INTRUSION DETECTION USING GRNN AND RANDOM FOREST

Pooja Bhadauria[1]
MTech. Student, Dept. of Computer Science & Engineering R.D. Engineering College at Duhai, Ghaziabad, India

Prof. Jaideep Kumar[2]
Professor & Head of Department, Dept. of Computer Science & Engineering R.D. Engineering College at Duhai, Ghaziabad, India

Received: 14 March 2020 Revised and Accepted: 8 July 2020

Abstract:

Along with the number of people building computer networks connected to the internet, then increasingly potential to cyber threats such as network intrusion (interference on the network). What includes intrusion in computer networks is the act of trying bypassing computer network system security mechanisms. One attempt to detect intrusion in the network is to differentiate network traffic activity. To distinguish normal network traffic activity with abnormal is difficult and tedious. Network analysts must examine all large and wide-ranging data to find the order anomalies (odd) on the network connection. GRNN and Random Forest can be used to group events on the network based on attributes. Each event on the network will be derived into a unique section by the decision tree. Order events on the network are mapped to the sequence of connected sections. By building rules based on part order generate intrusion alerts that can detect any attempt to do an intrusion. However, using the GRNN and Random forest over CICIDS2017 dataset the scheme is able to achieve 79.37% of accuracy over the cyber attacks.

Keywords: Cyber Security. Distributed Denial of Service, Machine Learning, Generalized Regression Neural Network, Random Forest

1. INTRODUCTION

It is inevitable that the internet has become a necessity for human life, especially India, which is a large country with a large population. Such as the data released by the Internet Service Providers Association of India, India ranks 11thworldwide in the number of attacks caused by servers that were hosted in the country, which accounts of 2,299,682incidents in Q1 2020 as compared to 854,782 incidents detected in Q4 2019 [1]; The large numbers of internet users in India is in fact directly proportional to cyber attacks. Almost 25% of Internet users in India have experienced cyber attacks once a year. This data shows that India is a country that is still fragile in preventing and dealing with cyber attacks. When it comes to internet users who are prone to be the target of attacks, internet users in India come from various backgrounds of interest. There are general users who only use the internet for the purpose of communicating with other people, most of these general internet users use cell phones as a medium of use the internet. There are also commercial internet users who use the internet to run a business; these users usually have a website for displays business products and services. Commercial user websites it connects to the business server and company administration computer [2]. Of these two examples of internet users, it is the second user who has a greater risk of hacking. An attacker can break into the server to destroy business assets or the attacker can break into the administration computer to access important user files. With the nominal amount of attack data in India as well as users who are vulnerable to cyber attacks, awareness of the importance of network security is still lacking. Not even a few internet users in India are unfamiliar with network security and how it impacts misuse of access in the network [3]. This situation will make hackers more interested in attacking sites, servers or personal computers of someone who does not really understand network security. The Distributed Denial of Service (DDoS) cyber security attacks in 2019 that India was in the top 15 countries in the world that received the most DDoS attacks, namely 3.96% of attacks. DDoS around the world; Figure 1 shows a diagram of DDoS attacks received by India according to Kaspersky Research Lab research data[4].

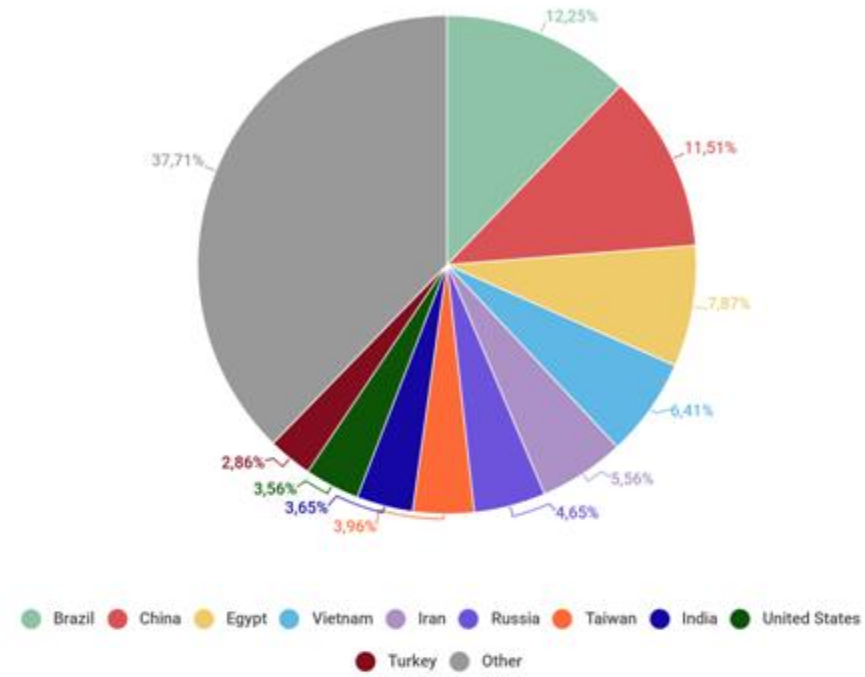


Figure 1: Country Based DDoS Attack Diagram[5]

From the data above, it should be noted that DDoS attacks cannot be ignored. The threat of DDoS attacks can paralyze the website server and of course the damage to the server operation causes the server to be inaccessible to any client. DDoS running by utilizing the flow of network traffic from the client to the server, so that DDoS cannot be rejected before the server crashes. One of the ways that can be done to prevent DDoS is by distinguishing normal traffic from DDoS attack traffic. This method can be done by implementing amalgamation of the machine learning methods, namely Random Forest and Neural Network using the previous data network traffic objects [6].

