Analysis Distributed Denial-of-Service Attack Deploy Deep Learning Techniques

Sirajuddin Qureshi, Jingsha He, Saima Tunio, Nafei Zhu, Faheem Ullah, Ahsan Nazir, and Ahsan Wajahat

(Corresponding author: Nafei Zhu)

Faculty of Information Technology, Beijing University of Technology Beijing. 100124, China

Email: znf@bjut.edu.cn

(Received May 12, 2022; Revised and Accepted May 28, 2023; First Online Aug. 28, 2023)

Abstract

Network devices are essential to connect nodes and users on any given network. Network devices perform the additional task of protecting services and users from known and unknown attacks. This feature of network devices to stop or minimize network attacks to secure the nodes and all attached devices needs further research studies and experimentation to confirm their resilience against potential known attacks. Denial-of-Service (DoS) Attack is one of the deadliest attacks that make network services and devices unavailable. One of these attacks, which is growing significantly, is the Distributed Denial of Service (DDoS) attack. DDoS attack has a high impact on crashing the network resources, making the target servers unable to support valid users. The current methods use the standard datasets to deploy the Deep Learning (DL) Model for intrusion detection against DDoS attacks in the network. However, these methods suffer several drawbacks, and the used datasets do not contain the most recent attack patterns – henceforward, lacking in attack variety. In this paper, we proposed an interruption detection model system and, against DDoS attacks is based on DL technique, the combination of the Recurrent Neural Network (RNN) and Deep Neural Network (DNN) algorithms are compared with an autoencoder. We evaluated our DL model system using the newly released dataset CIC_DDoS2020, which contains a comprehensive variety of DDoS attacks and addresses the gaps of the existing current datasets (CIC_DDoS2020). We obtained a significant improvement in attack detection compared to other benchmarking methods. Hence, our model provides excellent confidence in securing these networks.

Keywords: Deep Learning (DL) Technique; Deep Neural Network (DNN Algorithm); Distributed Denial of Service (DDoS Attack); Intrusion Detection Against; Recurrent Neural Network (RNN Algorithm)

1 Introduction

The Distributed Denial of Services attack (DDoS) attack is a popular threat in the services provided online. Such a type of attack is used to target the packets and destroy different network resources such as the bandwidth of the network is compromised, the servers or the equipment can be crashed [20]. One of the useful methods to overcome the DoS attack is packet filtration on the DDoS attack routers that prevent DDoS attacks by identifying as well as blocking the attack before reaching its target. The authors provided another method for the prevention of DDoS attacks. An algorithm based on learning as well as the statistical analysis was proposed for a packet filtration system. [24] researched Distributed Denial of Services attacks (DDoS). The DDoS in the UDP-based network has been abused by troublemakers. Amplification systems can face these vulnerable challenges. The authors divided this problem into four steps. The first step was to monitor and classify the source of amplification that showed the high diversity in OS architecture.

Based on the results, the authors collaborated with the security community in a large-scale campaign for reducing the vulnerable NTP up to 92%. The authors analyzed and found the root cause of amplification attacks these may be the networks that are allowing the spoofing of IP addresses. The authors deployed a method for identifying spoofing-enabled networks. In our recent experiment, no significant differences in tuber yield of water yam strain cv. proposed a defense system for facing the Distributed Denial of Services attacks that is the combination of both software-defined controllers as well as the decision-making system based on fuzzy. The Numerical results in the research of the authors show that their proposed system has a very low computation load as well as the response times are high as much as 38. 04% of the N intake were thought to have been derived from the air and strains of nitrogen-fixing bacteria (NFB) were subsequently isolated from the stem and roots. The publisher regrets to inform the readers that the typesetter misinterpreted the correction from the author. The text 'In the case of the Andean condor, recent satellite tracking revealed that the home range of immature birds (299, 770 km2) is more than twofold that of adults (> 290, 000 km2) in northern Patagonia an attack prevention system known as Link Scope. According to the authors, it is a novel system that can employ both hope-to-hope as well as end-to-end network measure techniques for capturing the abnormal paths for detecting flooding attacks. The authors tackled several numbers of problems such measurement of long-scale internet paths. The authors deployed link scope in 7174 lines of the Python Programming code and detected with accuracy.

1.1 Distributed Denial of Services Attack (DDoS)

This research Distributed Denial of Services attack (DDoS). The DDoS in the UDP-based network has been abused by troublemakers. Amplification systems can face these vulnerable challenges. The authors divided this problem into four steps. The first step was to monitor and classify the source of amplification that showed the high diversity in OS architecture. Based on the results, the authors collaborated with the security community in a large-scale campaign for reducing the vulnerable NTP up to 92%. The authors analyzed and found the root cause of amplification attacks these may be the networks that are allowing the spoofing of IP addresses. The authors deployed a method for identifying spoofing-enabled networks.

In normal network circumstances, a user and server perform a three-way handshake. However, DDoS attack, a user sends a multiple handshake request by sending packets to the server but does not reply to the server by sending an acknowledgment. In this process, the connection between the user and server remains half-opened for a certain period of time. These half-open connections formulate and resource constraint on the server and the server remains busy for that whole period of time. In the meanwhile, if a legitimate user tries to develop a connection with the server, then the server declines to open any of the connections and hence this scenario can perfectly be called a denial-of-service attack. This TCP and IP connection is the backbone of the DDoS packets which are exchanged between a user and a server. This is also known as TCP/IP three-way handshake where these three steps are performed.

1) A client request to the server for a connection through DDoS: (Synchronizepackets);

2) The server replies with Synchronization-Acknowledgement: DDoS-ACK;

3) A client replies to the server with DDoS-ACK to develop and connection.

Different scientists have proposed different approaches to handle DDoS attacks at various network levels.

1.2 Router Based DDoS Attack

This research work encompasses and router-based DDoS attack. They further evaluated the router-based DDoS attack. The authors proposed a novel approach for preventing the DoS attack in-network, and an approach of DPF was introduced [23]. The authors have shown that the DPF is a packet filtration system that is able to achieve scalability and proactiveness as well. they have shown that there is an intimate relationship between the effectiveness of DPF in mitigating the DoS attack [22]. The silent feature of the authors' research was to filter the spoofed packet and prevent attacks.

To achieve this research study, it is very essential to analyze as well as to compare the quality of routing devices that are used to manage the network. More specifically, the focus may be kept on comparing the performance of Ubuntu router pc router DDoS attacks from both ends (Intranet and Internet). The Quality of Service (QoS) may also need to be analyzed by using a static routing method in accordance with parameter delay, throughputs, and loss of packets. According to collection comparison results, the DDoS attacks router is most stable in moderate conditions with 47 ms delay parameter 54 KBit/s throughput and time of crowded condition delay parameter 51 ms, packet loss 7% while router PC path is only stable at the low time seen from 45 ms delay parameter, throughput 54 KBit/s. In terms of comparison of hardware interface and software interface, router DDoS is better in the software interface, and PC router is better in hardware interfaces.

