TRIBHUVAN UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

A Seminar Report on

DETECTING DDOS ATTACKS USING LOGISTIC REGRESSION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE MASTER'S DEGREE IN COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

Submitted By

Brihat Ratna Bajracharya

19/075

Submitted to

Central Department of Computer Science and Information Technology

December, 2019

TRIBHUVAN UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY
CENTRAL DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

SUPERVISOR'S RECOMMENDATION

I hereby recommend that this seminar report is prepared under my supervision by **Mr. Brihat Ratna Bajracharya** entitled "**Detecting DDoS Attacks using Logistic Regression**" be accepted as fulfillment in partial requirement for the degree of Masters of Science in Computer Science and Information Technology.

……..

Mr. Jagdish Bhatta
Central Department of Computer Science
and Information Technology, TU

Date:

TRIBHUVAN UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

CENTRAL DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

CERTIFICATE OF APPROVAL

The undersigned certify that they have read, and recommended to the Institute of Science and Technology for acceptance, a seminar report entitled "**Detecting DDoS Attacks using Logistic Regression**" submitted by **Mr. Brihat Ratna Bajracharya** in partial fulfillment of the requirements for the Master's degree in Computer Science and Information Technology

Evaluation Committee

……..
Mr. Nawaraj Paudel
Head of Department
Central Department of Computer Science and Information Technology

……...
Mr. Jagdish Bhatta
Supervisor
Central Department of Computer Science and Information Technology

……..
Mr. Bikash Balami
Internal Examiner

DATE OF APPROVAL:

TABLE OF CONTENT

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
CSV	Comma Separated Values
DDoS	Distributed Denial of Service
DNS	Domain Name Server
DoS	Denial of Service
FBI	Federal Bureau of Investigation
FSM	Finite State Machine
HTTP	Hyper Text Transfer Protocol
IAT	Inter Arrival Time
ICMP	Internet Control Message Protocol
IP	Internet Protocol
LDAP	Lightweight Directory Access Protocol
LS-SVM	Least Square Support Vector Machine
MSSQL	Microsoft Structured Query Language
NetBIOS	Network Basic Input/Output System
NTP	Network Time Protocol
PNN	Probabilistic Neural Network
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENT

I would like to express my indebted gratitude and sincere thanks my supervisor Mr. Jagdish Bhatta for his continuous support, guidance and supervision by which this seminar has been possible. I would also like to thank Central Department of Computer Science and Information Technology for arranging such a schedule for the academic course.

Every attempt has been made to most of the details in each and every aspects of the seminar in this documentation so that the reader can clearly understand about the seminar. I would be pleased to get the feedback on this report. Finally I would like to express our gratitude to my family, friends and well wishers for encouraging and supporting me.

Brihat Ratna Bajracharya
19/075

ABSTRACT

In the present day digital world, most of the electronic devices have become smart and most of them have capability to connect to the internet. This capability has both pros and cons. The benefit is that these devices can be controlled from anywhere but possess a serious risk of being attacked and compromised. Those attack may have different intentions behind it like for fun, for ransom, theft of data, exploitation of privacy, and personal reasons. So, it is crucial to detect those attack before the attack could do any damage and also prevent similar attack in the future. One of the most common attack is the distributed denial of service (DDoS) attack in which the victim's machine is attacked from multiple systems across the internet using infected botnets of networks. The attacker controls those infected botnet and is programmed to launch attack packet flood. During the study of some of the researches relating to evidence gathering and attack detection system, the logistic regression classifier was implemented on CICDDoS2019 dataset and the result from the classifier was analyzed on two datasets. The first dataset consists of Portmap attack in which binomial logistic regression was used and second dataset consist of LDAP and NetBIOS variant of DDoS attack in which multinomial logistic regression method was used to classify these two variants with normal data. The accuracy for the Portmap dataset was found to be 99.91% with f1 score of 0.9913 and the accuracy for the LDAP dataset was found to be 99.94% with f1 score of 0.9847.

Keywords: ***network security, DDoS attack, classification, logistic regression***

1. INTRODUCTION

A DoS attack is a denial of service attack where a computer (or computers) is used to flood a server with TCP or UDP packets. In this attack, the continuous packets from attacker overloads the victim's computer resources making the service unavailable to other devices and users throughout the network. A DDoS attack is the most common type of DoS attack in which multiple systems target a single system with a DoS attack. The targeted network is bombarded with packets from multiple locations. DDoS attack is more complicated to recover. DDoS attacks are not only a widespread attack but also the second most common cybercrime attack to cause financial losses [3] according to the United States Federal Bureau of Investigation (FBI) and since the attack is generated from multiple computers across distributed network, it becomes very hard to differentiate DDoS attack from the genuine traffic. DDoS attacks can be done in two different ways, direct and indirect. Direct attack target the victim's machine in its weakness of the system while in indirect attack, attacks are performed on the elements associated with the victim's machine. Sometimes, there may be overload in the machine from the genuine traffic due to some breaking news or unexpected events. These events are called flash events. Although flash events resembles DDoS attack, it can be distinguished from DDoS attack because the access intent in case of flash event is genuine and shows natural flow pattern.