2. LITERATURE REVIEW

Among the various internet-based attacks, Denial of Service (DoS) attacks are a critical and ongoing threat to cybersecurity. In general, DoS attacks are implemented by forcing the victim's computer to reset or use up its resources, such as CPU cycles, memory or network bandwidth. As a result the targeted computers are no longer able to provide the services intended for legitimate users. When a DoS attack is managed by multiple distributed computers it is called a Distributed Denial of Service (DDoS) attack which is a popular attack method in cyberspace. From classic textbooks, we know security falls into three categories: confidentiality, availability, and integrity. It is clear that DDoS attacks fall into the availability category [7].

Distributed Denial of Service (DDoS) poses a major threat to the world of the internet, so many defense mechanisms have been proposed to combat them. Attackers continue to modify attack tools to counter the security system, until eventually researchers change their approach to dealing with the new attack [8-10]. The field of DDoS is developing very rapidly and this will become increasingly difficult to understand in view of the problem in general (Mirkovic and Reiher, 2004).With the emergence of new computing paradigms, such as Cloud computing and Mobile, and the emergence of interconnected technologies such as the Internet of Things, Denial of Services (DDoS) attacks have grown dramatically in volume, frequency, sophistication and impact making DoS (Denial of Services) is one of the most challenging threats on the Internet [11-14]. In March 2015, GitHub faced a major DDoS attack. The attack lasted for nearly a week and caused significant damage. In October 2016, a series of DDoS attacks were carried out en masse against Dyn's DNS servers, causing disruption to several major websites including Airbnb, Netflix, and Spotify. In September 2017, the UK National Lottery was subjected to a DDoS attack during peak times of operation, forcing the organization to take the lottery website and mobile app offline for about 90 minutes. In February 2018, Github was taken offline for about 10 minutes by a DDoS attack that peaked at 1.35 Tbps [15].

In machine learning research that discusses DDoS, training data is needed to support the development of a training model. Training data referred to as a dataset is past data that has been created or used by experts in the research field. [16] realized the importance of the correctness of the dataset for training data so that an evaluation was carried out on the eleven previous datasets that had existed since 1998. They got the result that the datasets in the past were outdated and their usefulness was not reliable in research. Some datasets are deemed to lack volume traffic diversity, in addition some datasets are deemed unable to cover multiple attacks, while others anonymous packages and payloads information that cannot reflect current trends. The research resulted in a new dataset called CICIDS2017 which is considered to have corrected the shortcomings of the previous dataset. Then [17] conducted an analysis on the CICIDS2017 dataset, they considered that the dataset still had several shortcomings. The research was carried out by re-marking the dataset by giving the information label provided by the Canadian Institute Cyber security. It was found that several classes were considered to be unbalanced, but this problem has been corrected by re-labeling steps. Random forest is an algorithm development of the Decision Tree using several sets of Decision Tree, where each Decision Tree is trained using individual samples that are randomly regrouped. In the classification process, individuals are based on votes from the most votes in the population tree collection. The resulting Random Forest has many trees, and each tree is constructed the same way. Trees with variable x will be built as far as possible with trees with variables y and in its development, as the dataset increases, the tree also grows. The placement of trees that are far apart means that if there is a tree around tree x, it means that the tree is a development of tree x [18-20].The Random Forest method can increase the accuracy result, because generating child nodes for each node is done randomly. This method It is used to build a decision tree consisting of a root node, internal node, and leaf node by randomly taking data attributes according to the applicable provisions. Root node is the node that is located at the top, or commonly referred to as the root of the decision tree. Internal node is a branch node, where this node has at least two or more outputs and only has one input. Meanwhile, the leaf node or terminal node is the last node that has only one input and no output [21]. There are three important aspects in the random forest method [22, 23] namely:

1. Perform bootstrap sampling to build multiple prediction trees.
2. Each tree predicts with predictors randomly.
3. Random forest make predictions by combining the results from each decision tree by selecting the majority value for classification or the average for regression.

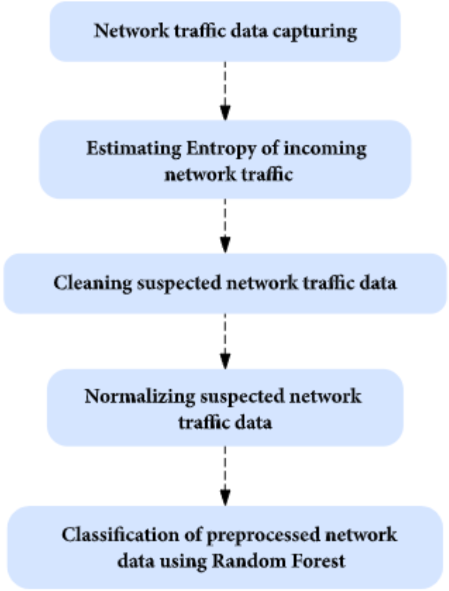

Network traffic data capturing
Estimating Entropy of incoming network traffic
Cleaning suspected network traffic data
Normalizing suspected network traffic data
Classification of preprocessed network data using Random Forest

Figure 2 : DDoS Detection using Random Forest [23]

According to Figure 2, the following shows an overview of the Random Forest flow [24]. Figure 2 conducted research on DDoS detection using data mining techniques, including the Random Forest. They randomly split the data into 60% for model training and 40% for testing. The training was carried out by building trees. They assume that the number of trees does not determine the high accuracy level and will even make the training time longer, therefore it is necessary to pay close attention to the improvement of random forest parameters. The results of training with Random Forest yield an accuracy of 98.02%.

