Flow-Based DDoS Detection Using Deep Neural Network with Radial Basis Function Neural Network

Ting-Chung Leung* and Chung-Nan Lee†
Department of Computer Science and Engineering
National Sun Yat-Sen University, Kaohsiung, Taiwan
*E-mail: fatcowlolz@gmail.com Tel: +886-7-5252000 ext. 4335
†E-mail: cnlee@mail.cse.nsysu.edu.tw Tel: +886-7-5252000 ext. 4335

Abstract— **This paper presents a Distributed Denial-of-Services (DDoS) attack detection method using Deep Neural Network (DNN) based on flow-based network information. In the proposed method, Radial Basis Function is adopted for feature selection to enhance the accuracy of the DNN. The flow-based dataset CICIDS2017 is used for training and testing the detection accuracy. Experimental results show that the DDoS detection accuracy of the proposed method under the dataset are 99.37%.**

Keywords—Distributed Denial-of-Service, Deep Neural Network, Radial Basis Function Neural Network, DDoS Classification

I. INTRODUCTION

As information technology has been evolving to become faster and more powerful, the attack from hackers also becomes more sophisticated and vicious. The traditional DoS attack principle is that the hacker uses a strong single computer to attack the target, and this is a point-to-point attack. A distributed denial-of-service (DDoS) attack is evolved from the traditional Denial of Service. It paralyzes the service of a network using the volume traffics from many different sources to flood the victim. DDoS attacks are implemented in different layers of network by using different protocols, such as UDP and TCP, etc. Hence, how to prevent DDoS attacks plays a key role to network security.

Since AlphaGo wins the most professional chess player in the world, as well as the highly sophisticated A.I. for any non-player controlled characters has greatly enhanced the overall player experience, the deep learning now becomes a household name. Some of the biggest technology companies like Google and Microsoft have applied the deep learning to many products with wonderful results. Deep learning technology is now an indispensable part of human life, in spite of people do not know it, such as image and speech recognition, natural language processing, biomedical information, etc.

The role of Intrusion Detection System (IDS) is of paramount importance to protect organizations from cyber-attacks. IDS is a system for monitoring network traffic and issuing alerts when suspicious activity is detected. Due to the signature syntax and rule of these open-source tools are different, unless the same set of rules used together between tools. Though the capabilities of every open-source IDS tool are different, it is able to detect different attacks.

Once the hacker uses multiple compromised computers that may be called “zombies”, to launch a DDoS attack. The hacker invades many victim hosts, install the DDoS attack program into the host and control them to launch the attack, causing the target unable to keep connection or even system crash. Many people often become DDoS attack accomplices without knowing it, because hackers attack a large number of computers in an indirect way, and anyone’s host may be one of them. From the report of McAfee Labs in 2016 – 2017, attack such as DoS and scan attacks is up to 35% in the top list of cyber-attacks. Therefore, in this paper, we focus on DDoS or DoS attack detection.

Since the characteristic of reflection-based DDoS are the volume of traffic, such as bytes, packets or flows. It is easy to detect through volume of traffic. Instead for the exploitation-based DDoS attacks, the traffic flow is very low compared to the volumetric attack, but it led to a bigger threat to controller compared to other attacks. Such as TCP/SYN attack will lead to great impact to target machine, especially to cloud servers. As once the cloud servers’ crashes or malfunctions, it is not able to resume the communication and services, the worst case is that data lost when a server is crashed. Therefore, it is important to develop an-approach to detect the exploitation-based DDoS attacks.

Nowadays, deep learning can be used to solve many problems, and cybersecurity is one of them, from classifying attacks to anomaly behavior. In this paper, TensorFlow is used, it is an open source library that performs deep learning introduced by Google. On the other hand, DNN is used in this paper. DNN is considered to have low complexity and lower accuracy than GRU. However, by increasing the complexity of DNN, it can also perform high precision like GRU. On the other hand, we use RBFNN for feature selection. Since there would be some useless features in the dataset, feature selection may filter out those useless features, as a result it will lead to increase the accuracy of deep learning, as well as the network is much simpler than not using feature selection.

