Machine Learning Based Effective Classification of Distributed Denial of Service Attacks

Aanshi Bhardwaj, Veenu Mangat, Renu Vig

Abstract: ***Distributed Denial of Service Attack (DDoS) is a deadliest weapon which overwhelm the server or network by sending flood of packets towards it. The attack disrupts the services running on the target thereby blocking the legitimate traffic accessing its services. Various advanced machine learning techniques have been applied for detection of different types of DDoS attacks but still the attack remains a potential threat to the world. There are mainly two broad categories of machine learning techniques: supervised machine learning approach and unsupervised machine learning approach. Supervised machine learning approach requires labelled attack traffic datasets whereas unsupervised machine learning approach analyses incoming network traffic and then categorizes it. In this paper we have attempted to apply four different classifiers for the detection of DDoS attacks. The four classifiers applied are Logistic Regression, Naïve Bayes, K- Nearest Neighbor and Artificial Neural Network. The chosen classifiers provide stable results when there is a large dataset. We compared their detection accuracy on KDD dataset which is a benchmark dataset in the field of network security. This paper is novel as it explains each pre-processing step with python conversion functions and explained in detail all the classifiers and detection accuracy with their functions in python as well.***

Keywords: Machine Learning, Logistic Regression, K-NN, Naïve Bayes, Artificial Neural Network.

I. INTRODUCTION

Distributed Denial of Service attack (DDoS) is comparatively simple but one of the deadliest attacks. It overwhelms the server with huge volume of normal or incomplete traffic. It compromises the services of the servers and makes it difficult to respond to legitimate users. It has been a nightmare for industrial operations, security and availability. The main motivation behind DDoS attacks can be blackmailing for money, demonstration of attack capabilities, hacktivism or business rivalry. The biggest DDoS attack till date has been experienced by Github on February 2018 with incoming traffic rate at 1.3 terabytes per second [1].

On 13th March 2019, Facebook suffered significant problems with its services. Users of Facebook and Instagram were not able to log into their accounts. Many people inspect this event to be related to DDoS but Facebook itself rejects.

Wikipedia and World of Warcrafts suffered from massive DDoS attack on 6th September 2019 [2]. The Wikipedia users of several European countries including UK, France, Italy, Poland, Netherlands, Germany suffered from outages due to this attack. By Friday evening Wikipedia led to total outage in US and other countries. The attack on World of Warcrafts effected the gaming experience of users by inducing latency and connection issues.

DDoS attacks can be targeted towards depleting bandwidth or depleting resources of a network or a combination of both these approaches. These attacks can be broadly classified into three main types based on different characteristics, methods, and attack vectors. These categories are: volumetric (Gbps), protocol (pps) and application layer (rps) attacks. Volumetric attack or floods target the bandwidth of the network and can be launched through botnets or amplification. Protocol attacks target the compute and memory of servers and intermediate devices and often work at layers 3 and 4 of the OSI model on network devices like routers. Most attacks can be categorized depending on the vector and packet size, and the categories often overlap. Application layer attacks or layer 7 attacks, target servers that execute a web application. These threats are harder to detect as attackers in most cases make legitimate requests like a website user, and need far fewer bots to attack. Consequently, these attacks also show up as much smaller traffic spikes. Application layer attacks are measured in requests per second (rps) or the number of requests made of an application. An application layer attack is considered a resource-based attack.

Many detection and prevention techniques have been devised but every year new attack vector comes up which fails the already available techniques. So, it becomes very important to have updated detection schemes. In this paper we have analyzed four different classifiers for the detection of DDoS attack on KDD dataset and compared their detection accuracy.

Section 2 provides related DDoS detection techniques. Section 3 presents the research methodology describing the tool, different classifiers and evaluation metrics. Section 4 depicts the experimental setup describing KDD dataset and experimentation results. Section 5 depicts conclusion and future scope.

Revised Manuscript Received on December 30, 2019.
***** Correspondence Author

Aanshi Bhardwaj*, M.E., Department of Information Technology ,UIET, Panjab University, Chandigarh (Panjab), India.

Veenu Mangat, M.E., Department of Information Technology ,UIET, Panjab University, Chandigarh (Panjab), India.