Generalized Regression Neural Network GRNN is one of the networks included in the radial network base. Kernel nonlinear regression forms the basis of the operation of this GRNN, with the prediction model ***m(x)*** in the Kernel regression model ***y=m (x_i) + Ɛ*** as [25,26]]:-

$$\widehat{m}(x) = E(y|X) = \frac{\int_{-\infty}^{\infty} yf(X,y)dy}{\int_{-\infty}^{\infty} yf(X,y)dy} \quad (eq.1)$$

The main concept is that the output expected value is determined by the set of inputs (Spretch, 1991). Where ***y*** is the output that GRNN will predict, then ***X*** is the input vector consisting of p independent variables ***(x_1, x_2, ..., x_p)***, and the expected value of ***y*** against ***X*** or ***E [y | X]*** is the expected value of the output y given the input vector ***X*** and ***f (X, y)*** is the joint probability density function of ***X*** and ***y*** [27].

When the probability density ***f (X, y)*** is unknown, it can be estimated using the nonparametric consistent estimator proposed by the following equation:-

$$f(X,y) = \frac{1}{[2\pi^{(p+1)/2\sigma(p+1)}]} \frac{1}{n} \sum_{i=1}^{n} exp\left[-\frac{(x-x_i)^T(x-x_i)}{2\sigma^2}\right] exp\left[-\frac{(y-y_i)}{2\sigma^2}\right] \quad (eq.2)$$

Basically, the GRNN network structure is the same as a neural network in general, but because GRNN is based on Kernel regression theory, the output layer for GRNN is also derived from the estimated function ***m(x)*** in the Kernel regression model. Then the value $\hat{y}$ is applied on the basis of a neural network with simplified computation [28].

$$\hat{\mathbf{y}}(\mathbf{X}) = \frac{\sum_{\imath=1}^{n} \widehat{\mathbf{y}_\imath \mathbf{exp}}\left[-\frac{\mathbf{D}_\imath^2}{\boldsymbol{\sigma}}\right]}{\sum_{\imath=1}^{n} \mathbf{exp}\left[-\frac{\mathbf{D}_\imath^2}{\boldsymbol{\sigma}}\right]} \quad (\mathbf{eq.3})$$

Where$\mathbf{D}_\mathbf{i}^\mathbf{2}$ can be approached using the Euclidean distance between input vector and input weight vector in training data.

$$\boldsymbol{D_i^2} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{p}} (\boldsymbol{x_i} - \boldsymbol{v_{ij}})^{\mathbf{2}} \quad (\boldsymbol{eq.4})$$

There are 4 layers in the GRNN network, namely the input layer, pattern layer, summation layer and output layer. The network architecture formed in GRNN has the same number of neurons as the training input data set. Each layer in data processing with the GRNN network has a different role, first the data will be input via the input layer, this layer is only responsible for receiving input data to be processed. Then the input vector will then be forwarded to the pattern layer, in this pattern layer there is an activation function that functions as a network signal transfer.