The contributions of this paper are mainly in the following:

- We design and build up a DNN based method to increase the accuracy of classifying whether the network flow is a DDoS flow or not.
- We verify whether using Radial Basis Function Neural Network (RBFNN) as the feature selection

approach can improve all deep learning model's performance in terms of accuracy.

- The proposed method performs better than other neural network like GRU.

The remainder of this paper is as follows. Section 2 introduces the exploitation-based DDoS attacks and then discusses some related literatures, introduces the existing technology and research background related to this paper. The third section presents the proposed method. The fourth section conducts some experiments and presents their results. Finally, we draw conclusions in the last section.

II. Related Work

A. *Taxonomy of DDoS Attack*

DoS can be mainly classified as two types, reflection-based DDoS attacks and exploitation-based DDoS attacks. UDP flood attack is a kind of typical reflection-based DDoS attack which sends large number of UDP packets to the target machine. These UDP packets are sent to random ports of the target machine with high frequency. Then the server starts dropping flows. But the switches and the server may be in connection or disconnection sometimes. When the attack is stopped, the communication is resumed to be normal. This type of attack is not serious for the controller.

Exploitation-based DDoS attack such as TCP/SYN flood consumes the resources by exploiting TCP-three-way handshake. Since the IP of the incoming connection are fake, no response will be made by the target machine for its SYN/ACK packet. The port of the connection must be kept open unnecessarily. With many spoofed SYN requests, all the ports of victim machine are kept open unnecessarily and blocked, and the connection with the legitimate users is disconnected. This type of attack only needs small amount of traffic to make target machine unavailable, attackers only need to hold the fake connections. This attack keeps sending SYN packets to the target until the server is malfunctioned. Although the traffic flow is very low but is leading to a much bigger threat to the server. The server also cannot handle such a great number of flows and will be crashed abruptly, and the server is unable to be recovered. Therefore, this type of attack can paralyze the server easily. No matter the attack is stopped, the communication of the controller will not be resumed. As the fake connections are still being hold, and host is repeating to try to finish the handshake.

Therefore, it can be seen that the exploitation-based DDoS attacks, such as TCP/SYN attack will lead to great impact to a target machine, especially to cloud servers. As once the cloud server crashes or malfunctions, it is not able to resume the communication and services, the worst case is that data are lost during the server is crashed.

B. *Literature Review of DDoS Detection and Attack Dlassification*

In the past, network administrator used to do flow on the switch or network devices directly. As time passing, software-defined network is now better developed, network administrator needs only to do flow control at the controller of the software-defined network, but both types of network need to address the concern of cyber-attack. Here we only focus on a type of cyber-attack, DDoS. There is much literature on classifying different types of DDoS in packet-based and classifying normal flows and DDoS flows in flow-based. But classifying different types of DDoS in flow-based is lack of discussion.

Chockwanich and Visoottiviseth [1] compared the accuracy between using Snort (one kind of intrusion detection systems) and deep learning-based detection systems on detecting DDoS attack. They presented the accuracy of Snort, RNN, stacked RNN, CNN are 0.4716, 0.9976, 0.9975, 0.9956, respectively. While the processing time of Snort is the shortest on detecting attacks than other deep learning models as the payload of pcap files is used as the testing set, due to Snort only scans the packet header. However, the accuracy of Snort is the lowest. Hsieh and Chan [2] proposed a neural network based DDoS detection method as well as implemented the system in an Apache Spark cluster. They used 2000 DARPA LLDOS 1.0 as the training dataset while perform experiment in real network environment. 2000 DARPA LLDOS 1.0 is used as dataset which is a packet-based information. Hou et al. [3] introduced a method to detect DDoS traffic using machine learning. They extracted flow-based features and pattern-based features from sampling data in real-time by using NetFlow. The results show that the average accuracy is more than 99% and a false-positive less than 0.5%.