Renu Vig, Ph.D, Department of Engineering and Technology, Panjab University, Chandigarh (Panjab), India.

II. RELATED WORK

Authors proposed a detection system to detect DoS attacks based on Naïve Bayes classification has been proposed [3]. The system is primarily designed for transport layer i.e. TCP and UDP traffic. For training the classifier, the system extracts the traffic features like window arrival time, size of payload, source/destination port number, count of source/destination port number, packet interval gaps, etc. It computes the Naïve Bayesian model probabilities for different events and stores into a data structure offline. This data structure information is then used during deployment to determine the state of the network as normal or anomalous, at any point of time. Modi et al. [4] have developed a NIDS - Network Intrusion Detection System (NIDS) that incorporates Bayesian Classifier and Snort in the cloud environment to detect anomalies. Snort has the capability to identify well known and unspecified attacks in cloud. Experiments depicted that detection rate tends to increase by lowering false positives and false negatives. Bayesian classifier has high accuracy as compared to other classifiers like neural network classifier, decision tree, etc. It effectively predicts whether an event belongs to normal or intrusion class. The technique uses Snort which is based on signature-based technique and thus predicts known attacks. Chen et al [5] have proposed a detection system to protect key components of cloud computing against DDoS attacks. Cloud infrastructure employs Spark and Hadoop MapReduce to increase the data processing speed. K-means clustering algorithm is used for anomaly detection, wherein the detection results are visualized in web applications for better monitoring of security operations.

Anomaly based approach using neural network and particle swarm optimization (PSO) has been used for detection of DDoS attacks in cloud space by Rawashdeh et al [6]. A hypervisor based IDS is created which uses Artificial Neural Network (ANN) based classifier with PSO for finding optimal weights. Experimentation has been done on newly generated dataset consisting of TCP SYN flood and UDP attack. The proposed scheme outperforms ANN based model in terms of accuracy.

An intrusion detection system consisting of unsupervised technique i.e. hierarchical clustering and supervised technique i.e. Support Vector Machine (SVM) has been proposed [7]. The clustering algorithm first produces a high quality reduced data such that reduced dataset represents all the data points in the actual dataset. The produced dataset is given to SVM for training and testing on some selected parameters. Experimentation has been performed on KDD dataset and showed accuracy 95.72% and false positive rate of 0.7%.

Some statistical parameters have first been selected in [8], such as interval arrival packet time mean from same IP address, probability of existence of an IP in 15 seconds, answer, authority and additional resource record, minimum, average and maximum packet size. This data has been applied for training and testing of various machine learning algorithms- Decision Tree (DT), Multi-Layer Perceptron (MLP), Naïve Bayes (NB), and Support Vector Machine (SVM) for detection of DDoS attacks. It has been shown that decision tree gives the detection accuracy of 99.3%. Information Gain, Gain ratio and Chi Square methods have been used for feature selection.Researchers have first defined a pre-processing module which removes the redundant data having low correlation [9]. The processed data then goes to detection module. In detection module, Snort detects known attacks by matching the attack pattern with known rules present in knowledge base. Then machine learning based method i.e. C4.5 decision tree, detects the unknown attacks and predicts whether the user is legitimate or not. Openstack Juno has been used for setting up a cloud environment. Experimental results show that C4.5 gives accuracy of around 98% which is more as compared to Naive Bayes and K-means.The proposed Ent- SVM model in [10] first calculates entropy of raw features like number of source and destination IP address, number of source and destination port number, number of network packets and its type. These entropy values are normalized and are given to SVM for accurate classification of attacking and non- attacking users. KDD and NSL-KDD datasets have been used for experimentation and the proposed method has shown high accuracy and ability to handle large scale networks effectively.

III. METHODOLOGY

A. Pre-Processing of Dataset

In our work, we have defined two modules Pre-Processing module and detection module, Pre-processing of the data is very important step. It makes the available dataset into a format which is feasible for further experimentation and analysis. Our work in this paper, focuses only on detection of DDoS attacks. So, in the first step of pre-processing we have fetched those connections which are DoS. KDD has total 494021 which have different types of attacks like DoS, Probe, R2L and U2R. Out of these total connections, 391458 connections are fetched which have different DoS attacks like back, land, Neptune, smurf, pod, teardrop. Table 1 shows description of different Dos attacks and the number of instances of each category of attack.