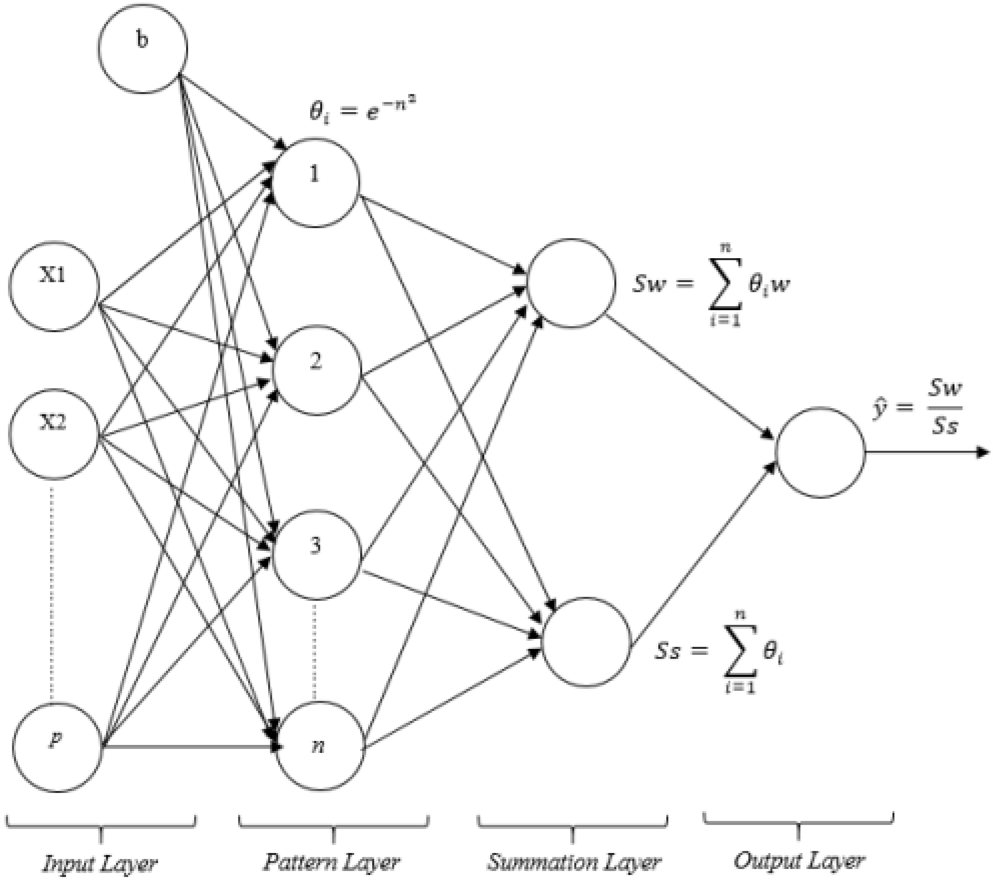

b
$\theta_i = e^{-n^2}$
1
2
3
n
X1
X2
p
$Sw = \sum_{i=1}^{n} \theta_i w$
$\hat{y} = \frac{Sw}{Ss}$
$Ss = \sum_{i=1}^{n} \theta_i$
Input Layer
Pattern Layer
Summation Layer
Output Layer

Figure 3: GRNN Architecture

In the GRNN network, the activation function used is radial basis activation as:-

$$\boldsymbol{\theta_i = e^{-n^2}} \;\; \textbf{\textit{With}} \;\; \boldsymbol{n = b.D_i} \; \textbf{\textit{b (eq.5)}}$$

is the bias weight of the input layer, specifically for GRNN the bias weight of the input layer is 0.8326 / spread for all neurons. Then $\boldsymbol{D_i}$ can be approached using the Euclidean distance.

The found distance will then be used to find the value of the activation function on each neuron in the input layer to the hidden layer. Then after the activation function is formed, it then enters the layer or the next layer, namely the summation layer, on this layer there are two the first network processing is the arithmetic summation of functions activation ***(Ss)*** and weighted addition ***(Sw),*** where the weight on this layer is the same with the target vector value.

$$\boldsymbol{Ss = \sum_{i=1}^{n} \theta_i \ \ (eq.6)} \text{ and } \boldsymbol{Sw = \sum_{i=1}^{n} \theta_i W_i \ \ (eq.7)}$$

After obtaining the sum of the activation value and the weighted value, then enter the output layer, where the output will be obtained from the value of the division between the weighted addition and the sum of the activation function.

$$\hat{\mathbf{y}} = \frac{\mathbf{Sw}}{\mathbf{Ss}} \ \ (\mathbf{eq.8})$$

3. PROPOSED METHODOLOGY

In this section the scheme explains the data used as objects in the study. There are two kinds of data in this research, namely secondary data and primary data. Both data have the same role, secondary data is a dataset from experts, while primary data is data constructed by researchers. Each data will be formed of training data and testing data, some of the data from secondary data will be used as training data which is useful for building a training model system and the other part from secondary data will be used as testing data which is useful for testing the model system has been built. The same rule applies to primary data which makes some data as training data and some others become testing data. Primary data is data that is obtained directly by researchers in the form of network traffic lines. To perform primary data acquisition in the form of network traffic, it is necessary to take the following steps:

a. Configure Hacking Environment

Primary data acquisition begins with building an environment as if it were a hacking environment. In the practice of hacking, researchers used Ubuntu-based version 6 of the BackBox operating system as the operating system for hackers. The BackBox operating system is installed on the VirtualBox tools. Just installing an operating system is certainly not enough to do hacking, it takes an additional tool to be able to run DDoS hacking practices on the target server. Researchers used a tool, namely Metasploit, a tool that runs on an Ubuntu-based operating system whose function is to carry out DDoS hacking practices on a target server. Besides being used to install a hacker operating system, VirtualBox is also used to install a virtual server, namely Metasploitable2. Metasploitable2 is what is then treated as the destination server. Metasploitable2 is of course in accordance with the hacking practices used by researchers because it is based on Metasploitable2's initial function, which is as a safe place to run penetration experiments and security research (Metasploitable 2, 2020).

b. DDoS Hacking Simulation

After the hacking environment is configured, the next step is to simulate hacking using the DDoS method. VirtualBox is used to run virtual servers, namely Metasploitble2 and the hacker's operating system, Ubuntu's BackBox. When Metasploitable2 is running in the background of the system, the researcher will open the Google Chrome web browser as a viewer for the Metasploitable2 virtual server. On the other hand, the BackBox operating system that has been run will be used by researchers to run the Metasploit tool, which is a tool that functions to run DDoS hacking commands. Metasploit runs on two or more tab windows which means hacking is executed by two or more Metasploit tools (bots) so that the implementation of distributed hacking is achieved, namely the Denial of Service hack which is distributed by the help of several bots. In addition to implementing distributed Denial of Service on servers, some of these bots aim to maximize hacking penetration so that targets can be paralyzed in less time. The DDoS hack was aimed at Metasploitable2 which was run previously.