There are many methods of detecting these DDoS attacks, such as classification [1], regression [5], ANN based [2] and entropy based [6] which are presented in section 2 of this report. One of the method that can be used to detect DDoS attack is the logistic regression classifier. Logistic regression is used when the output of the model is categorical i.e. when the dependent variable of the model have distinct outcomes. If the classifier is used to classify the output into two classes (in our case, the traffic packet is either normal or is a variant of DDoS attack), then it is called binomial logistic regression. This classifier can also be extended to predict more than two outcomes which is called multinomial logistic regression.

This classifier is tested on two different dataset obtained from University of New Brunswick to detect the DDoS attack variant. Binomial logistic regression is used in one of the dataset to classify the packets into normal and Portmap variant of DDoS attack and multinomial logistic regression is used in second dataset to classify the packet into normal, LDAP or NetBIOS variant of DDoS attack.

2. LITERATURE REVIEW

Many researchers have contributed their time in research of effectively detecting the DDoS traffic differentiating from genuine traffic. Researchers have dived into detection using various methodologies and techniques. This section briefly points out some of the detection techniques used for the detection of DDoS attacks and to correctly identify flash events

2.1. Classification Based Approach

Sahi et al. [1] has proposed a new classifier system for the detection as well as prevention of DDoS TCP flood attacks in their paper. Their proposed system can identify these attacks regardless of the form of in which these attacks comes to the system. In this classification based system, the detection sub-system collects the incoming packet within a time frame, and then passed through the blacklist checker where the source IP of the packet is checked with the database of blacklist. If the source IP is matched in blacklist, it is directly sent to prevention for safety, If the packet passed the blacklist check, it is then passed through the classifier which decides whether the packet is normal or abnormal (from attacker). The normal packets are forwarded to the destination IP while the abnormal packets' source IP is added to the blacklist thus simplifying future detection and the packet is sent to prevention sub-system

2.2. Entropy based system

Gera et al. [6] has proposed the system based on the anomaly of traffic patterns to differentiate between DDoS flood packets and genuine flash event traffic. This system is based on the fact that DDoS are artificial packets generated with the sole purpose of attacking the victim to flood its resources. So, this attack must have some kind of pattern. On the other hand, flash events pattern are more genuine and random and this randomness factor between two can be used for the detection of DDoS flood packets. Some of the parameters that can be used for this kind of detection are as follows:

a. Time Interval – before the DDoS attack, the network traffic is generally low. Suppose at time t1 the network shows less traffic and when the DDoS attack starts, the network traffic will spike up. Let this time be t2. In DDoS attack the network traffic abruptly increase while in case of flash event, the traffic will increase gradually (i.e. spread of a breaking news takes place in gradually increasing pace and end in same fashion)

b. Source entropy – It is the number of source IP addresses from which attack is launched. The traffic that comes from the same network is considered traffic cluster. A threshold entrophy is defined. Then while analysing the incoming traffic, if it is found that there are more source IPs but less traffic cluster, it is considered a flash event (i.e. more number of people pinging the news some few times). Similarly, if there are more source IPs and more traffic cluster, it is considered as a spoofed DDoS attack (i.e. attackers are using multiple IPs to attack the victim's machine and these multiple source IPs are used multiple times). Finally if the source IP is same and the traffic cluster is same, it is also considered DDoS attack (i.e. attackers attacking the victim's machine from the same IP; non-spoofed).

2.3. Regression Based Network Traffic Analysis

Divakaran et al. [5] proposed a framework of gathering evidences to detect traffic sessions related to attacks and malicious activities. They applied regression models to detect fundamental anomalous patterns. Their framework include three stage approach for evidence gathering.

a. Modeling and analyzing sessions to detect anomalous patterns – they defined a flow as a set of packets localized in time, and studied the inter-arrival time (IAT) of the flows in a session. Attack bots are likely to generate flows in fixed interval of time, i.e. the randomness in the arrival time is lesser than in normal scenario. The flow size is also another vital parameter to distinguish anomalous session with normal genuine session. Another parameter is the degree of the end host (number of distinct IP addresses that an end host communicates to). DoS attack can be detected using the modeling session using the degree of destination IP address.

b. Detecting scans and illegitimate TCP state sequences – The anomaly can also be detected using the TCP state sequence, the normal flow of a TCP packet can be modeled in a finite state machine (FSM) with SYN, SYN+ACK, DATA and FIN packet. Any other flow that does not corresponds to this normal flow is considered illegitimate. From this FSM, we can say TCP flow is either legitimate or illegitimate is a binary decision depending on the FSM.

c. Evidence correlation and decision making – Spacial correlation is performed in this stage i.e. using IP addresses to correlate the anomalous pattern. One example is to look for the flows with the same IP address. Instead of spacial correlation, correlation in time can also be used but spacial correlation technique is more natural option as attacker may be passive for a while after successful exploit.