2 Related Works

[4] argues that most notions of flatness are problematic for deep models and cannot be directly applied to explain generalization. Specifically, when focusing on deep networks with rectifier units, we can exploit the geometry of parameter space induced by the inherent symmetries that these architectures exhibit to build equivalent models corresponding to arbitrarily sharper minima. Furthermore, if we allow to reparametrize of a function, the geometry of its parameters can change drastically without affecting its generalization properties. The key objective of a Distributed Denial of Service (DDoS) attack is to compile multiple systems across the Internet with infected zombies/agents and form botnets of networks [16]. The purpose of this paper is to detect and mitigate known and unknown DDoS attacks in real-time environments. We have chosen an Artificial Neural Network (ANN) algorithm to detect DDoS attacks based on specific characteristic features (patterns) that separate DDoS attack traffic from genuine traffic. Detection of DDoS attacks in the wake of flash crowds is a challenging problem to be addressed [6]. The existing solutions are generally meant for either flash crowds or DDoS attacks and more research is needed to have a comprehensive approach for catering to the needs of detection of spoofed and non-spoofed variants of DDoS

attacks. This paper proposes a methodology that can detect DDoS attacks and differentiate them from flash crowds. NS-2 simulations are carried out on the Ubuntu platform for validating the effectiveness of the proposed methodology. Nowadays, in the field of SDN, various machine learning (ML) techniques are being deployed for detecting malicious traffic. Despite these works, choosing the relevant features and accurate classifiers for attack detection is an open question. For better detection accuracy, in this work, Support Vector Machine (SVM) is assisted by kernel principal component analysis (KPCA) with a genetic algorithm (GA) [15]. In the proposed SVM model, KPCA is used for reducing the dimension of feature vectors, and GA is used for optimizing different SVM parameters. To reduce the noise caused by feature differences, an improved kernel function (N-RBF) is proposed. The experimental results show that compared to single-SVM, the proposed model achieves more accurate classification with better generalization. Moreover, the proposed model can be embedded within the controller to define security rules to prevent possible attacks by attackers. If the attack source is single, then the attack is referred to as denial of service (DoS) and if the attack is sourced from divergent servers, then it is referred to as DDOS. It is imperative from the analysis that there are constraints in the existing models since most of these models are user session-based and/or packet flow patterns [13]. The session-based evolution models are vulnerable to botnets and packet flow pattern-based models are vulnerable if attack sources are equipped with human resources and/or proxy servers. Hence, there is an inherent need for improving the solutions towards addressing the App-DDoS attacks over the system. The crux for such a system is about ensuring fast and early detection with minimal false alarms in streaming network transactions and ensuring that genuine requests are not impacted. To address such a system, the model of Bio-Inspired Anomaly-based App-DDoS detection is aimed, and the proposed model is depicted in detail along with experimental inputs. As for the mitigation approaches, to detect the flooding DDoS attack, the conventional schemes using the bloom filter, machine learning, and pattern analysis have been investigated. However, those schemes are not effective to ensure high accuracy (ACC), a high true positive rate (TPR), and a low false-positive rate (FPR) [3]. In addition, the data size and calculation time are high. Moreover, the performance is not effective from the fluctuant attack packet per second (PPS).

Threats of distributed denial of service (DDoS) attacks have been increasing day by day due to the rapid development of computer networks and associated infrastructure, and millions of software applications, large and small, addressing all varieties of tasks. Botnets pose a major threat to network security as they are widely used for many Internet crimes such as DDoS attacks, identity theft, email spamming, and click fraud [7,19]. Botnet-based DDoS attacks are catastrophic to the victim network as they can exhaust both network bandwidth and resources of the victim machine. An Ad hoc Network is a wireless multi-hop network with various mobile, self-organized, and wireless infrastructure nodes.

The goal of [1,5] is to implement a simulation model called DDoS Attack Simulation Model (DDoS) in Network Simulator 2(NS-2) and to examine the effect of DDoS attacks on various routing protocol types in MANET namely: Zone Routing Protocol (ZRP), Ad hoc On-Demand Distance Vector (AODV) protocol and Location-Aided Routing (LAR) protocol. The introduced model uses the NS-2 simulator to apply DDoS on the three chosen routing protocols. In terms of throughput and end-to-end latency under the consequences of the attack, the performance of three routing protocols was analyzed. Distributed Denial of Service (DDoS) attack has become one of the most destructive network attacks which can pose a mortal threat to Internet security. Existing detection methods cannot effectively detect early attacks. In this paper, we propose a detection method for DDoS attacks based on generalized multiple kernel learning (GMKL) combined with the constructed parameter R. The super-fusion feature value (SFV) and comprehensive degree of feature (CDF) are defined to describe the characteristic of attack flow and normal flow [4].

The network traffic was classified by the detection system in a controlled network environment using different sampling rates. In the experiments, raw network traffic of the CIC_DDoS 2020 [10,18], datasets and the raw network traffic captured in the customized testbed experiments were employed. DDoS attacks Detection system has reached high accuracy and low false-positive rate. Experiments were conducted using two Virtual network traffic classifieds.

Antidote system [2,19] presents a means of interaction between a vulnerable peripheral service and an indirectly related Autonomous System (AS), which allows the AS to confidently deploy local filtering rules under the control of the remote service.

3 Methodology

DDoS attacks on the Internet in a modern collaborative way. In this approach, the system collects network traffic samples and classifies them. Attack notification messages are shared using a cloud platform for convenient use by traffic control protection systems. Whole process is illustrated in Figure 1. The crucial steps from model build to system operation.

First, normal traffic and DDoS signatures were extracted, labeled, and stored in a database (CIC_DoS2020), which was then created using feature selection techniques. Finally, the most accurate MLA was selected, trained, and loaded into the traffic classification system. (e.g. architecture of the detection system was designed to work with samples of network traffic provided by industrial standard traffic sampling protocols, collected from network devices). The samples are received and grouped in

Table 1: Current related works

References	Dataset	Online	L/H_DoS	DoS_Attacks
[16]	CIC-DoS	True	False	True
[18]	None	False	True	False
[19]	DoS_Customized	True	False	True
[8]	Developed_Authors	False	True	True
[21]	CIC_DoS (2019)	True	False	False
Proposed DL Model	CIC_DoS (2020)	True	True	True