c. Network Traffic Recording

This is where the Wireshark application comes in. When the Metaploitable2 server hacking process took place, hackers took advantage of the resources of the Metasploit tool to continuously send requests to the Metasploitable2 server to respond. As long as the paralyzed response expected by the hackers can be achieved, Wireshark is executed to monitor network traffic running on the Metaploitable2 server. In the Wireshark tool, the configuration is set to run network traffic capturing on VirtualBox, this is because the Metasploitable2 server runs on VirtualBox. During the hacking process, Wireshark performs its function to capture network traffic between servers and clients, in this case the Metasploit tool that acts as a client, in which this tool plays a dual role, namely as a client as well as a hacker. This network traffic recording step aims to obtain test data, so it is not enough to only get hacked traffic data. Apart from getting hacked data in the form of recording network traffic from the Metasploit request to the Metasploitable2 response, normal data is also needed. Normal data is obtained from normal requests on the client without using commands from the Metasploit tool, normal traffic is also recorded by Wireshark which is then treated as normal traffic. All network traffic data recorded by Wireshark is then saved in XML data format.

d. Network Traffic Extraction

After recording network traffic, all recording data is saved in the XML format. However, the XML format is not in accordance with the proper format in the data training process, so it is necessary to extract the data into a data that is in accordance with the data training process, namely the CSV data format. A tool that has a function to convert XML data to CSV is CICFlowMeter. CICFlowMeter itself is a generator and analysis tool for detecting network anomalies, which have been widely used in cybersecurity datasets such as Android Adware-General Malware dataset (CICAAGM2017), IPS / IDS (CICIDS2017), and Android Malware dataset (CICAndMal2017) (Ahlaskari, 2020)).

CICFlowMeter cannot run by itself, other tools are needed to run this CICFlowMeter. There are two options for running CICFlowMeter namely using JetBrains' Intellij IDEA or using Eclipse, research using Intellij IDEA. CICFlowMeter runs on the Intellij IDEA terminal, however, additional configuration is needed in the IDE terminal, namely Apache Maven installation. After installing Apache Maven, CICFlowMeter can be run on Intellij IDEA terminal. After installing CICFlowMeter, of course this tool is used to convert XML formatted data to CSV.

e. Labeling

The network traffic recording data has been obtained, and it has been converted into CSV format data, this CSV format data can be converted back into XLSX format data if in the study using XLSX format data. The data obtained from the simulation results still do not have a label to distinguish which one is hacked traffic data and which one is normal traffic data. Researchers carry out manual labeling, namely considering the time when the simulation is carried out. Researchers carry out simulations separately between hacking simulations and normal simulations, so that it is expected to get traffic that is in accordance with the activities carried out without being mixed with unexpected activities. Labeling is the final step in network traffic data acquisition.

For other data, namely secondary data. Secondary data is data obtained from existing sources. In this study the researchers decided to use secondary data called the CICIDS2017 Dataset from the Canadian Institute for Cybersecurity University of New Brunswick in Fredericton Canada. CICIDS2017 contains benign traffic and cyberattacks in general, which resemble real-world data in PCAP format and data from network traffic analysis using CICFlowMeter with CSV format (CICFlowMeter, 2020). Researchers used CSV / XLSX formatted data that was in accordance with the requirements of the next process. The following is a sample dataset taken from the CICIDS2017 dataset.

No	**Total Length of Fwd Packets**	**Fwd Packet Length Max**	**Fwd Packet Length Mean**	**Avg Fwd Segment Size**	**Subflow Fwd Bytes**	**Init Win bytes forward**	**Act data pkt fwd**	**Label**

1	13	7	7	7	13	33	2	*BENIGN*
2	7	7	7	7	7	31	0	*BENIGN*
3	58	52	28.5	28.5	58	1025	2	*BENIGN*
4	7	7	7	7	7	1027	1	*BENIGN*
5	27	21	9.636.666.667	9.636.666.667	27	8197	2	*DDoS*
6	25	7	7	7	25	257	3	*DDoS*
7	27	21	8.266.666.667	8.266.666.667	27	8197	5	*DDoS*
8	55	21	8	8	58	257	7	*DDoS*

Table 1: Sample Dataset CICIDS2017

Explanation of the stages of research system design as listed in Figure 4: below

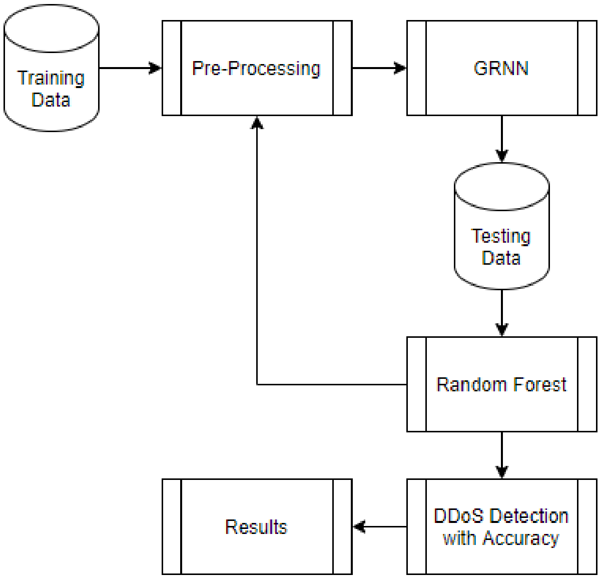


Figure 4: Workflow of Proposed Scheme

- **ALGORITHM GRNN**

The GRNN algorithm can be written as follows:

1. Searching for D_{ij} (distance of the i-th data with the j-th data)
2. Where i, j = 1, 2, 3, ..., Q

$$D_{ij} = \sqrt{\sum_{K=1}^{R} (P_{ik} - P_{jk})^2}$$

1. Looking for $a1_{ij}$ the results of activation with the radial basis function of the data distance multiplied biased. Where is with $a_{ij} = e^{-(b1*D_{ij})^2}$ spread b1 $= \frac{\text{TotalHits.}}{\text{spread}}$. Spread is the spread of data whose values are positive real numbers.