After the framework stages are completed, the analyzed data is processed to remove the outlier using a threshold for higher accuracy. The attack count if found nearly same will definitely raise suspicion. Also increasing count of attack packet with respect to time is also another type of anomaly detection technique. Regression models are then used to detect anomalous pattern. Authors have used linear regression to find the relationship between destination IP addresses with time. Quadratic regression is also used to detect anomalies using polynomial fit.

2.4. Using Artificial Neural Network

Saied et al. [2] used supervised ANN (feed forward, error back propagation) approach on three different type of packets: TCP, UDP and ICMP. The ANN ICMP uses three input nodes (namely source_ip, icmp_sequence, and icmp_ip) with four hidden layer nodes to get the result in one output node. Similarly, ANN TCP uses five input nodes (namely source_ip, tcp_sequence, source_port, destination_port, and tcp_flags) and four hidden layer nodes to get outpur output in one output node. And in ANN UDP, four input layer nodes (namely source_port, destination_port, source_ip, and packet_length) are taken with three hidden layer to get result in one output layer node. The ANN is trained to get output in either 0 or 1 indicating whether the packet is normal or abnormal respectively.

The basic flow for ANN based supervised learning is as follows:

1. DDoS detectors are installed in different network, each detector registering the IP of neighboring detectors and communicate via encrypted messages and continuously monitors the number of passing packets

2. If the number of packets are greater than certain threshold, then there is suspicion of the attack. The packets are sorted and IP of the victim is identified. The ANN retrieves the required patterns and prepares for the ANN engine. The trained ANN engine takes the input and produces the output.

3. The second step is repeated for three times and majority result is considered final out of the three outcomes.

This ANN engine needs to be trained regularly for detecting up-to-date DDoS attacks.

3. METHODOLOGY

The flowchart for the logistic regression classifier is shown in figure 1. The process starts by reading the dataset. The dataset then goes through data pre-processing where values in the dataset are made suitable for analysis. This processed dataset is then splitted into two parts. First part is used to train the logistic regression classifier and is called training set and remaining part called as testing set is used to predict the output of the modelled classifier. The predicted output is compared with actual output of the testing set result analysis. The implementation detail of logistic regression classifier is presented in section 4.