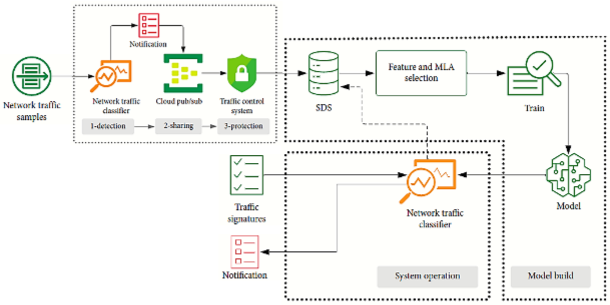


Figure 1: DDoS attacks on the Internet in a modern collaborative

flow tables in the receiver buffer. US, when the table length is greater than or equal to the reference values, they are presented to the classifier responsible for labeling them. If the flow table expires, it may be processed one more time. (e.g. occurrence of small flow tables is higher at lower sampling rates or under some types of DDoS attacks - DDoS flood attacks). Table 2 details the parameters for fine-tuning the system. (e.g. complete algorithm of the detection system is summarized in Figure 1). During each cycle of the detection process, traffic samples are received and stored in a flow for each new flow, and a unique identifier (FlowID) is calculated based on the 5-tuple (src_IP, dst_IP, src_port, dst_port, and transport_protocol) in Steps 1 and 2. If this is a new flow, [13, 14, 19], there is not any other flow table stored with the same FlowID, the flow table is registered in a shared memory buffer. Otherwise, if there is a flow table registered with the same FlowID such as the previously calculated one, the data of the new flow will be merged with the data in the existing flow table in steps 3 and 4. After the merging operation, if the length is greater than or equal to the reference value $Tl \leq Tmax$, the flow table is classified, and if it is found to be an attack, a notification is emitted. Meanwhile, in Step 7, the cleanup task looks for expired flow tables in the shared buffer, i.e., flow tables that exceed the expiration time of the system $E < ET$. For each expired flow table, the system checks the table length. If the flow table length is less than or equal to the minimum reference value $Tl \leq Tmin$, this flow table will be processed by Step 8. A new FlowID is calculated using the 3-tuple (src_IP, dst_IP, and transport_protocol).

3.1 Traffic Sampling

Detection uses a network traffic sampling technique because processing all the packets in the network can be a computationally expensive task, even if only the packet headers are parsed. In many cases, performing a deep inspection and analyzing the data area of the application layer is unfeasible for detection systems. Among the protocols adopted by the industry for sampling network traffic, the sFlow protocol is widely used in current devices, e.g. technique used by sFlow is called N-out-of-N sampling.

In this technique, n samples are selected out of N packets. One way to achieve a simple random sample is to randomly generate n different numbers in the range of 1 to N and then choose all packets with a packet position equal to one of the n values. Besides, the sample size is fixed in this approach [14, 24]. Flow monitoring system consists of an agent (embedded in a switch, a router, or an independent probe) and a collector. Architecture used in the monitoring system is designed to provide continuous network monitoring of high-speed switched and routed devices. Agent uses the sampling technology to capture traffic statistics from the monitored device and forward them to a collector system.

3.2 Feature Extraction

In supervised classification strategies, a set of examples is required for training the classifier model. (Is set is commonly defined as the signature database. Each instance of the database has a set of characteristics or variables associated with a label or a class. In this work, the goal is to identify characteristics in network traffic that can distinguish normal network behavior from DoS attacks. Study is focused on the analysis of the header variables of the network and transport layer packets of the TCP/IP architecture because it allows saving computational resources and simplifies the deployment in the ISP networks [11,12]. To achieve this research study, a proper literature review has been conducted to validate the null hypothesis. Additionally, a real environment, as well as a simulation setting, has been developed for experimentation of this research work. Additionally, PYTHON has been utilized to develop a proper testing environment by defining a DDoS router and by fine-tuning and optimizing it against DDoS attacks. At the initial stage, the router has been tested against a DDoS attack without fine-tuning the configuration against this kind of attack. Test readings have been taken, and data have been analyzed for comparison at the later stages. Since DDoS attack exploits TCP and IP three-way handshake and lets the half-open connection be used for malicious data transfer or theft of data. A similar attack has been simulated in Deep Learning Techniques. Readings of real vs simulated data have been compared and presented in the following subsequent sections. As the last step, the countermeasures of DDoS attacks have been taken to see if there is any improvement in router performance by comparing the result before and after taking countermeasures. This research study is going to help the research community to effectively take measures against DDoS attacks and make the performance of routers improved in many folds. In the following, a PYTHON simulation environment is presented where a test network is presented.

To effectively detect the flooding DDoS attack, we proposed lightweight detection using a bloom filter against flooding DDoS attacks. To detect the flooding DDoS attack and ensure high accuracy, a high true positive rate, and a low false-positive rate, the dec-all (decrement-all) operation and the checkpoint are flexibly changed from the fluctuant PPS in the bloom filter. Since we only consider the IP address, all kinds of flooding attacks can be detected without the blacklist and whitelist. Moreover, there is no complexity to recognize the attack. By the computer simulation with the datasets, the authors introduce the DDoS attacks and discuss the incapability of network-level detection methods for catching the DDoS attacks. These attacks are growing rapidly, are harder to detect, and cause severe problems in accessing a particular online service (or webserver) as compared to the Net-DDoS attacks. In invulnerability attacks, the attacker browses for unprotected openings in the software implementation and exploits them to bring the system down or to recruit zombies for further attacks. These attacks use the exacted performance of different protocols (such as TCP/IP and HTTP) to ravage the resources of the victim server and prevent it from processing events or requests from authorized users.

PYTHON network simulation environment contains DDoS router software called RouterOSWinBox v5. 20 which is installed on the PC having network connectivity with the network switch. Router_OS can also be installed on Router_BOARD and serve as Router Operating System. DDoSRouterOS relates to a switch. There are three more nodes that relate to the switch which are a test client, a web server, and an attacker machine. These three machines are connected on a separate VLAN of the switch. A test client is connected to VLAN 10, a web server machine is connected to Vlan 20, and an attacker machine is connected to Vlan 30. After achieving these connections, a ping is performed to see whether every node is reachable. As a major research contribution of this research work, it was of utmost importance to find out the attacker on the network. Since DDoS attack prevention is only possible when an attacker is found. We have achieved this task by examining ICMP packets. Since attackers commonly use ICMP protocol to generate a DDoS attack. Through properly configuring DDoS we traced out the origin of the ICMP packet. Specifically, we traced out the TCP and IP address of the attacker by properly configuring the built-in firewall for DDoS attacks. After tracing out the IP address of the attacker, the DDoS grab the IP address of the attacker. After tracing the TCP/IP address of the attacker we made sure to enlist the attacker's IP address into the built-in firewall by setting Level-2 (L-2) and Level-3 (L-3) policies. L2 is the network portion that is specifically associated with the local area network where no routing is required. On the other side, L-3 is the network portion of which is specifically associated with the routing portion means that routing is strictly required. As a summary of the handling of the attacker through DDoS attacks, we essentially perform the following four steps importantly what we need to look at in the above code is line 5. As we know that the ground_truth output(y) is of the form [0, 0. . . . , 1.. . 0] and predicted ŷ is of the form [0.34, 0. 03. . . . . . , 0.45], we need the loss to be a single value to infer the total loss from it. For this reason, we use the sum function to get the sum of the differences/error for each value in the y and ŷ hat vectors for that timestamp. The total_loss is the loss for the entire model inclusive of all time stamps.

