benign	0.97	1.00	0.98	
	Averaged	0.98	0.93	0.95	0.97
CNN-BiLSTM	DDoS	1.00	0.99	0.99	
	Benign	1.00	1.00	1.00	
	Averaged	**0.998**	**0.994**	**0.996**	**0.998**



(a) Loss

(b) Accuracy

Fig. 3. Accuracy and Loss results per epoch for our proposed *CNN-BiLSTM* model

attacks using UNB datasets for DDoS attack detection.

The work in [13] was particularly interesting to us since it is similar to our approach for detecting DDoS attacks. The model was trained and tested using a hybrid approach of CNN and RNN on ISCX2012 dataset. They achieved an accuracy value of 99.71%. An ensemble solution to detect network intrusions can be created by combining Multi-Objective Genetic Algorithms (MOGAs) and Neural Networks (NNs), as presented by the researchers in [14]. The proposed approach demonstrated an overall detection accuracy of 97%.

Similarly, the authors in [15] proposed a hybrid approach of Ensemble of Feature Selection (EFS) and Adaptive Grasshopper Optimization Algorithm (AGOA) to identify DDoS attacks. This approach starts by ranking each attribute using the EFS method, which is the first step in selecting a subset of attributes that are highly ranked. Following the reduction of datasets, AGOA is applied to determine characteristics that can be used to predict traffic behavior in networks. This approach provided an overall accuracy of 99.13%.

The work in [16] presented four different deep learning models for the detection of DDoS attacks on Internet of Things (IoT) networks. Models are created by combining LSTMs, CNNs, and fully connected layers. Models input layers consist of 82 units, one for each flow level in CIC2017, and output layers provide probability information for a given flow to be part of a DDoS attack. A good classification score can be achieved with 1D-CNN+LSTM, but false negative rates seem to be higher in the other models.

The authors in [17] developed an algorithm called DeepGFL that aims to extract high-order features from low-order features using a hierarchical graph representation. A Decision Tree and Random Forest classifier were trained using the graph representation of the features to validate the proposed framework and tested them at CIC2017. It is evident from the presented results that our solution is susceptible to FN, which results in very low F1 scores even though the precision scores on many types of attacks are fairly good.

VI. Conclusion and future work

This research responded to the rapid increase of malicious traffic over the Internet due to the significant growth of Bad bot traffic and the growing number of vulnerable IoT nodes. The proposed model utilises IoT fog nodes and their essential role and position in the IoT network to mitigate DDoS attacks. A CNN-BiLSTM hybrid deep learning model has been proposed. The CNN-BiLSTM is a convolutional neural network in conjunction with a Bidirectional long-short-term memory adopt to detect the spatial information and temporal features to predict DDoS attacks. The achieved results demonstrate that the proposed CNN-BiLSTM model outperforms the classical deep learning models and the state-of-the-art results. In future work, the fog nodes will be trained to not only detect DDoS attacks but block them and notify other vulnerable nodes in the IoT network.

References

[1] M. Al-Khafajiy, T. Baker, M. Asim, Z. Guo, R. Ranjan, A. Longo, D. Puthal, and M. Taylor, "Comitment: A fog computing trust management approach," *Journal of Parallel and Distributed Computing*, vol. 137, pp. 1–16, 2020.

[2] X. Chen, L. Xiao, W. Feng, N. Ge, and X. Wang, "Ddos defense for iot: A stackelberg game model enabled collaborative framework," *IEEE Internet of Things Journal*, 2021.

[3] Imperva, "2022 imperva bad bot report — evasive bots drive online fraud," 2022. Available at https://www.imperva.com/resources/resource-library/reports/bad-bot-report.

[4] H. Yahyaoui, Z. Maamar, M. Alkhafajiy, and H. Al-Hamadi, "Trust-based management in iot federations," *Future Generation Computer Systems*, vol. 136, pp. 182–192, 2022.

[5] M. Al-khafajiy, T. Baker, H. Al-Libawy, Z. Maamar, M. Aloqaily, and Y. Jararweh, "Improving fog computing performance via fog-2-fog collaboration," *Future Generation Computer Systems*, vol. 100, pp. 266–280, 2019.

[6] A. Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue, "On reducing iot service delay via fog offloading," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 998–1010, 2018.

[7] Q. Fan and N. Ansari, *Towards Workload Balancing in Fog Computing Empowered IoT*. IEEE Transactions on Network Science and Engineering, 2018.

[8] W.-S. Kim and S.-H. Chung, "User-participatory fog computing architecture and its management schemes for improving feasibility," *IEEE Access*, vol. 6, pp. 20262–20278, 2018.

[9] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1171–1181, 2016.

[10] M. Al-khafajiy, L. Webster, T. Baker, and A. Waraich, "Towards fog driven iot healthcare: challenges and framework of fog computing in healthcare," in *In Proceedings of the 2nd International Conference on Future Networks and Distributed Systems*, p. 9, ACM, Jun 26 2018.

[11] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization.," *ICISSp*, vol. 1, pp. 108–116, 2018.

[12] Z. Zheng, Z. Chen, F. Hu, J. Zhu, Q. Tang, and Y. Liang, "An automatic diagnosis of arrhythmias using a combination of cnn and lstm technology," *Electronics*, vol. 9, no. 1, p. 121, 2020.

[13] G. Wei and Z. Wang, "Adoption and realization of deep learning in network traffic anomaly detection device design," *Soft Computing*, vol. 25, no. 2, pp. 1147–1158, 2021.

[14] G. Kumar, "An improved ensemble approach for effective intrusion detection," *The Journal of Supercomputing*, vol. 76, no. 1, pp. 275–291, 2020.

[15] S. Dwivedi, M. Vardhan, S. Tripathi, and A. K. Shukla, "Implementation of adaptive scheme in evolutionary technique for anomaly-based intrusion detection," *Evolutionary Intelligence*, vol. 13, no. 1, pp. 103–117, 2020.

[16] M. Roopak, G. Y. Tian, and J. Chambers, "Deep learning models for cyber security in iot networks," in *2019 IEEE 9th annual computing and communication workshop and conference (CCWC)*, pp. 0452–0457, IEEE, 2019.

[17] Y. Yao, L. Su, and Z. Lu, "Deepgfl: Deep feature learning via graph for attack detection on flow-based network traffic," in *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*, pp. 579–584, IEEE, 2018.