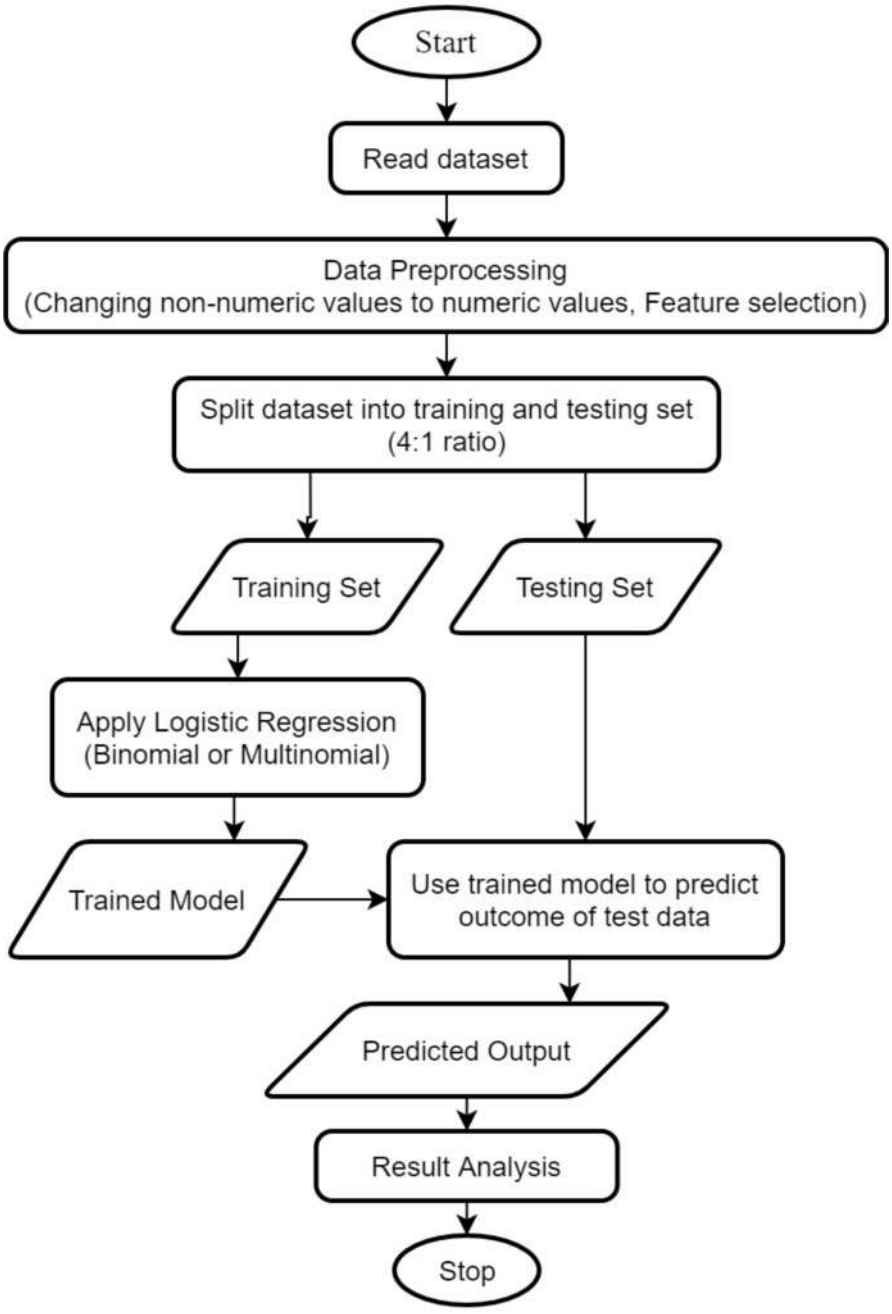

Figure 1: Logistic Regression Flowchart

3.1. Logistic Regression

Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous (only two possible outcomes) variable. In other words, it is used to predict a binary outcome (1/0, Yes/No, True/False) for given a set of independent variables. Logistic regression can be considered as special case of linear regression when outcome variable Is categorical which depends on lots of dependent variables (called features). Logistic Regression can also be used to determine cases of getting more than two outcome for e.g. married/unmarried/divorced. In this case, it is called multinomial logistic regression. Some familiar applications of logistic regression are email spam filter, fraud detection, cancer diagnosis, etc. [10]

The idea is to estimate the probability of an outcome being a 1 or a 0. Given that, the probability of the outcome being a 1 is given by p then the probability of it not occurring is given by 1-p. This can be seen as a special case of Binomial distribution called the Bernoulli distribution. Then the problem is converted in the form of generalized linear regression model given by $y=\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_n x_n$ where y is the predicted value, $x_1,x_2,\ldots,x_n$ are independent variables and $\beta_0, \beta_1,\ldots,\beta_n$ are coefficients to be determined. We can express this in vector form as $y=w^T X$ where $w=[\beta_0 \beta_1 \beta_2 \ldots \beta_n]$ and $X=[1 x_1 x_2 \ldots x_n]$.

Next we compute the odds as $odds=\frac{p}{1-p}$ and then take the natural log of the odds to make it continuously linear which is called logit given as $logit(p)=\log(\frac{p}{1-p})$. Now, we can write $logit(p)=y=w^T X$ This logit function acts as link between logistic and linear regression. We can now estimate the values for p by taking natural exponential on both sides. The final result will be $p(y=1)=\frac{1}{1+e^{-y}}$. This is known as the Sigmoid function. A sigmoid function is a bounded, differentiable, real function that is defined for all real input values and has a non-negative derivative at each point. A standard choice for a sigmoid function is the logistic function defined by the formula $S(y)=\frac{1}{1+e^{-y}}$. This function is also known as logistic function and has a characteristic S- shaped curve or sigmoid curve.

3.1.1. Steps in Logistic Regression

Step 1: Classifying inputs to be in class zero or one.

First, we need to compute the probability that a training set belongs to class 1 (we can also call it to be a positive class) using the Logistic Function. In this case, our y parameter is, as seen in the below given function.

$$P(y=1)=\frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\ldots+\beta_n x_n)}}$$

The coefficient β_0 , β_1, …, β_n in formula are selected to maximize the likelihood of predicting a high probability for observations belonging to class 1 and predicting a low probability for observations actually belonging to class 0.

Step 2: Defining a boundary values for the classifier

We now define a threshold boundary in order to clearly classify each given input values into one of the classes. We can choose a threshold value as per the business problem we are trying to solve, generally which is circled around 0.5. So if the probability values come out to be > 0.5 we can classify such observation into class 1 type, and the rest into class 0. The choice of threshold value is generally based on error types, which are of two types , false positives, and false negatives.

A false positive error is made when the model predicts class 1, but the observation actually belongs to class 0. A false negative error is made when the model predicts class 0, but the observation actually belongs to class 1. The perfect model would classify all classes correctly: all 1´s (or trues) as 1´s, and all 0´s (or false) as 0´s. So we would have FN = FP = 0.