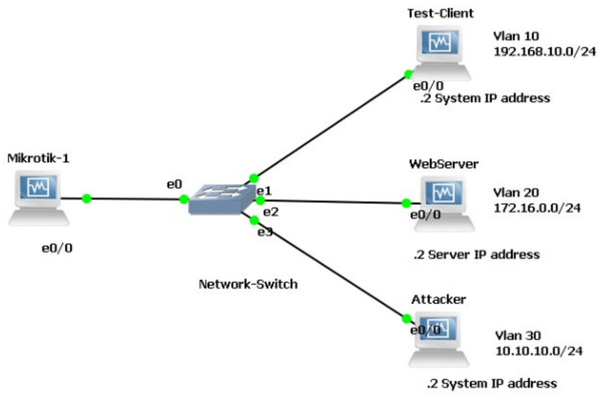


Figure 2: A particular online web server as compared to the DDoS attacks

To know more about the loss derivatives, please refer to this blog. There are two gradient functions that we will be calculating, one is the multiplication_backward and the other is addition_backward. In the case of multiplication _backward, we return 2 parameters, one is the gradient with respect to the weights (DLoss/DV) and the other is a chain gradient which will be a part of the chain to calculate another weight gradient. In the case of addition

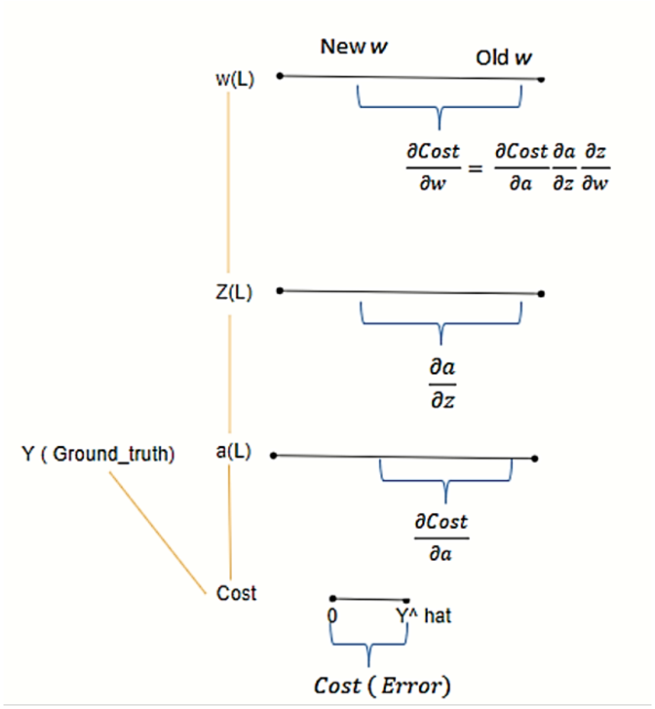


Figure 3: The total_loss is the loss for the entire model

backward while calculating the derivative we find out that the derivative of the individual components in the add function(ht_unactivated) are dh_unactivated/dU_frd= 1 as (h_unactivated = U_frd + W_frd_) and the derivative of dU_frd/dU_frd= 1. Since the cost is a function output of activation a, the change reflected by the activation is represented by dCost/da. Practically, it means the change (error) value seen from the point of view of the activation nodes. Similarly, the change of activation with respect to z is represented by DA/DZ, and z with respect to w is given by DW/DZ. We are concerned with how much the change (error) is with respect to weights. Since there is no direct relation between weights and cost, the intermediate change values from cost all the way to the weights are multiplied (as can be seen in the equation above). After configuring the network on PYTHON, it was important to test the stress level of the DDoS. Importantly, we must check whether processor usage is normal, or it increases while communicating. We have used the ping command to test the stress level of broadcasting a ping of 65000 bits/sec to verify the stress level of the DDoS attacks. The IP Ping command is to be incorporated here furthermore; it was important to verify the stress level SYN attack not only on DDoS but also on the other network nodes which are connected through DDOS Attacks.

3.3 System Architecture DDoS Attacks

The factors are data from the network history and network background of the DDoS attacks. These above-specified variables will be extracted from the given data and will be provided to the neural network layers as the input. Then the neural network layers are also given the target output for the mapping of the input variable to the corresponding output variable by adjusting the weight of DDos Networks attacks. The Activation Function is used to ease this task. An activation function of a layer defines the output of that layer given an input or set of inputs.

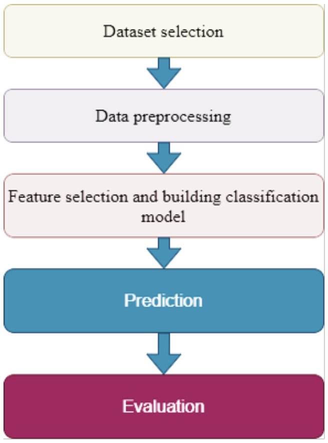


Figure 4: System architecture DDoS attacks

- Dataset selection;
- Data preprocessing;
- Feature selection and building a classification model;
- Prediction;
- Evaluation.

3.3.1 Dataset Selection

From the provided dataset by the (CIC_DDoS2020). We have modified it by decreasing the number of dimensions in the dataset for the implementation of our DDoS network services. The data collected for the process may contain missing values, noise, or DDoS network attacks. This leads to producing inconsistent information from the process. A data process with high-quality data will produce efficient data results. The dataset after selection and understanding is loaded into Python programming language.

3.3.2 Data Preprocessing

Import the Libraries: There are many libraries we have used for this experiment.

1) **NumPy:** which is the fundamental package for scientific computing with Python.
2) **Pandas:** is for data manipulation and analysis. In particular, we have used operations for manipulating numerical data.

3) **Matplotlib:** is a Python plotting library we have used to plot the figures in an interactive environment across platforms.

4) **Seaborn:** we have used to visualize the static data to data visualization based on matplotlib because it provides informative statistical graphics.

5) **NumPy:** is used for the multidimensional array as we have used in our work to compute with and manipulate these arrays. Fancy impute is used to handle the missing values because fancy impute can be easily used to replace missing values in huge data sets.

Import the Data-set: By using the Pandas library we import our dataset and the file I used here is used firstly we use CSV files because of their lightweight. After importing the dataset, we can see the head function (This function returns the first n rows for the object based on position. It is useful for quickly testing if your object has the right type of data in it.

Missing Values: The concept of missing values is important to understand in order to successfully manage data. If the missing values are not handled properly by the researcher, that's the way we have removed the missing values and have chosen the Imputation method to handle the missing values. This method can only be used with numeric data and we are using numerical data that's the way we have chosen the Imputation method, to replace the missing values within each column separately and independently from the others.

3.3.3 Feature Selection and Building Classification Model

Splitting the dataset into Training and Test In our model we have organized as training dataset contains 99% and the testing dataset 1% from the original data and the model learns on this data to be generalized to other data later on. We have the test dataset (or subset) in order to test our model's prediction on this subset and got a better result.

Feature selection: It is a process of selecting the most significant features from a given dataset. In many cases, Feature Selection can enhance the performance of a deep learning model as well. There are two types of feature selections Unsupervised, and Supervised, we have chosen the Supervised type which uses the target variable (e. g. remove irrelevant variables). For the usage of the Unsupervised type of feature selection, we have chosen the "Attacks" and for the Supervised type of feature selection, we have chosen the "Normal" variable for further processing.

Feature Scaling: It is a step of Data Pre-Processing that is applied to make independent variables or features of data. It basically helps to normalize the data within a particular range. And it also helps in speeding up the calculations in an algorithm and another befit is that is generally performed during the data preprocessing steps.