Roopak et al. [6] proposed the hybrid CNN+LSTM framework which performs a better accuracy than other deep learning models and machine learning models. Flow-based dataset CICIDS2017 [21] is used, and the accuracy of detecting attack flows is 97.16%. Li et al. [9] introduced a DDoS detection method based on hybrid deep learning model, DCNN-DSAE. While using attack flow and normal flow for testing, the accuracy can reach 98.53%. They suggested that SDN is able to import this DDoS detection system since it is able to collect the flow features and create flow entry. Since the dataset is collected by the system in the experiment, features are selected before the flow entry is created, but no feature selection method is implemented. Assis et al. [10] proposed a defense system against intrusion and DDoS attacks based on SDN. They used the Gated Recurrent Units (GRU) method on the detection system. CICDDoS 2019 [21], a flow-based dataset, is used, and 83 features are used in the experiment to avoid data bias. The average result of accuracy, precision, recall and f-measure rates of GRU is 99.94%, which is higher than other methods, such as DNN, CNN, LSTM, Support Vector Machine (SVM), Logistic Regression (LR), k-Nearest Neighbors (kNN) and Gradient Descent (GD). There is no feature selection in their work.

Cil et al. [11] proposed a DDoS detection system based on DNN. A DNN model with 69 units in input layer, 3 hidden layers with 50 units for each and 2 units in output layer. CICDDoS2019, a flow-based dataset, is used. In this experiment, feature selection is processed manually, 8 features are deleted that do not contribute to the training, and 9 features are deleted due to their values are "0" meaningless, and 69 features are used finally. Two results are given, the first result detects attack flows, with an accuracy of 99.97%, while the second result is classifying the attack flows into reflection-based attack and exploitation-based attack, with an accuracy of 94.57%. Elsayed et al. [12] proposed DDoSNet that is a system of intrusion detection on DDoS in SDN. The system is based on deep learning, RNN, with autoencoder. CICDDoS2019 is used as the dataset for training and testing. While the data preprocessing, only 77 features is obtained as the unwanted features are removed manually, but no feature selection method is presented. The total number of samples for training, validation and testing are 161523, 46150 and 23000, respectively. The accuracy on detecting attack flows and normal flows is 99%.

In [4], Safe-Guard Scheme (SGS) is proposed for protecting control plane against DDoS attacks in SDN. A sequence-counter-based DDoS detection method is proposed in [5] for real-time DDoS detection. In [7], k-means++ clustering is used improve the learning of SVM to detect DDoS packets from packet information. In [8], Unal et al. discussed about the performance of deep learning is better than traditional machine learning algorithm on detecting DDoS attacks.

Zhou et al. [14] proposed a mathematical-based DDoS attacks detection using packet size interval. In [15], Girma et al. discussed about some impacts that DDoS attacks affect cloud computing, such as SYNC flood attack and HTTP flood attack. Mirkovic et al. [16] proposed some benchmarks on defending DDoS attack and discussed on some impacts that DDoS attacks, such as UDP, ICMP and SYN attacks. Belabed et al. [17] proposed smart routing to defend link-flooding attacks and discussed on the algorithm of attacker and defender on DDoS attacks. Yan et al. [18] discussed on DDoS attacks affect cloud computing, such as SDN, as well as taxonomy of DDoS attacks. Bhosale et al. [19] proposed mechanisms on classification of DDoS prevention and discussed on some DDoS attack tools.

C. Radial Basis Function Neural Network

RBFNN is an artificial neural network, which is based on radial basis function as activation function. A linear combination of radial basis functions is as the format of the network's output. The linear combination is based on the input and neuron parameters of the network. Since the ability of nonlinear fitting is strong, it can perform well on feature selection, and referenced by Mak [13]. The network structure of RBFNN is shown in Figure 1.