3.1.2. Types of Logistic Regression

Logistic regression classifier can be classified into three types based on the outcomes that we use in the classifier. They are:

1. *Binomial logistic regression* – This type of regression is used when there are only two possible outcomes which can be in form of 0/1, Yes/No or True/False. Sigmoid function is used in classification in this type. The problem is first converted in form of generalized linear regression model as $y=\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_n x_n$ where y is the predicted value, x_1,x_2, …,x_n are independent variables and β_0, β_1,…,β_n are coefficients. Then the odds and logit (natural log of odds) are computed as $logit(p)=\log(\frac{p}{1-p})$. Finally natural exponential is taken on both sides of this logit to obtain $p(y=1)=\frac{1}{1+e^{-y}}$ which is the sigmoid function.

A threshold value is taken as a boundary between two possible outcome. The result from sigmoid function is the probability of the training set. Higher probability than threshold means that the training set belongs to one class and lower probability means that the training set belongs to other class.

2. *Multinomial logistic regression* – This regression type is used when we need to classify the outcomes into three or more possible classes. In this classifier, softmax function is used instead of sigmoid function. Softmax function is an activation function that turns logits into probabilities that sum to one. It outputs a vector that represents the probability distributions of a list of potential outcomes. The probabilities for each possible outcome for multinomial logistic regression is given by the softmax function defined as below:

$$P(y^i) = \frac{e^{y^i}}{\sum_{j=0}^{k} e^{y^i_j}}$$

where $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$, k is the number of outcomes considered and i runs from 0 to n.

3. *Ordinal logistic regression* – This is special type of multinomial logistic regression which is used when the possible outcomes are in order.

3.2. Confusion Matrix

A confusion matrix (also known as error matrix) is a predictor of model performance on a classification problem. The number of correct and incorrect predictions is summarized with count values and broken down by each class. The confusion matrix shows the ways in which the classification model is confused when it makes predictions on observations, it helps us to measure the type of error our model is making while classifying the observation into different classes.

		Actual	
		Positive	Negative
Predicted	Positive	**True Positive**	**False Positive**
	Negative	**False Negative**	**True Negative**

Table 1: Confusion Matrix

3.2.1. Elements of Confusion Matrix:

1. *True Positive (TP):* This refers to the cases in which we predicted "YES" and our prediction was actually TRUE
2. *True Negative (TN):* This refers to the cases in which we predicted "NO" and our prediction was actually TRUE
3. *False Positive (FP):* This refers to the cases in which we predicted "YES", but our prediction turned out FALSE
4. *False Negative (FN):* This refers to the cases in which we predicted "NO" but our prediction turned out FALSE

3.2.2. Learning Metrics From Confusion Matrix:

1. *Accuracy* – It answers how much the classifier is correct and is defined by following formula.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2. *Precision* – It answers how often the model correctly predicts positive values when it predicts positive and is defined by the relation

$$Precision = \frac{TP}{TP + FP}$$

3. *Recall* – It answers how often the model correctly predicts actually positive result and is defined by the relation. It is also known as sensitivity

$$Recall = \frac{TP}{TP + FN}$$

4. *F1-score* – It is the harmonic mean of precision and recall. So, it is defined as:

$$F1\,score = \frac{2 * Precision * Recall}{(Precision + Recall)}$$

4. IMPLEMENTATION AND ANALYSIS

4.1. Implementing Logistic Regression Classifier

Logistic regression classifier is implemented in Python 3 programming language using python libraries pandas, NumPy, scikit-learn, and Matplotlib. The first step in implementation is to read the dataset. The dataset is in CSV format. Python library pandas is used to read this dataset. Next step is the data preprocessing. This is done to format the data so that it can be used for analysis. Feature selection is an essential part of data preprocessing used to extract most relevant features out of all features from the dataset to reduce the dimensions of the dataset without much loss of the information. This also helps to interpret the importance of features in analysis.

After data preprocessing, the dataset is splitted into training set (which is used to train the model) and testing set (which is used to verify the trained model to evaluate its performance metrics). The training and testing set is generally taken in the ratio of 4:1. Another python library scikit-learn is used to split the dataset into training set and testing set. Logistic regression classifier is modeled using the training set obtaining the values of the coefficients.

Binomial logistic regression was used for the portmap dataset. In scikit-learn's LogisticRegression() function, Limited-memory Broyden–Fletcher–Goldfarb–Shanno (lbfgs) solver was used. This solver was selected as it stores only last few updates thus saving memory during computation and it is the default solver of latest Scikit-learn version 0.22.0. Also, maximum number of iterations was taken as 200 for the model to converge.

Similarly, Multinomial logistic regression was used for LDAP dataset. The solver selected in this case is also lbfgs as for the portmap dataset because it is relatively faster than other solver (like liblinear, newton-cg, sag, saga) and is memory efficient and also data do not need to be normalized. Another parameter named multi_class is used for multinomial analysis. And the maximum number of iterations is taken as 1000 for the model to converge.

These obtained coefficients are the classifier model for the dataset. This model is tested using the testing dataset. For this, the features of the testing dataset is given as input to the trained model and predicted output is obtained from the model. This predicted model is compared with the actual output of the testing set to evaluate the performance measure of our classifier. The result from above step is analysed using the confusion matrix. From the confusion matrix, various performance metrics can be calculated as described in section 3.2.2. Metrics

such as accuracy, precision, recall and f1 score is calculated to describe the result using confusion matrix.