Table I: Different types of Dos attacks in KDD

Attack name	**Description**	**Instances**
Back	Large number of TCP packets send through IP spoofing	2023
Land	TCP/SYN packets are send with same destination and source address.	21
Neptune	Large number of SYN packets are send, no session is established	107201
Pod	Ping of death attack (ICMP packets>6400 bytes)	264
Smurf	ICMP echo reply packets are send to broadcast address.	280790
Teardrop	Overlapping of IP fragments causing machines to reboot	979

In the second step of pre-processing, Protocol_type, service and flag are categorical features which are converted to discrete variables by sklearn.preprocessing.LabelEncoder and numpy.unique.

Table 2 shows feature names and the number of categorical features each feature consists.

Table II: Number of categorical values in different features

Feature Name	Categorical values
Protocol Type	3
Service	65
Flag	9

B. Detection Module

After the data is processed, the preprocessed data goes to detection module for DDoS detection. The detection module consists of machine learning classifiers. The classifiers are applied on the processed data to detect Dos attacks. The four classifiers Logistic regression, K-Nearest Neighbor, Naïve Bayes and Artificial Neural Network are used for classifying Dos connections and normal connections. Now the next section describes the machine learning classifier along with their python implementation.

B1. Machine Learning Classifiers

Machine learning is a domain of artificial intelligence in which the machines learn from the behaviour of data being fed into it. The data which is fed is called as training data. After learning the data, it creates a model for detection of attacks. This data is called test data on which the model is tested. Now we describe the various machine learning algorithms and their python functions in this paper.

- **Logistic regression [11]**

Logistic regression is used in predictive analysis. It describes relationship between dependent variable and nominal, ordinal, interval or ratio level independent variable. It is used for binary classification which uses logistic function called as sigmoid function for prediction. The corresponding python function for regression analysis is described below.

All machine learning models in python are implemented as classes. Sklearn. Linear_model class is used for implementing machine learning models. The logistic regression class is imported along with sklearn.linear_model class. Logistic Regression is passed with attributes according to the requirement. Logistic Regression class with its default parameters is stated as:

sklearn.linear_model.LogisticRegression(penalty, dual, tol, C, fit_intercept, intercept_scaling class_weight, random_state, solver, max_iter, multi_class, verbose, warm_start, n_jobs, l1_ratio)

The values and meaning of some important parameters are: Penalty: used for regularization for penalization which can have value either L1or L2, dual**:** boolean value can be either primal formulation or dual formulation, tol: stopping value C: optional float value which determines regularization inverse strength, Fit_intercept: Boolean value to add a constant to decision function, intercept_scaling: optional float value used with liblinear solver, class_weight : optional value for specifying weights of classes, random_state: optional value used as seed for shuffling the data, solver: type of algorithm for optimization, max_iter: optional value to specify maximum number of iterations for algorithm to converge, multi_class: type of class either binary, multinomial or auto, verbose: optional value for verbosity, warm_start: used if previous value is used for initialization, n_jobs: CPU cores to use, l1_ratio: float or none value for elastic net mixing.

- **K-Nearest Neighbour [12]**

KNN is a non-parametric, lazy learning algorithm. KNN Algorithm is based on feature similarity. It determines in what group the data point is to be placed in by looking the data point around it. The data point is placed in a specific group if it is more similar to the datapoints present in it or in terms of minimum distance to the group among the neighbors. The python class for KNN is discussed below.

KNeighborsClassifier class is imported along with sklearn.linear_model class. KNeighborsClassifier is passed with attributes according to the requirement. KNN with its default parameters is stated as:

sklearn.neighbors.KNeighborsClassifier(n_neighbors, weights, algorithm, leaf_size, p, metric, metric_params, n_jobs)

The values and meaning of some important parameters are: n_neighbors: int value to specify number of neighbors, weights: optional value used for prediction, Algorithm: type of algorithm for calculating nearest neighbors, leaf_size: optional int value to specify the leaf size, p: power parameter to specify manhattan_distance or euclidean_distance or minkowski_distance, metric: to specify distance metric for tree, metric_params: additional values for metric parameter, n_jobs: to specify the amount of parallel jobs to run.