3. Looking for layer weight and layer bias weight.
4. Obtained output
5. Comparing training data with testing data.

- **ALGORITHM RANDOM FOREST**

```
function RandomForest(S , F)
    H ← ∅
    for i ∈ 1, . . . , B do
        S (i) ← A bootstrap sample from S
        hi ← RandomizedTreeLearn(S (i) , F)
        H ← H ∪ {hi}
    end for
    return H
end function
function RandomizedTreeLearn(S , F)
    At each node:
        f ← very small subset of F
        Split on best feature in f
    return The learned tree
end function
```

4. SIMULATION AND RESULTS

The data analysis method used in this research is two types of methods from the Neural Network branch, the first is generalized analysis and Random Forest for DDoS attacks evaluation.

- **Data Analysis Stages**

There are three outline stages that will be carried out in this research the first is data preprocessing, generalized analysis using the GRNN method and the last one is forecasting using the Random Forest method.

Data preprocessing stage: At this stage, the point is to prepare the data to be analyzed well, at this stage several things are done as follows:

1. Prepare data in XML format.

2. Check for missing data

3. Divide the data into 2 parts, namely training data and test data, proportions these are 75% for training data and 25% for test data.

- **Generalized analysis stage using the GRNN Method**

Here are some of the steps taken in venerability detection using the GRNN method:

1. Determine the optimal input variable

2. Conduct training on training data. At this stage it is done several trainings with different spread values up to find the spread value that corresponds to the smallest error value.

3. Perform the testing phase using the network that has been built at the training stage.

4. Make forecasts using a network that has been optimal based on the training and testing stages.

5. Obtain best results and conduct discussions.

- **Testing using Random Forest Method**

In the selection of features in the training data and test data, the algorithm used is Random Forest. This algorithm is often used for feature selection due to the tree-based strategy used naturally give an assessment of how well they increase the purity of the knot. This reduces the average impurities in all trees (called Gini Impurity). The node with the greatest decrease in impurity occurs at the beginning tree, while records with the least decrease in impurity occur at the end of the tree. Thus, by trimming the tree under the knot certain, we can make part of the most important features. For example the T data set contains examples from n classes, then the equation the Gini index as in equation below.

$$Gini(T) = 1 - \sum_{j=1}^{n} (p_j)^2$$

If the T data set is divided into two subsets T1 and T2 with the respective sizes N1 and N2, the Gini index of data separation contains examples from n classes, then the gini index (T) is defined as in equation below.

$$Gini(T) = \frac{N_i}{N}\, gini(T_i) + \frac{N_2}{N}\, gini(T_2)$$

Random Forest does not use all training samples in the construction of a tree but leaves a set of Out of Bag (OOB) samples, which can be used to measure the accuracy of forest classification. To measure the importance of certain features in a tree structure, feature values are randomized in the OOB sample and compare the classification accuracy between the complete OOB sample and the OOB sample with the specified feature. After assessing how important the features are in the data set by conducting training in each tree, some features were selected as important. Features are selected based on the standard rules of feature selection that is based on the average threshold value of each feature. Feature selection flowchart using Random Forest is shown in Figure 5.

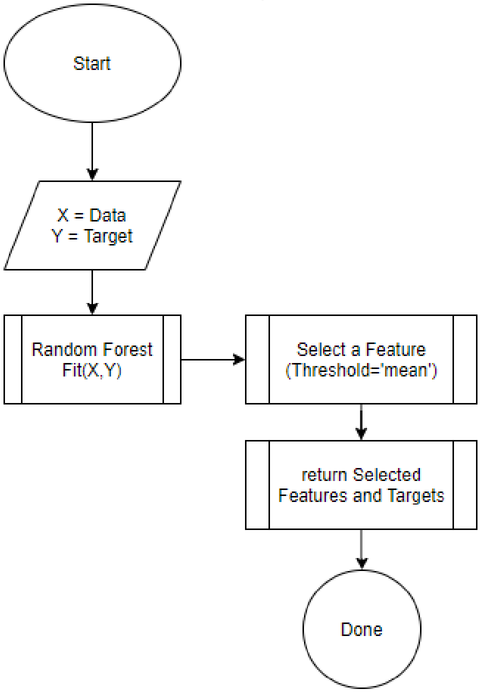


Figure 5: Feature Selection Flow chart with Random Forest

The following is an example of a manual calculation of feature selection using the Gini Importance Index in Random Forest. The data used are data that has been normalized as can be seen in Table 2.