4.2. Dataset Description

The dataset used for the implementation is the CICDDoS2019 dataset [9], obtained from University of New Brunswick (Canadian Institute of Cybersecurity, website: https://unb.ca).

	Unnamed_0	Flow_ID	Source_IP	Source_Port	Destination_IP	Destination_Port	Protocol	Timestamp	Flow_Duration
0	24	192.168.50.254-224.0.0.5-0-0-0	192.168.50.254	0	224.0.0.5	0	0	2018-11-03 09:18:16.964447	114456999
1	26	192.168.50.253-224.0.0.5-0-0-0	192.168.50.253	0	224.0.0.5	0	0	2018-11-03 09:18:18.506537	114347504
2	176563	172.217.10.98-192.168.50.6-443-54799-6	192.168.50.6	54799	172.217.10.98	443	6	2018-11-03 09:18:18.610576	36435473
3	50762	172.217.7.2-192.168.50.6-443-54800-6	192.168.50.6	54800	172.217.7.2	443	6	2018-11-03 09:18:18.610579	36434705
4	87149	172.217.10.98-192.168.50.6-443-54801-6	192.168.50.6	54801	172.217.10.98	443	6	2018-11-03 09:18:18.610581	36434626
5	0	172.217.9.238-192.168.50.6-80-54805-6	192.168.50.6	54805	172.217.9.238	80	6	2018-11-03 09:18:18.626325	3
6	1	172.217.9.238-192.168.50.6-80-54805-6	172.217.9.238	80	192.168.50.6	54805	6	2018-11-03 09:18:18.667379	2
7	144429	172.217.9.238-192.168.50.6-80-54805-6	192.168.50.6	54805	172.217.9.238	80	6	2018-11-03 09:18:18.667575	2
8	224	255.255.255.255-0.0.0.0-67-68-17	0.0.0.0	68	255.255.255.255	67	17	2018-11-03 09:18:18.758942	28870362
9	25	172.16.0.5-192.168.50.4-0-0-0	172.16.0.5	0	192.168.50.4	0	0	2018-11-03 09:18:19.155867	118365715

10 rows × 88 columns

Total_Fwd_Packets	...	Active_Std	Active_Max	Active_Min	Idle_Mean	Idle_Std	Idle_Max	Idle_Min	SimillarHTTP	Inbound	Label
45	...	2.833711e+04	98168.0	3.0	9529897.25	3.515826e+05	10001143.0	9048097.0	0	0	BENIGN
56	...	1.213149e+05	420255.0	4.0	9493929.75	3.515411e+05	9978130.0	8820294.0	0	0	BENIGN
6	...	0.000000e+00	62416.0	62416.0	36373056.00	0.000000e+00	36373056.0	36373056.0	0	0	BENIGN
6	...	0.000000e+00	62413.0	62413.0	36372291.00	0.000000e+00	36372291.0	36372291.0	0	0	BENIGN
6	...	0.000000e+00	62409.0	62409.0	36372216.00	0.000000e+00	36372216.0	36372216.0	0	0	BENIGN
2	...	0.000000e+00	0.0	0.0	0.00	0.000000e+00	0.0	0.0	0	0	BENIGN
2	...	0.000000e+00	0.0	0.0	0.00	0.000000e+00	0.0	0.0	0	1	BENIGN
2	...	0.000000e+00	0.0	0.0	0.00	0.000000e+00	0.0	0.0	0	0	BENIGN
5	...	0.000000e+00	2501634.0	2501634.0	8789576.00	2.955921e+06	10912366.0	5413491.0	0	0	BENIGN
40	...	3.462641e+06	7515650.0	103.0	9687762.70	5.445120e+06	18391321.0	5118819.0	0	1	Portmap

Figure 2: Portmap dataset

This dataset includes the result of abstract behaviour of 25 users based on the HTTP, HTTPS, FTP, SSH, and email protocols. This dataset consist of modern reflective DDoS attacks such as PortMap, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, SYN, NTP, DNS, and SNMP.

Among these variants, logistic regression on PortMap and LDAP variant of DDoS attack was studied and implemented. The dataset of Portmap attack consist of 88 columns and 191694 records. First ten records of this dataset are shown in figure 2. Out of 88, 87 columns are various features and last column ('Label' column) is the deciding class for the attack. This last column consist of two values: Benign and Portmap. This dataset consist of various features like Flow_ID, Source_IP, Source_Port, Destination_IP, Destination_Port, Protocol, Timestamp, Flow_duration, and many other features.

Similarly, another dataset of LDAP attack consist of 88 columns and 2113234 records. Like Portmap dataset, this dataset also has the same 87 features and one deciding class at the end called 'Label'. Unlike Portmap dataset, the deciding class has three values: Benign, NetBIOS, and LDAP. So, for this dataset, we need to use multinomial logistic regression.