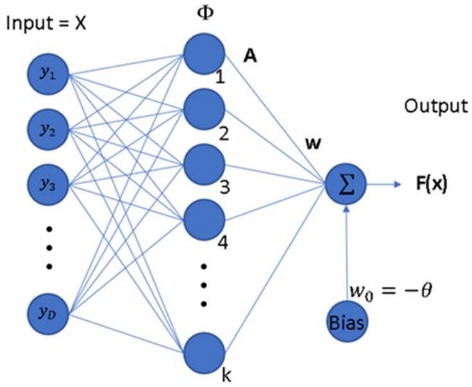


Figure 1. Radial basis function network

III. The Proposed Method

A. Architecture of the Proposed Method

The architecture of the proposed method is shown as Figure 2. The traffic flows are the input that are divided into training set and testing set. The training sets are first as input for the RBFNN for feature selection. In the process of feature selection, clustering is first being done, then the input is projected into RBF for weights calculation. After the RBFNN processed, the weight of each feature is calculated. We can ensure the weights of each feature, and only those features with non-zero weights are used in the next step. After that, the DNN model is built based on the number of features selected and start training. Then we use the well trained DNN model for testing with the testing set. The pseudo code of the proposed method is shown in Figure 2.

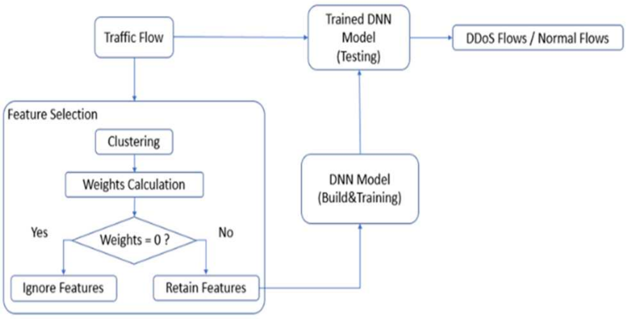


Figure 2. Architecture of the proposed method

B. Deep Learning Model

In the proposed method, DNN is chosen rather than GRU, CNN or RNN because we suppose that each DDoS flow information of the dataset is a single data and is not data with time series, therefore there is no advantage for GRU, CNN or RNN. The DNN model architecture of detecting flow whether it is DDoS attack or not is shown in Figure 3.

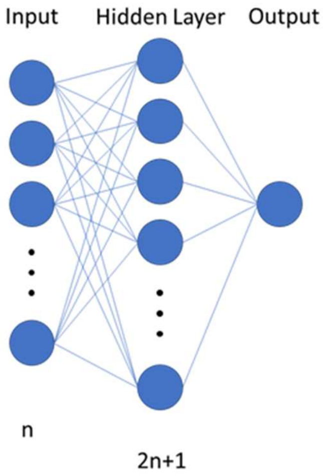


Figure 3. DNN model of the proposed method

In order to train the neural network, the number of inputs is n and the nodes of hidden layer should at least be 2n. Finally, there will be only one node for output since we expect the output to be 0 or 1, that represents whether it is a DDoS attack or not for the flow. For the optimizer, we use Adam that retains the gradient speed adjustment for the direction of the past gradient and Adam's adjustment of the learning rate of the square of the past gradient. In addition, Adam has the "offset correction" for the parameters, so that each learning rate will be determined.

C. *Radial Basis Function Network as Feature Selection*

In Figure 4, when a set of data X is inputted to the RBFNN, it is transferred to each RBF unit in the hidden layer. To calculate the distance of the data mapped into the RBF from the center, we use K-mean as the inductive algorithms to find out the symbolic center points c_k representing the data group. Then the input X is projected to another space through the radial basis function as A, after the calculation of linear regression, one can get the weight W.