3.3.4 Prediction

It is important to use the correct model from various models present because the model chosen plays a crucial role in determining the efficiency, and accuracy of the prediction system. predicting models use this data to predict whether the particular case may be a loan default case or not. However, from various models, there is no specific model which can be said as the most optimal model. DNN has better adaptability than other predicting models and this model is able to construct a non-linear model and can better predict.

3.3.5 Evaluation

In the final stage, the designed system is tested with the test set, and the performance is assured. Evolution analysis refers to the description and model of regularities or trends for objects whose behavior changes over time. Common metrics calculated from the confusion matrix are Precision; Accuracy. The calculations for the same are listed below.

$$\begin{aligned}
\text{Precision} &= \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \\
\text{Recall} &= \frac{\text{True Positives}}{\text{True -nonPositive} + \text{False -non Negative}} \\
\text{F} &= \frac{\text{2 x Precision x Recall}}{\text{Precision} + \text{Recall}} \\
\text{Accuracy} &= \frac{\text{True Positives} \times \text{False -non Negative}}{\text{Total Sum of True Positive False Naative}}
\end{aligned}$$

In the reprocessing, classification, and evaluation of the traffic during Steps 4 and 5, the raw data traffic was replayed by TCP and IP Replay software in a specific DDoS port and sampled by the Flow agent for DDoS. The probability model is a conditional model over a dependent class variable with a limited number of outcomes means classes, and conditions on the feature variables F_1 to F_n.

$$P(c = F_1, \cdots, F_n)$$

If the value of n is large, basing a model is infeasible. Then we are reformulating the model then it is feasible or tractable.

$$\begin{aligned}
& P(c = F_1, \cdots, F_n) \\
\rightarrow \quad & [(P(c)P(F_1, \cdots, F_n) \rightarrow (F_1, \cdots, F_n)]
\end{aligned}$$

The above equation can be written in plain as follows

$$\textbf{posterior} = \textbf{(prior*like\_li\_hood)/evidence)}$$

We are only concentrating on the numerator because the denominator not depending on the class and values of

Algorithm 1 RNN usisg in DL Model

```
Input: Database Descriptors
Output: Selected Variables
Begin
Create empty optimized model set;
For i ⇐ 1 to Number of rounds do
    Define all the Descriptor database variables as current variables
    while True do
        Split database in training and test partition;
        Create and train the model using training data partitions;
        Select the most important variables from the trained model;
        Calculated the cumulative importance of variables from the trained model;
        if max ( cumulative importance of variables)¡ variables of importance threshold then
            Exit loop;
        end
        Train the model using only the most important variables;
        Test the trained model and calculate the accuracy;
        if Calculated accuracy < threshold then
            Exit loop;
        end
        Add current model to optimized model set;
        Define the most important variables from the trained model as the current variables;
end
```

Algorithm 2 DNN using in DL Model

```
Input: Network attacks Characterized by type
Output: Specialized attacks based on attack intentions
Lets a set attacks originating from different source $s\_1, s\_2, s\_3 \dots, s\_n$
Define $AR_k$ whereby $K, \leq 1$
Set AD as the conditions for attack dependencies $ad_1, ad_2, ad_3 \dots, ad_n$
Define AD for $AR_k$ whereby $AD$ , $AR_K \geq 1$
Designate $I$ as the set all probable attack intentions $i\_1, i\_2, i\_3 \dots, i\_n$
Do Define $I$ for each $AR_k$
While AR, AD is associated with $AR_k$
End While
End Do
End
```

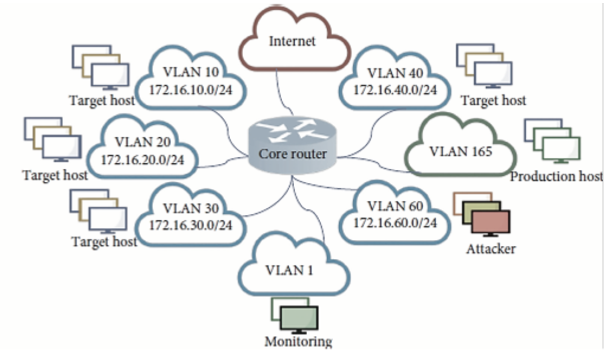


Figure 5: Customized topology, network traffic file is reprocessed on SVM-01 [16]

features F.

$$\begin{aligned} P &= TP + FP \\ R &= TP + FN \\ A &= TP + FN + FP + TN. \end{aligned}$$

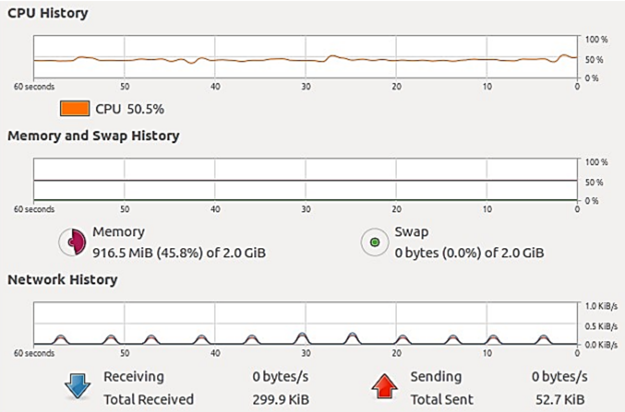
Evaluation Metric From the confusion matrix table Accuracy (A) and F-measure are the metrics that are used for the evaluation of the classifier performance. F- Measure is defined in terms of Recall (R) and Precision (P). If evaluation metrics have a higher value, then the classifier is best suitable for the data set. The evaluation metrics are described effectively by a confusion matrix.

System Setup: The DDoS attacks detection system has three main parameters that directly influence its performance. The parameters shown in Table 1 allow the user to calibrate the detection system according to the operating environment. In scenarios where the DDoS attacks are too large, for example, traffic samples are discarded before processing by the classifier. On the other hand, is too small, the True Positive (TP) increases because the classifier has little data to analyze. In the case of slow False Negative (FN) DDoS, low True Negative (TN) and also False Positive (FP) reduce the attack detection rate due to in-memory flow table-2 expiration time (E_T = 2).

Evaluation Metrics: System performance was evaluated using the Precision (PREC), Recall (REC), and F-Measure (F) metrics present in the literature [8,17]. PREC measures the ability to avoid false positives, while REC-Measures system sensitivity. F is a harmonic average between PREC and REC. In this context, (i) True Positive (TP) is the attack traffic predicted correctly, (ii) True Negative (TN) is normal traffic also predicted correctly, (iii) False Positive (FP) is the normal traffic predicted incorrectly, and (iv) False Negative (FN) is the attack traffic predicted incorrectly.

The DDoS attack Detection system and the classification result were compared with the attack plan. Figure 9 summarizes the procedures carried out by the proposed validation methodology.

Figure 5: Customized topology, network traffic file is reprocessed on SVM-01 [16] The customized topology network traffic file is reprocessed on SVM-01 [16], and the Flow agent collects traffic samples and sends them to DDoS attack Detection.