No.	Sub Flow Data Pkts.	Fwd Act Data Pkts.	Fwd Seg Size Avg	Fwd Pkts. Len Max	Label
1	-1.126406	-1.128386	-1.214576	-1.225676	0
2	-1.221245	-1.229886	-1.324576	-1.225676	0
3	-1.251245	-1.269886	-1.314576	-1.285676	0
4	0.222627	0.229112	-0.322595	-0.278166	0
5	-0.121568	-0.102887	-0.329294	-0.278166	1
6	0.202627	0.212112	-0.312595	-0.278166	1

Table 2 Testing using Random Forest

1. For example the feature used is the Fwd Pkt Len Max (T) feature. Data on these features are separated. T has 3 values (3/6) which are equal to - 0.285676, 3 values (3/6) are the same as -0.278166.
2. For T == -0.285676 and Label == 0, 3/3 values are equal to 0.
3. For T == -0.285676 and Label == 1, 0/3 values are equal to 1.
4. $Gini(T) = 1 - \left(\left(\frac{3}{3}\right)^2 + \left(\frac{0}{3}\right)^2\right) = 1 - (1 + 0) = 0$
5. For T == -0.278166 and Label == 0, 1/3 the value is equal to 0.
6. For T == -0.278166 and Label == 1, 2/3 the value
7. $Gini(T) = 1 - \left(\left(\frac{1}{3}\right)^2 + \left(\frac{2}{3}\right)^2\right) = 1 - (0{,}11 + 0{,}44) = 1 - 0{,}55$
8. Then weigh and the amount of each separation based on the proportion of data each is divided

$$Gini(T) = \left(\frac{3}{6}.0\right) + \left(\frac{3}{6}.0.45\right) = 0 + 0.255 = 0.255$$

9. Obtained a value or score from the Fwd Pkt Len Max feature based on the calculation of the Gini Importance Index which is equal to 0.225.

10. Next is to do the same calculation on each feature of the data used for the results as depcited in table no.3

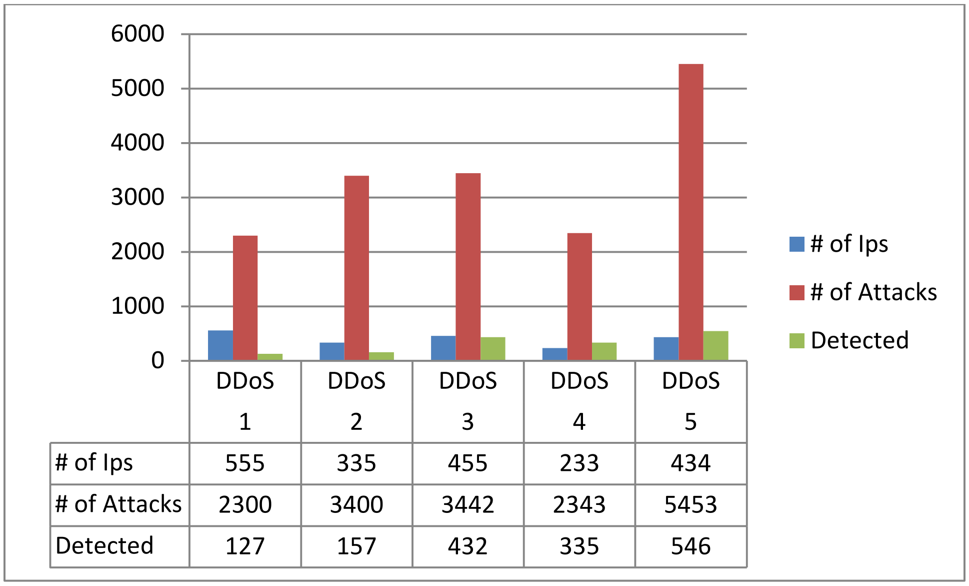
S.No	Category	# of Ips	# of Attacks	Detected
1	DDoS	555	2300	127
2	DDoS	335	3400	157
3	DDoS	455	3442	432
4	DDoS	233	2343	335
5	DDoS	434	5453	546
Average of Attacks		**402.4**	**3387.6**	**319.4**

Table 3: Evaluation using Proposed Scheme

$$Accuracy = \frac{Avg(Attacks\ Detected)}{Avg(\#ofIps)} * 100$$

$$79.370\% = \frac{319.4}{402.4} * 100$$

Therefore the above scheme produced the accuracy of 79.37% using GRNN and Random Forest.



	DDoS 1	DDoS 2	DDoS 3	DDoS 4	DDoS 5
# of Ips	555	335	455	233	434
# of Attacks	2300	3400	3442	2343	5453
Detected	127	157	432	335	546

Figure 6: Graphical Representation of Results Achieved

5. CONCLUSION AND FUTURE SCOPE

Conclusion: The research that has been done is a Distributed detection research Denial of Service (DDoS) uses the GRNN and Random Forest algorithm with objects network traffic research labeled network traffic BENIGN (normal) and DDOS (attack). There are two data used in the research consisted of secondary data and primary data. Secondary data used is the CICIDS2017 dataset which the data is a set of network traffic data lines built by experts in cyber security sector and is used for research purposes. Meanwhile the primary used is the network traffic data line that is built by the researchers themselves and obtained from the simulation results of DDOS hacking practices. Both types of data have the same role, namely as input to the detection system Distributed Denial of Service (DDOS) built by researchers. From both data will be divided into two parts for system training and system testing purposes, part of the data from secondary data is used as training data and partly others are used as test data, the same rules apply to primary data. Models built from training data on secondary data are used to perform testing data detection on secondary data. So is model built from training data on primary data used for testing data detection on primary data. In this study the conclusions obtained that from the results of tests carried out on each data in turn, the researcher conduct trials on secondary data first to find results, then continued with trials on primary data. The result shows that GRNN and Random Forest can detect data testing on the data secondary with accuracy result of 79.37%.