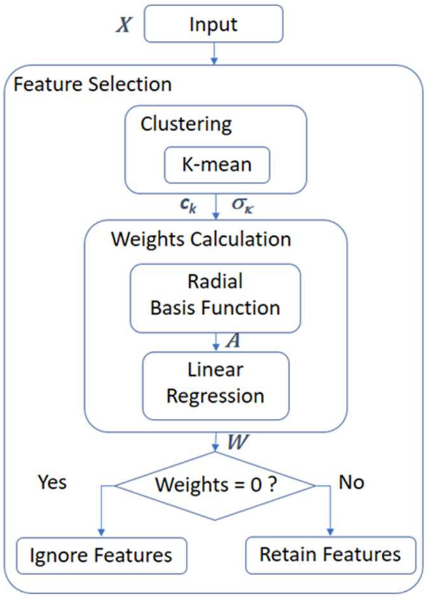


Figure 4. Radial basis function learning process

Given a data set $X=\{x_1,\cdots,x_n\} \in R^D$, the set of data X is clustered by the K-mean. The symbol // // denotes the Euclidean norm while σ_k is the standard deviation of each c_k and c_k is the center of a cluster S_k and D is the number of features. In the proposed method, the number of features is equal to the number of centers, which means K=D.

The average distance of c_k to all other centers c_m is calculated as λ_k, while k is the index of centers. Equation (1) is the formula to define λ_k.

$$\lambda_k = \sqrt{\frac{1}{M}\sum_{m=1}^{M} \|c_m - c_k\|^2} \qquad (1)$$

In equation (2), a_{nd} is the output of RBF unit in the hidden layer while d is the index of features.

$$a_{nd} = \Phi(\| X - c_k \|) \qquad (2)$$

In this paper, Gaussian function is used as the basis function and the following represents the range of the basis function and the formula as equation (3).

$$\Phi(\| X - c_k \|) = exp\left\{-\frac{1}{2\lambda_k^2}\|X - c_k\|^2\right\} \qquad (3)$$

After calculating on the input data transferred to RBF units, then add weights and transfer to the output layer, finally calculate the output through the linear regression.

$$F(X) = \sum_{k=1}^{K} w_k \cdot exp\left\{-\frac{1}{2\lambda_k^2}\|X - c_k\|^2\right\} + w_0 \qquad (4)$$

In equation (4), weight w_k is an output from each RBF unit to output layer, and d is the number of the RBF units in the hidden layer. The weight w_0 is the bias and usually is a constant and we set it as 0.

In matrix form, one can get equation (5).

$$F(x_n) = \sum_{d=1}^{D} w_d \cdot a_{nd} + w_0 \qquad (5)$$

while a_{n0}=1 , one can get the RBF matrix in Figure 5.

$$\begin{bmatrix} F(x_1) \\ F(x_2) \\ \vdots \\ F(x_n) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & a_{11} & a_{12} & \cdots & a_{1d} \\ 1 & a_{21} & a_{22} & \cdots & a_{2d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & a_{n1} & a_{n1} & \cdots & a_{nd} \end{bmatrix}}_{} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix}$$

$$F = AW$$

Figure 5. RBF in matrix

Then the weight W can be calculated as in equation (6).

$$W = A^{-1}F \qquad (6)$$

The weight W in equation (6) is calculated through the output F times the inversed input matrix A. Finally, only those features with non-zero weights are selected.

IV. Performance Evaluation

Before getting the experiment started, we analyzed the dataset. For CICIDS2017 [21], since there are two files, we use the dataset of Wednesday as the training set and Friday as the testing set. There are around 690,000 flows with 78 features and

1 label, ratio between normal flows and attack flows are around 6:4 for the dataset of Wednesday, while around 220,000 flows with 78 features and 1 label, ratio between normal flows and attack flows are around 6:4 for Friday. We randomly exchange 100,000 flows between the dataset of Wednesday and Friday and one of the features is repeated and then deleted.

The DNN model used in this experiment is shown in Figure 6 and parameter values in Table 1, there are n units as input and n hidden layers with 2n+1 unit, and with 1 output, while there are n features. For the environment, we used 2,000 epochs, batch size with 500, sigmoid as activation function, binary cross entropy as loss function and Adam as optimizer that are listed in Table 1.