As there are some non numerical values in some of the features in the dataset such as Source_IP, Flow_ID, Timestamp and most importantly, the Label column is also non-numeric which cannot be used in analysis. So these non numeric data need to be changed into numeric data. One way to do this is to assign a distinct numeric value to each unique non-numeric value. As such, all non-numeric data was changed into numeric data for both dataset using this technique. After conversion, the dataset looks as shown in figure 3. Now, the Source_IP, Timestamp, Flow_ID features contains numeric values.

	Unnamed_0	Flow_ID	Source_IP	Source_Port	Destination_IP	Destination_Port	Protocol	Timestamp	Flow_Duration
0	24	94238	161	0	33	0	0	80260	114456999
1	26	39286	137	0	33	0	0	25731	114347504
2	176563	177204	67	54799	179	443	6	132519	36435473
3	50762	156514	67	54800	10	443	6	150107	36434705
4	87149	132848	67	54801	179	443	6	23012	36434626
5	0	187888	67	54805	163	80	6	82928	3
6	1	187888	141	80	78	54805	6	185496	2
7	144429	187888	67	54805	163	80	6	113771	2
8	224	146542	154	68	176	67	17	134822	28870362
9	25	42694	202	0	68	0	0	23065	118365715

10 rows × 88 columns

Figure 3: Portmap dataset after changing non-numeric values into numeric values

4.3. Feature Selection

Since, there are a lot of features in the dataset and it is not possible to include all the features for predicting the kind of attack. Also it will be time consuming to train the model including all the features, we reduced the dimension of the dataset by employing feature selection. For selecting the best features that represent the data without much loss of information, we employed correlation technique to find out the relationship between these 87 features with 'Label' class. And, we choose some of the top features which are highly correlated with this deciding class

The features taken for Portmap dataset with its correlation coefficient are given below:

```
{'Protocol': 0.705635574606102,
 'Fwd_Packet_Length_Min': 0.7291026803636192,
 'Min_Packet_Length': 0.7291679201346289,
 'Source_Port': 0.8189050406122815,
 'Inbound': 0.8600933612454168,
 'Source_IP': 0.8660476930033244,
 'Label': 1.0}
```

Some features with lesser correlation coefficients for portmap dataset which are eliminated from analysis are given below:

```
{'Down_Up_Ratio': 0.6485234774068003,
 'URG_Flag_Count': 0.6150806663811492,
 'Bwd_Packet_Length_Min': 0.5505265792832202,
 'Destination_IP': 0.5434405331523054,
 'CWE_Flag_Count': 0.4208864290747812,
 'Avg_Bwd_Segment_Size': 0.41915492520184805,
 'Bwd_Packet_Length_Mean': 0.41915492520184805,
 'Fwd_IAT_Total': 0.3345362968706236,
 'Unnamed_0': 0.11675730059144739}
```

And the features taken for LDAP dataset with its correlation coefficient are given below:

```
{' Min Packet Length': 0.9276131094369818,
 ' Fwd Packet Length Min': 0.9277359022458002,
 ' Avg Fwd Segment Size': 0.9291694741755031,
 ' Fwd Packet Length Mean': 0.9291694741755031,
 ' Average Packet Size': 0.9292312255418383,
 ' Packet Length Mean': 0.9302060576330425,
 ' Fwd Packet Length Max': 0.9327888918318388,
```

```
 ' Max Packet Length': 0.9359158567754134,
 ' Label': 1.0}
```

And some of the eliminated features of LDAP dataset with their correlation coefficient are given below:

```
{' Protocol': 0.15101837887241756,
 ' Inbound': 0.14945687840206262,
 ' min_seg_size_forward': 0.05637313458482986,
 ' Fwd Header Length': 0.05629353491866543,
 ' Destination_Port': 0.012102190951823352,
 ' Bwd Header Length': 0.00633967032746798,
 ' Timestamp': 0.00018127588100869682,
 ' Flow ID': 0.0014231120955229765,
 ' SimillarHTTP': 0.014568932109410395,
 ' Active Std': 0.01713683548092619}
```

4.4. Regression Analysis

After selecting the most correlated features, the dataset was divided into train and test set in the ratio 4:1 and the regression model was trained with train set using scikit-learn's LogisticRegression() function. This was done for both datasets (Portmap and LDAP). The Portmap dataset consist of only two deciding classes in the 'Label' column, so binomial logistic regression was used while the LDAP dataset has three deciding classes, so multinomial logistic regression was used for this dataset.

Actual Label \ Predicted Label	NetBIOS	LDAP	Benign
NetBIOS	40488	2	14
LDAP	152	380867	50
Benign	29	1	1044

Figure 4: Confusion Matrix for LDAP Analysis

Actual Label \ Predicted Label	Benign	Portmap
Benign	936	5
Portmap	27	37371

Figure 5: Confusion matrix for Portmap Analysis

For portmap dataset, six features (Source_Port, Protocol, Fwd_Packet_Length_Min, Min_Packet_Length, Inbound, and Source_IP) were chosen for logistic regression analysis to

model the classifier and using the test set on modeled classifier, we computed the confusion matrix which is obtained as shown in figure 5.