CPU 50.5%

Memory
916.5 MiB (45.8%) of 2.0 GiB

Swap
0 bytes (0.0%) of 2.0 GiB

Receiving 0 bytes/s
Total Received 299.9 KiB

Sending 0 bytes/s
Total Sent 52.7 KiB

Figure 6: Through properly configuring out the origin of ICMP packets

4 Results

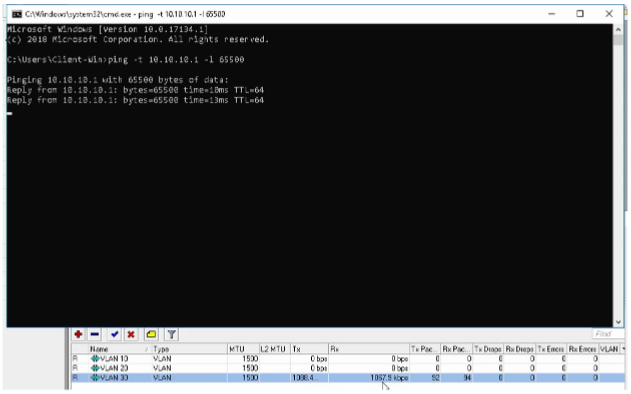
As a major research contribution of this thesis work, it was of utmost importance to find out the attacker on the network. Since DDoS attack prevention is only possible when an attacker is found. We have achieved this task by examining ICMP packets. Since attackers commonly use ICMP protocol to generate a DDoS attack. Therefore, we are encompassing the direct stress level of DDoS and the stress level of nodes connected through DDoS. We utilize the ping command to test the stress level of network nodes. We have verified the stress level of the Test-PC and Web server through the ping command.

The major finding of stress level on the single attack is presented in the following. It was recorded that when a single hacker is engaged in the process of a DDoS attack on DDoS, we observed 100% of CPU usage on the DDoS router see Figure 5. Additionally, it was observed that when the single hacker DDoS attack was performed on Webserver, a severe bandwidth constraint was observed as can be seen in Figure 6.

Figure 6: Through properly configuring out the origin of ICMP packets Through properly configuring DDoS we traced out the origin of the ICMP packet. Specifically, we traced out the IP address of the attacker by properly configuring the built-in firewall of DDoS. After tracing out the IP address of the attacker, the DDoS grab the IP address of the attacker. After tracing the IP address of the attacker, we made sure to enlist the attacker's IP address into the built-in firewall of DDoS by setting Level-2 (L-2) and Level-3 (L-3) policies. L2 is the network portion that is specifically associated with the local area network where no routing is required. On the other side, L-3 is the network portion which is specifically associated with the routing portion means that routing is strictly required. As a summary of handling the attacker through DDoS attacks, we essentially perform the following four steps.

1) To identify the IP address of the attacker through ICMP packets.

2) To add the IP address of the attacker into the firewall list to set the policy.



```
Microsoft Windows [Version 10.0.17134.1]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Client-Win>ping -t 10.10.10.1 -l 65500

Pinging 10.10.10.1 with 65500 bytes of data:
Reply from 10.10.10.1: bytes=65500 time=18ms TTL=64
Reply from 10.10.10.1: bytes=65500 time=13ms TTL=64
```

	Name	Type	MTU	L2 MTU	Tx	Rx	Tx Pac...	Rx Pac...	Tx Drops	Rx Drops	Tx Errors	Rx Errors	VLAN
R	VLAN 10	VLAN	1500		0 bps	0 bps	0	0	0	0	0	0	
R	VLAN 20	VLAN	1500		0 bps	0 bps	0	0	0	0	0	0	
R	VLAN 30	VLAN	1500		1088.4...	1067.9 kbps	92	94	0	0	0	0	

Figure 7: Single DoS attack having data burst stress level

3) To perform the above two steps on L-2 (which can be referred to as a chain input).

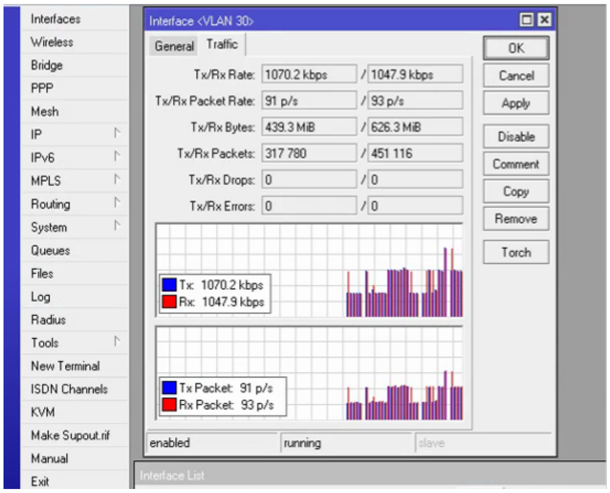
4) To perform the above two steps on L-3 (which can be referred to as chain forward).

This contains the outcome of the efforts made in this research study. Since the chief concern of this research study was to assess the resilience of DDoS routers against D0S Attacks. DoS attack causes the router to be overloaded and makes it reach its CPU usage to 100% and this attack causes the router to be unreachable by the clients or services. Additionally, in this research study, the DDoS_FLOOD attack which is one of the famous DoS attacks has been formulated on the DDoS router. The effectiveness of the DDoS_FLOOD attack on the DDoS router has been tested by properly observing the stress level of its processor in the PYTHON simulation environment. Furthermore, results have been taken precisely and presented in the following section for analysis.

4.1 Single DoS Attack

The impact of a single DoS attack (SYN attack) was measured on CPU stress and network bandwidth. In Figure 5. it can be seen that a single DoS attack put 2% stress on RouterOS, This stress level can be increased in many folds if a single DoS attack is converted into multiple DoS attacks and or if Data burst packet through broadcasting is increased. Additionally, the load on the network has also been verified which is 500 kbps on a normal single DDoS attack. It is pertinent to mention that this network load can be increased in many folds when it comes to multiple DDoS attacks.

After performing a single DDoS attack and a Double DDoS attack and it was required to check the resilience of the DDoS router on script attack. To perform the script attack we developed a script having a constant loop for increasing data bursts of 65000 each. As can be seen in Figure 5, a script can be seen to perform a double DDoS attack.



Interface <VLAN 30>

Tx/Rx Rate: 1070.2 kbps / 1047.9 kbps
Tx/Rx Packet Rate: 91 p/s / 93 p/s
Tx/Rx Bytes: 439.3 MiB / 626.3 MiB
Tx/Rx Packets: 317 780 / 451 116
Tx/Rx Drops: 0 / 0
Tx/Rx Errors: 0 / 0

Tx: 1070.2 kbps
Rx: 1047.9 kbps

Tx Packet: 91 p/s
Rx Packet: 93 p/s

enabled running slave

Figure 8: Distributed denial-of-service attack deliver connectivity to nodes and users on the given network

Network devices are using a vital role to deliver connectivity to nodes and users on any given network. Network policies perform the additional task of shielding facilities and users from known and unknown attacks. This feature of network policies to stop or curtail network attacks in order to make the nodes and all attached devices secure needs further research studies and experimentation to confirm their resilience against potential known attacks. The Distributed Denial-of-Service Attack (DDoS Attack) is known as one of the deadliest attacks which make network services and/or devices completely unavailable. DDoS routers are very well known for their performance and functionalities among their peers.