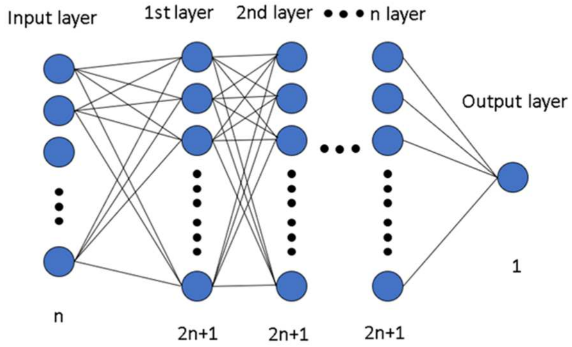


Figure 6. DNN model for binary output

Table 1. DNN environment

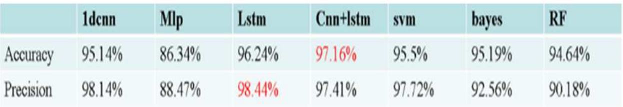
Model	Parameter	Value
DNN	Hidden layer	77
	Output unit	1
	Epoch	60
	Batch	500
	Activation function	Sigmoid
	Loss function	Binary_crossentropy
	Optimizer	Adam

The result of testing with all features is 97% in accuracy and 0.11 in loss as listed in Table 2. Comparing with Roopak's [6] results is listed in Table 3. The result is a little bit worse than one of Roopak's [6] results.

Table 2. Result of using all features

Number of Features	Number of Epochs	Accuracy	Loss
All	60	97.02%	0.11

Table 3. Roopak's [6] results



	1dcnn	Mlp	Lstm	Cnn+lstm	svm	bayes	RF
Accuracy	95.14%	86.34%	96.24%	97.16%	95.5%	95.19%	94.64%
Precision	98.14%	88.47%	98.44%	97.41%	97.72%	92.56%	90.18%

In order to enhance the performance, we apply RBFNN to do feature selection to remove some meaningless features. For those features get a weight will be kept. We find that some of the features have weight with zero, which means they are totally meaningless.

After feature selection, we build and retrain the DNN model with selected features, that means only 66 features are used. One can find that the accuracy increases rapidly and reach 99% within 20 epochs, and the accuracy and loss become stable around 60 epochs as listed in Table 4.

Table 4. Result with feature selection

Number of Features	Number of Epochs	Accuracy	Loss
All	60	97.02%	0.107
66	20	99.05%	0.044
66	60	99.36%	0.027
66	80	99.33%	0.029

Comparing again with Roopak's [6] results in Table 5, one can see that the proposed DNN model with RBFNN as feature selection has a result of accuracy with 99.36%, which is better than Roopak's LSTM and CNN+LSTM models' results.

Table 5. Comparing again with Roopak's [6] results

	Accuracy
LSTM (Roopak's results)	98.44%
CNN+LSTM (Roopak's results)	97.41%
DNN with feature selection (Our result)	99.36%

V. Conclusions

In this paper, we have proposed a DDoS detection method based on DNN with RBFNN for feature selection to detect flow-based DDoS attacks. Different from other detection methods, the proposed method can be used in detecting DDoS attack from network flows, especially exploitation-based DDoS attack flows, and classifying different types of DDoS attack flows. By training and testing with the dataset CICIDS2017, the accuracy of detecting all different kinds of DDoS attacks with 66 features is up to 99.36%. In conclusion with increasing of the complexity of the DNN and using RBFNN for feature selection, the proposed method can perform better than GRU and other existing deep learning models.

References

[1] N. Chockwanich and V. Visoottiviseth, "Intrusion Detection by Deep Learning with TensorFlow," 21st International Conference on Advanced Communication Technology, pp. 654-659, 2019.

[2] C. J. Hsieh and T. Y. Chan, " Detection DDoS Attacks Based on Neural-Network Using Apache Spark," 2016 International Conference on Applied System Innovation, pp. 1-4, 2016.

[3] J. P. Hou, P. P. Fu, Z. G. Cao and A. L. Xu, "Machine Learning Based DDoS Detection Through NetFlow Analysis," 2018 IEEE Military Communications Conference, pp. 565-570, 2018.