Similarly for LDAP dataset, eight features (Min Packet Length, Fwd Packet Length Min, Avg Fwd Segment Size, Fwd Packet Length Mean, Average Packet Size, Packet Length Mean, Fwd Packet Length Max, and Max Packet Length) were chosen for classifier model. Using the trained model in test data, we computed the confusion matrix as shown in figure 4.

Using the values of confusion matrix, four performance metrics were calculated for both trained models. The calculated performance metrics are accuracy, precision, recall and f1 score. Performance metrics values for the Portmap dataset is obtained as below:

```
Accuracy: 0.9991653407757114
Precision: 0.9859144205687491
Recall: 0.9969822699890354
F1 Score: 0.9913826605347251
```

And performance metrics values for the LDAP dataset is obtained as below:

```
Accuracy: 0.9994132219085904
Precision: 0.9792599420442646
Recall: 0.9903806428884815
F1 Score: 0.984741430294035
```

4.5. Result Analysis

Accuracy is the ratio of correctly predicted observations to total observations. The accuracy for Portmap and LDAP dataset is found to be 0.9991 and 0.9994 respectively which means our portmap classifier model can detect 99.91% of Portmap variant and LDAP classifier model can detect 99.94% of LDAP variant of DDoS attack. The precision and recall score for portmap classifier is 0.9859 and 0.9969 respectively and the precision and recall score for LDAP classifier is 0.9792 and 0.9903 respectively. The higher score of all three metrics shows that we have modeled a good classifier for detecting DDoS attack. Finally, the f1 score is a great measure that takes into account both accuracy and precision used when we have multiple class and uneven distribution of data on these classes. The f1 score for portmap and LDAP classifier is 0.9913 and 0.9847 respectively which is very high suggesting that we have modeled a good classifier for detecting DDoS attack and its variant.

5. DISCUSSION AND CONCLUSION

In this study, a logistic regression classifier was used to predict the detection of three variants of DDoS attacks using the dataset obtained from University of New Brunswick (CICDDoS2019). Among different features in the dataset, a correlation test was done for each of the feature with the detection label to check the relevancy and few top features that are highly correlated with the label were used in the logistic regression. The Portmap attack detection used six features in the regression classifier and has an accuracy of 99.91 % detection with f1 score of 0.9913 while the LDAP and NetBIOS attack detection used eight features in the regression classifier and has an accuracy of 99.94 % detection with f1 score of 0.9847. Hence, from the result analysis of these performance metrics, we can conclude that the logistic regression classifier is suitable for detecting DDoS attacks and its variants.

REFERENCES

[1] A. Sahi, D. Lal, Y. Li, and M. Diykh, "An Efficient DDoS TCP Flood Attack Detection and Prevention System in a Cloud Environment", in *IEEE Access*, vol. 5, pp. 6036-6048, 2017.

[2] A. Saied, R. E. Overhill, T. Radzik, "Detection of known and unknown DDoS attacks using Artificial Neural Networks", in *Neurocomputing 172*, pp. 385-393, 2016.

[3] B. Cashell, W. D. Jackson, M. Jickling, and B. Webel, The Economic Impact of Cyber-Attacks, document CRS RL32331, *Congressional Research Service Documents*, Washington, DC, USA, 2004.

[4] C. Bedon, A. Saied, Snort-AI (Version 2.4.3), "Open Source Project", 2009. Available from: http://snort-ai.sourceforge.net/index.php/

[5] D. M. Divakaran, K. W. Fok, I. Nevat, and V. L. L. Thing, "Evidence gathering for network security and forensics", in *Digital Investigation 20*, pp. S56-S65, 2017.

[6] J. Gera, and B. P. Battula, "Detection of spoofed and non-spoofed DDoS attacks and discriminating them from flash crowds", in *EURASIP Journal on Information Security,* doi:10.1186/s13635-018-0079-6, 2018.

[7] M. Roesch, Snort (Version 2.9), "Open Source Project", 1998. Available from: http://www.snort.org/.

[8] R. Russell, Iptables (Version 1.4.21), "Open Source Project", 1998. Available from: http://ipset.netfilter,org/iptables.man.html/.

[9] *DDoS Evaluation Dataset "CICDDoS2019" Dataset*, [online] Available: http://205.174.165.80/CICDataset/CICDDoS2019/Dataset/CSVs/CSV-03-11.zip

[10] *"Logistic Regression For Dummies: A Detailed Explanation".* Accessed on: Dec. 9, 2019. [Online]. Available: https://towardsdatascience.com/logistic-regression-for-dummies-a-detailed-explanation-9597f76edf46