In this research study, the resilience of the DDoS router is going to be tested against DDoS attacks. The DDoS attack causes the router to be overloaded and makes it reach its CPU usage to 100% and this attack causes the router to be unreachable by the clients or services.

The four important data pre-processing techniques are data cleaning, data integration, data reduction, and data transformation. Here feature selection has a major role in preprocessing to select suitable features that affect the accuracy of the algorithm and it comes under data reduction. The data sets are implemented using the Deep Neural Network (DNN) model and the Recurrent Neural Network RNN model. The comparison of accuracies obtained from the two models was made and arrived at the conclusion was that the loan credibility behavior of DDoS network users can be predicted more accurately using the proposed model.

Table 3 showed that precision, recall, precision, f score, and accuracy are show attack and normal scores.

The proposed DL system model can reduce the data dimensionality by automatically extracting the features from input data. we also use various metrics to evaluate our proposed model, such as precision, recall, precision, F-score, and accuracy, to have a systematic benchmarking analysis with other related.

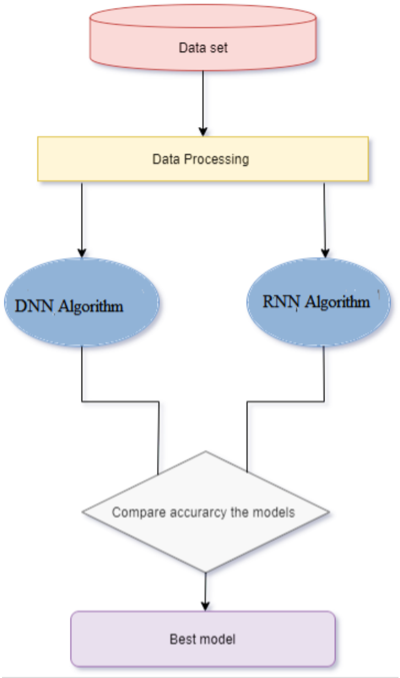


Figure 9: Proposed model for DDoS attacks dataset

Figure 10 is compared to show learners of RNN and DNN algorithms, and techniques achieved the best performance in terms of precision, recall, F-score, and accuracy of the RNN 99% and DNN 89. 78%. The ability of the ANN proposed model to deal with a high ratio of complex nonlinear relationships makes them promising techniques for detecting network intrusion. It can be used to tackle the limitation of the traditional classification methods, which are implemented to identify the anomalies in traffic based on the domain of the services. this paper is to represent the potential of the ANN proposed model for anomaly detection systems. We achieved a new technique based on RNN-Autoencoder classified the input traffic into normal or malicious types of URLs

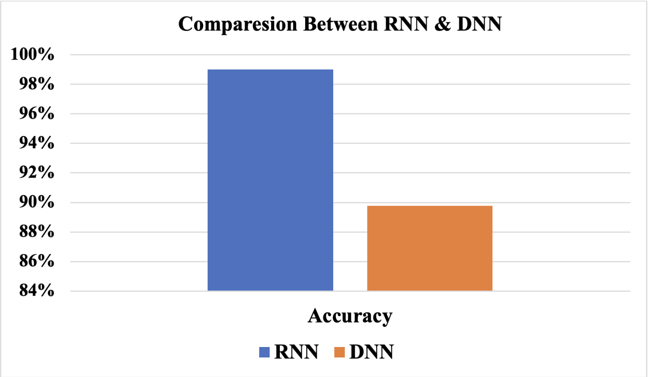


Figure 10: Compared between RNN and DNN for the DDoS Attacks

Table 2: Attack and normal values

	Attack	Normal
Attack	0.99	0.99
Normal	0.01	0.99

Table 3: Precision, Recall, Precision, F score, and Accuracy are showing attack and normal scores

Techniques	**Precision**		**Recall**		**precision**		**Accuracy**
	Attack	Normal	Attack	Normal	Attack	Normal	
DNN	**0.63**	0.74	**0.54**	0.67	0.56	0.70	**89. 78%**
RNN	0.99	**1.00**	0.99	**0.99**	0.99	0.99	**99%**

F-score, and the accuracy of the RNN is 99%.

5 Conclusion

This article is presented the DDoS attack Detection (ANN) proposed model system, an online approach to a DDoS attacks detection system. The internet network users and show learners compared RNN and DNN algorithms, and techniques achieved the best performance in terms of precision, recall, F-score, and accuracy of the RNN is 99%, and then DNN is 89. 78%, is classified network traffic based on samples taken by the flow protocol directly from network devices. In this research study, the resilience of the router is going to be tested against DDoS Attacks. The DDoS attacks cause the router to be overloaded and make it reach its CPU usage to 100% and this attack causes the router to be unreachable by the clients or services. Not only that but it causes all operations on packets performed by the router CPU such as packet filtering, TCP/IP ping, and logging, queuing may also cause overloading of the router. Particularly in this research study, DDoS attacks that are focused on DDoS attacks have been formulated on DDoS routers. The DDoS attacks have been tested in the PYTHON simulation environment and a physical environment and results have been taken for analysis and further processing which are implemented and identified the anomalies traffic based on the domain of the services. This research paper has represented the potential of DL techniques for anomaly detection (DL) proposed model system and we are achieved a new DL technique based on RNN-Autoencoder are classified the input traffic into normal or malicious types of URLs. This research study has fulfilled the gap in testing identified security DDoS network services against DDoS attacks and will help the research community to develop new mechanisms to make the routers more powerful against identified security, and DDoS attack holes systems. We achieved a new DL technique based on RNN-Autoencoder classified the input traffic into normal or malicious types of URLs.

6 Future Work

Furthermore, the major achievement of this research work is to secure the webservers through a firewall by hiding TCP/IP ping on the web services. If the webserver cannot be pinged then it is more secure. The most important thing to achieve is that with the current configuration simulation, we can track down the IP address of the hacker and can set policies.

References

[1] M. Abdelhaq, R. Alsaqour, M. Alaskar, F. Alotaibi, R. Almutlaq, B. Alghamdi, B. Alhammad, M. Sehaibani, D. Moyna, "THE resistance of routing protocols against DDOS attack in MANET," *International Journal of Electrical & Computer Engineering*, vol. 10, no. 5, 2020.

[2] J. Cheng, M. Li, X. Tang, V. S. Sheng, Y. Liu, and W. Guo, "Flow correlation degree optimization driven random forest for detecting DDoS attacks in cloud computing," *Security and Communication Networks*, pp. 1-14, 2018.

[3] S. Choi, Y. An, and I. Sasase, "A lightweight detection using bloom filter against flooding DDoS attack," *IEICE Transactions on Information and Systems*, vol. 103, no. 12, pp. 2600-2610, 2020.

[4] L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio, "Sharp minima can generalize for deep nets," in *International Conference on Machine Learning*, pp. 1019-1028, 2017.

[5] M. S. Elsayed, N. A Le-Khac, S. Dev, and A. D Jurcut, "Ddosnet: A deep-learning model for detecting network attacks," in *IEEE 21st International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM'20)*, pp. 391-396, 2020.