Future Scope: The researcher realizes that the research process is still from beginning to end far from perfect, it would be nice if the development continues so as to get better processes and results. From all data processing and the results obtained by the researchers, here are some suggestions from the researchers can be useful for interested parties, including:

a. In shaping a hacking environment and simulated attacks are expected to pay more attention to procedures so as to generate data sets proper and appropriate primer.

b. Be more careful in using the data used in the research.

c. Adding the application user interface to the program code.

REFERENCES

[1] ETCIO Spectrum Conclave, AUGUST-19-20, https://ciso.economictimes.indiatimes.com/tag/kaspersky
[2] Sandhya Keelery, Cyber crime in India - statistics & facts,Jul 15, 2020 https://www.statista.com/study/59177/cyber-crime-in-india/

[3] Bojana Dobran, 17 Types of Cyber Attacks To Secure Your Company From in 2020, FEBRUARY 21, 2019|IN SECURITY STRATEGY, RANSOMWARE, DATA PROTECTION, https://phoenixnap.com/blog/cyber-security-attack-types

[4] https://www.cloudflare.com/learning/ddos/famous-ddos-attacks/

[5] Oleg Kupreev, Ekaterina Badovskaya, Alexander Gutnikov, DDoS attacks in Q1 2019, https://securelist.com/ddos-report-q1-2019/90792/

[6] https://sucuri.net/guides/what-is-a-ddos-attack/

[7] Ni, Tongguang & Gu, Xiaoqing & Wang, Hongyuan & Li, Yu. (2013). Real-Time Detection of Application-Layer DDoS Attack Using Time Series Analysis. Journal of Control Science and Engineering. 2013. 10.1155/2013/821315.

[8] De Donno, Michele & Giaretta, Alberto & Dragoni, Nicola & Spognardi, Angelo. (2017). A taxonomy of distributed denial of service attacks. 100-107. 10.23919/i-Society.2017.8354681.

[9] Garg, Ankit. (2017). Distributed Denial of Service Attacks: A Survey.

[10] Mallikarjunan, Narasimha & Muthupriya, K. & Shalinie, S.. (2016). A survey of distributed denial of service attack. 1-6. 10.1109/ISCO.2016.7727096.

[11] Brooks, Richard & Ozcelik, Ilker. (2020). What is DDoS?. 10.1201/9781315213125-2.

[12] Brooks, Richard & Ozcelik, Ilker. (2020). DDoS Research: Evaluation. 10.1201/9781315213125-7.

[13] Brooks, Richard & Ozcelik, Ilker. (2020). Deceiving DDoS Detection. 10.1201/9781315213125-9.

[14] Onyeabor, Uche & Nehinbe, Joshua. (2020). An exhaustive study of DDOS attacks and DDOS datasets. International Journal of Internet Technology and Secured Transactions. 10. 268. 10.1504/IJITST.2020.10028403.

[15] LILY HAY NEWMAN, GitHub Survived the Biggest DDoS Attack Ever Recorded, https://www.wired.com/story/github-ddos-memcached/

[16] Sharafaldin, Iman & Habibi Lashkari, Arash & Hakak, Saqib & Ghorbani, Ali. (2019). Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy. 1-8. 10.1109/CCST.2019.8888419.

[17] Panigrahi, Ranjit & Borah, Samarjeet. (2020). The refined CICIDS2017 Dataset.

[18] Boehmke, Brad & Greenwell, Brandon. (2019). Random Forests. 10.1201/9780367816377-11.

[19] Berk, Richard. (2020). Random Forests. 10.1007/978-3-030-40189-4_5.

[20] Lee, Tae-Hwy & Ullah, Aman & Wang, Ran. (2020). Bootstrap Aggregating and Random Forest. 10.1007/978-3-030-31150-6_13.

[21] Joseph Sulistyo Nugroho, Nova Emiliyawati, Variable Classification System Of Consumer Acceptance Rate Against Cars Using Random Forest Method, Journal of Electrical Engineering Department of Electrical Engineering, Vol 9, No 1 (2017)

[22] Primajaya, Aji & Sari, Betha. (2018). Random Forest Algorithm for Prediction of Precipitation. Indonesian Journal of Artificial Intelligence and Data Mining. 1. 27. 10.24014/ijaidm.v1i1.4903.

[23] Karim Afdel and Mustapha Belouch, Detection System of HTTP DDoS Attacks in a Cloud Environment Based on Information Theoretic Entropy and Random Forest, https://www.hindawi.com/journals/scn/2018/1263123/

[24] Khan, Z., Gul, A., Perperoglou, A. et al. Ensemble of optimal trees, random forest and random projection ensemble classification. Adv Data Anal Classif 14, 97–116 (2020). https://doi.org/10.1007/s11634-019-00364-9

[25] Konakoglu, Berkant & Cakir, Leyla. (2018). Generalized Regression Neural Network for Coordinate Transformation.

[26] Al-mahasneh, Ahmad Jobran & Anavatti, S.G. & Pratama, Mahardhika. (2018). Applications of General Regression Neural Networks in Dynamic Systems. 10.5772/intechopen.80258.

[27] Al-mahasneh, Ahmad Jobran & Anavatti, S.G. & Garratt, Matt. (2019). Evolving General Regression Neural Networks for Learning from Noisy Datasets. 1473-1478. 10.1109/SSCI44817.2019.9003073.

[28] Alparslan, Chelenk. (2018). SUPPORT VECTOR REGRESSION AND GENERALIZED REGRESSION NEURAL NETWORKS FOR EVAPORATION MODELING.