- **Naïve Bayes [13]**

Naive Bayes classifiers belongs to the unit of probabilistic classifiers. The classifier is based on simple Bayes' theorem. The classifier assumes that there is no dependency between the features means that presence of a particular feature does not affect the other feature.

Bayes theorem,

$$P\left(\frac{A}{B}\right) = \left(P\left(\frac{B}{A}\right).P(A)\right)/P(B) \qquad (1)$$

With above theorem (1), we can find the probability of occurrence of A given B has already occurred. A is the hypothesis and B is the evidence.

GaussianNB class is imported along with sklearn.linear_model class. GaussianNB is passed with attributes according to the requirement. NB with its default parameters is stated as:

sklearn.naive_bayes.GaussianNB(priors, var_smoothing)

priors: to specify prior probabilities for the class. var_smoothing: addition of largest variance of all features for stability.

- **Artificial Neural Network [14]**

ANN is a very powerful machine learning tool. It is used for finding patterns which are very complex. These systems are human brain inspired and learn the same way as humans learn. This is the reason they have got the name artificial 'neural' network. ANN consists of input and output layers as visible layers for feeding the input and getting the output, while hidden layer is used for finding the patterns for the classification. Keras is a neural network API used for implementing ANN.

Steps followed while implementing:

1. **Define Model**

The set of functions used in Define Model are:
Sequential()- create sequential model
Add()- used to add layers in the model.
Dense (input dimension , number of nodes, activation function, weight initializer)is used to define fully connected layers.

2. **Compile Model**

The steps to be followed in Compile Model are:
Compile (loss function, optimizer, metric)
Loss function- Evaluation of weights
Optimizer- Finding optimized weights
Metric- Reporting results

3. **Fit Model**

Fit () is used to train model on loaded data.
GridSearchCV (estimator, param_grid, scoring, cv = 10) is used for exhaustively running the model over the data for getting optimized results and tuning the hyperparameters.
Figure 1 shows the flowchart depicting the various steps being followed in our work. KDD dataset which includes normal and malicious traffic is fed to the Data Pre-processing module. The pre-processing module fetches only the data which corresponds to DDoS attack and normal attack and removes the data which relates to Remote to Local (R2L), User to Root (U2R), and Probe attacks. The new extracted data set is again processed to convert categorical feature values to discrete values. Then at the last the processed data goes to the Detection module which applies four different classifiers for detecting normal traffic and DDoS traffic.

IV. EXPERIMENTAL SETUP

KDD was generated by capturing raw TCP/IP dump data by simulating a US Air Force LAN for a network. The dataset was downloaded from [15] which is 10 percent of the total. It contains of either attack packets (DoS, Probe, R2L and U2R) or normal packets. Dataset consist of 5 million connection records. There are 41 quantitative and qualitative features with normal and attack data. 41 features are categorized into: basic features and content features of connection, traffic features computed within two seconds time. The dataset is divided into 70% training and 30% testing data.

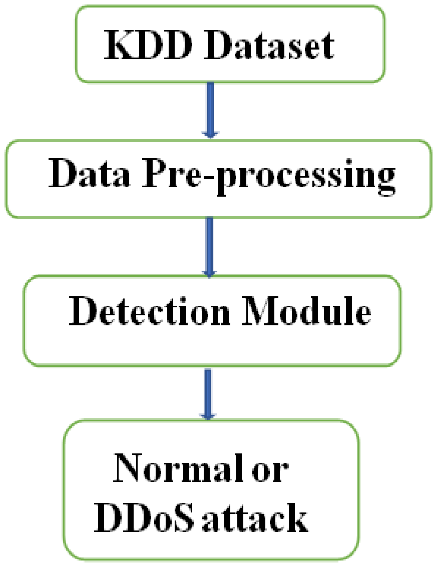

Figure1: Flowchart of Proposed Work

V. RESULTS AND DISCUSSION

The classification results produced by the four algorithms are in the form of detection accuracy. Detection Accuracy is the proportion of instances which are correctly classified to the proportion of total number of instances classified. It is computed using equation below:

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

The python formula for accuracy is:
sklearn. metrics. accuracy_score (y_true, y_pred, normalize, sample_weight)

where, y_true= ground truth or correct labels, y_pred= predicted labels given by classifier, normalize: provides the amount of correct instances if false else fraction of correct instances, sample_weight: to specify number of samples.