[6] J. Gera, B. P Battula, "Detection of spoofed and non-spoofed DDoS attacks and discriminating them from flash crowds," *EURASIP Journal on Information Security*, pp. 1-12, 2018.

[7] N. Hoque, D. K. Bhattacharyya, and, J. K. Kalita, "MIFS-ND: A mutual information-based feature selection method," *Expert Systems with Applications*, vol. 41, no. 14, pp. 6371-6385. 2014.

[8] F. S. D. Lima Filho, F. A. Silveira, A. de Medeiros Brito Junior, G. Vargas-Solar, L. F. Silveira, "Smart detection: An online approach for DoS/DDoS attack detection using machine learning," *Security and Communication Networks*, vol. 2019, Article ID 1574749, pp. 1-15, 2019. (https://doi.org/10.1155/2019/15747492019)

[9] M. Pokrinchak, and, M. M. Chowdhury, "Distributed denial of service: Problems and solutions," in *IEEE International Conference on Electro Information Technology (EIT'2)*, pp. 032-037, 2021.

[10] R. Panigrahi, and S. Borah, "A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems," *International Journal of Engineering & Technology*, vol. 7, no. 3, pp. 479-482.2018.

[11] S. H. Park, J. M. Goo, and C. H. Jo, "Receiver operating characteristic (ROC) curve: practical review for radiologists," *Korean Journal of Radiology*, vol. 5, no. 1, pp. 11-18. 2004.

[12] D. M Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," *arXiv preprint*, arXiv:2010.16061, 2020.

[13] K. M. Prasad, A. R. M. Reddy, and K. V. Rao, "BARTD: Bio-inspired anomaly based real time detection of under-rated App-DDoS attack on web," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 1, pp. 73-87, 2020.

[14] J. D. Rennie, "Tackling the poor assumptions of naive bayes text classification," in *International Conference on Machine Learning (ICML'03)*, 2003.

[15] K. S. Sahoo, B. K. Tripathy, K. Naik, S. Ramasubbareddy, B. Balusamy, M. Khari, and D. Burgos, "An evolutionary SVM model for DDOS attack detection in software defined networks," *IEEE Access*, vol. 8, pp. 132502-132513, 2020.

[16] A. Saied, R. E Overill, and T. Radzik, "Detection of known and unknown DDoS attacks using artificial neural networks," *Neurocomputing*, vol. 172, pp. 385-393, 2016.

[17] G. F. Scaranti, L. F. Carvalho, S. B. Junior, J. Lloret, M. L. Proença Jr, "Unsupervised online anomaly detection in Software Defined Network environments," *Expert Systems with Applications*, vol. 191, p.116225, 2022.

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

[19] S. Simpson, S. N. Shirazi, A. Marnerides, S. Jouet, D. Pezaros, and D. Hutchison, "An inter-domain collaboration scheme to remedy DDoS attacks in computer networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 879-893, 2018.

[20] J. R. Sun, M. S. Hwang, "A new investigation approach for tracing source IP in DDoS attack from proxy server", in *Intelligent Systems and Applications*, pp. 850-857, 2015.

[21] S. Ullah, M. A. Khan, J. Ahmad, S. S. Jamal, Z. Huma, M. T. Hassan, N. Pitropakis, W. J. Buchanan, "HDL-IDS: A hybrid deep learning architecture for intrusion detection in the internet of vehicles," *Sensors*, vol. 22, no. 4, p. 1340, 2022.

[22] A. Verma, R. Saha, N. Kumar, G. Kumar, "A detailed survey of denial of service for IoT and multimedia systems: Past, present and futuristic development," *Multimedia Tools and Applications*, vol. 81, no. 14, pp. 19879-19944, 2022.

[23] C. D. Xuan, M. H. Dao, "A novel approach for APT attack detection based on a combined deep learning model," *Neural Computing and Applications*, vol. 33, pp. 13251-13264, 2021.

[24] B. Zhang, C. Shen, B. Bealmear, S. Ragheb, W. C. Xiong, R. A. Lewis, R. P. Lisak, and L. Mei, "Autoantibodies to agrin in myasthenia gravis patients," *PloS One*, vol. 9, no. 3, p. e91816, 2014.

Biography

Sirajuddin Qureshi received his bachelor's degree in Computer Sciences from Quaid-e-Awam University of Engineering, Science & Technology, Pakistan. Afterwards, he pursued his Master's in Information Technology from Sindh Agricultural University Tandojam, Pakistan. Currently he is pursuing PhD in Information Technology at Beijing University of Technology, China. He has nine research publications to his credit as main author and co-author, which featured national and international journals and conferences. Sirajuddin's research areas includes but not limited to Network Forensics Analysis, Digital Forensics, Cyber security, Computer Networks and Network Security.

Jingsha He received the bachelor's degree in computer science from Xi'an Jiaotong University, China, and the master's and Ph. D. degrees in computer engineering from the University of Maryland, College Park, MD, USA. He worked for several multinational companies in USA, including IBM Corp. , MCI Communications Corp. , and Fujitsu Laboratories. He is currently a Professor with the Faculty of Information Technology, Beijing University of Technology(BJUT), Beijing. He has published more than ten articles. He holds 12 U. S. patents. Since August 2003, he has been published over 300 papers in scholarly journals and international conferences. He also holds over 84 patents and 57 software copyrights in China and authored nine books. He was a principal investigator of more than 40 research and development projects. His research interests include information security, wireless networks, and digital forensics.

Saima Tunio received the BSIT(Hons) with gold medal from Sindh Agricture University Tandojam, Pakistan.

Afterwards, she pursued her MSIT from Isra University Hyderabad, Pakistan. Currently she is pursuing PhD in Information Technology at Beijing University of Technology, China. She has more than five research publications to her credit as main author and co-author, which featured national and international journals and conferences. Saima's research areas includes but not limited to Information security. IoT security, Digital Forensics, Cyber security, Computer Networks.

Nafei Zhe received the B. S. and M. S. degrees from Central South University, China, in 2003 and 2006, respectively, and the Ph. D. degree in com- puter science and technology from the Beijing University of Technology, Beijing, China, in 2012. She was a Postdoctoral Research Fellow with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, from 2015 to 2017. She is currently an Associate Professor with the Faculty of Information Technology, Beijing University of Technology. She has published over 20 research papers in scholarly journals and international conferences. Her research interests include information security and privacy, wireless communications, and network measurement.

Ahsan Nazir has received his M. Sc degree from University of Engineering and Technology Lahore in 2016. From September 2015 to August 2018 he worked as software Engineer at Dunya Media group Lahore since September 2018 he is doing PhD in Software Engineering from Beijing University of Technology , Beijing China . He has published more than 10 journals and conference papers . His area of research include eGovernment, IoT, Software Engineering and Machine learning applications.

Ahsan Wajahat received the B. S. and M. S degrees in information technology from the Sindh agriculture University, Pakistan, in 2012 and 2016, respectively. He is currently pursuing the Ph. D. degree with the Beijing University of Technology, Beijing, China. His research interests include machine learning, information security, forensic network and data mining. He has received awards and honors from the China Scholarship Council (CSC), China.