[4] Y. Wang, T. Hu, G. M. Tang, J. C. Xie and J. Lu, "SGS: Safe-Guard Scheme for Protecting Control Plane Against DDoS Attacks in Software-Defined Networking," IEEE Access, vol. 7, pp. 34699 - 34710, 2019.

[5] C. W. Syu, "A Deep Learning Based Real-Time Intrusion Detection and Prevention System for Software Defined Networks," Master Thesis, Department of Computer Science, National Taichung University of Education, pp. 1-86, 2019.

[6] M. Roopak, G. Y. Tian and J. Chambers, "Deep Learning Models for Cyber Security in IoT Networks," 2019 IEEE 9th Annual Computing and Communication Workshop and Conference, pp. 452-457, 2019.

[7] A. R. Gawande, "DDoS Detection and Mitigation Using Machine Learning," Master Thesis, ETD Graduate, Rutgers, The State University of New Jersey, pp. 1-35, 2018.

[8] A. S. Unal and M. Hacibeyoglu, "Detection of DDOS Attacks in Network Traffic Using Deep Learning," International Conference on Advanced Technologies, Computer Engineering and Science, pp. 722-726, 2018.

[9] C. H. Li, Y. Wu, Z. Z. Qian, Z. J. Sun and W. M. Wang, "DDoS Attack Detection and Defense Based on Hybrid Deep Learning Model in SDN," Journal on Communications, vol.39, no.7, pp. 176-187, 2018.

[10] M. V. O. Assis, L. F. Carvalho, J. Lloret and M. L. Proença, "A GRU Deep Learning System against Attacks in Software Defined Networks," Journal of Network and Computer Applications, vol. 177, pp. 1-13, 2021.

[11] A. E. Cil, K. Yildiz, A. Buldu, "Detection of DDoS Attacks with Feed Forward Based Deep Neural Network Model," Expert Systems with Applications, vol. 169, pp. 1-8, 2021.

[12] M. S. Elsayed, N. A. Le-Khac, S. Dev and A. D. Jurcut, "DDoSNet: A Deep-Learning Model for Detecting Network Attacks," IEEE 21st International Symposium on A World of Wireless, Mobile and Multimedia Networks, pp.391-396, 2020.

[13] M.W. Mak, "Radial Basis Function Networks", Available at

[14] https://www.cs.ccu.edu.tw/~wylin/BA/EIE520_RBFNetworks.ppt, pp. 5, 17.

[15] L. Zhou, M. C. Liao, C. Yuan, Z. Y. Sheng and H. Y. Zhang, "DDoS Attack Detection Using Packet Size Interval," 11th International Conference on Wireless Communications, Networking and Mobile Computing, pp. 1-7, 2015.

[16] A. Girma, M. Garuba, J. Li and C. M. Liu, "Analysis of DDoS Attacks and an Introduction of a Hybrid Statistical Model to Detect DDoS Attacks on Cloud Computing Environment," 12th International Conference on Information Technology - New Generations, pp. 212-217, 2015.

[17] J. Mirkovic, E. Arikan, S. Wei, S. Fahmy, R. Thomas and P. Reiher, "Benchmarks For DDoS Defense Evaluation," 2006 IEEE Military Communications conference, pp. 1-10, 2006.

[18] D. Belabed, M. Bouet and V. Conan, "Centralized Defense Using Smart Routing against Link-Flooding Attacks," 2nd Cyber Security in Networking Conference, pp. 1-8, 2018.

[19] Q. Yan, F. R. Yu, Q. X. Gong and J. Q. Li, "Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges," IEEE Communications Surveys & Tutorials, vol.18, issue 1, pp. 602-622, 2015.

[20] K. S. Bhosale, M. Nenova and G. Iliev, "The Distributed Denial of Service Attacks (DDoS) Prevention Mechanisms on Application Layer," 13th International Conference on Advanced Technologies, Systems and Services in Telecommunications, pp. 136-139, 2017.

[1] https://www.unb.ca/cic/datasets/