The detection accuracy of different classifiers is tabulated. Table 3 represents the detection accuracy of various classifiers which shows that K-NN gives the highest accuracy which is 99.97%.
KNN gives the highest accuracy as it is a well-defined classifier and consistent with any distribution. It converges to the optimal separator with infinite training points accurately.

Table III: Detection accuracy of classifiers

Classifier Name	Accuracy
Logistic regression	99.53%
K-NN	99.97%
Naïve Bayes	99.68%
ANN	99.8%

Figure 2 shows the graph depicting accuracy of the four classifiers i.e. Logistic Regression, KNN, Naïve Bayes and ANN.

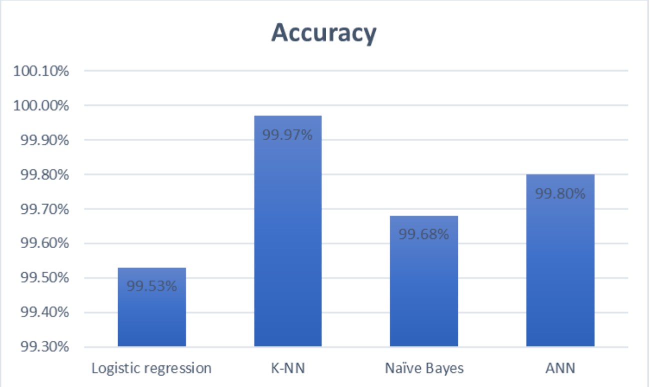

Figure 2: Accuracy comparison of various classifiers

VI. CONCLUSION AND FUTURE WORK

In this work we have used four different classifiers for detecting DDoS attacks. We have applied the four classifiers i.e. Logistic Regression, KNN, Naïve Bayes and ANN on KDD dataset. KDD is a benchmark dataset used in intrusion detection systems. The dataset is first pre-processed then it is fed to classifiers for detection. Experimentation has shown that KNN has performed the best among all the classifiers in terms of detection accuracy due to its consistency with any distribution. In the future we will test these classifiers against other datasets like ISCX, CAIDA, CIDDS to determine attack traffic and normal traffic.

ACKNOWLEDGMENT

This research work is supported by Technical Education Quality Improvement Project III (TEQIP III) of MHRD, Government of India assisted by World Bank under Grant Number P154523 and sanctioned to UIET, Panjab University, Chandigarh (India).

REFERENCES

1. Mohit Kumar. Biggest-Ever DDoS Attack (1.35 Tbs) Hits Github Website, 2018 (accessed July 25, 2018).
2. Lindsey O'Donnell. Wikipedia, World of Warcraft Downed by Weekend DDoS Attacks, 2019 (accessed September 11, 2019).
3. Rajagopalan Vijayasarathy, Serugudi Venkataraman Raghavan, and Balaraman Ravindran. A system approach to network modeling for ddos detection using a naive bayesian classifer. In 2011 Third International Conference on Communication Systems and Networks (COMSNETS 2011), pages 1-10. IEEE, 2011.
4. Chirag N Modi, Dhiren R Patel, Avi Patel, and Rajarajan Muttukrishnan. Bayesian classifer and snort based network intrusion detection system in cloud computing. In Computing Communication & Networking Technologies (ICCCNT), 2012 Third International Conference on, pages 1-7. IEEE, 2012.
5. Zhijiang Chen, Guobin Xu, Vivek Mahalingam, Linqiang Ge, James Nguyen, Wei Yu, and Chao Lu. A cloud computing based network monitoring and threat detection system for critical infrastructures. Big Data Research, 3:10-23, 2016.
6. Adnan Rawashdeh, Mouhammd Alkasassbeh, and Muna Al-Hawawreh. An anomaly-based approach for ddos attack detection in cloud environment. International Journal of Computer Applications in Technology, 57(4):312-324, 2018.
7. Shi-Jinn Horng, Ming-Yang Su, Yuan-Hsin Chen, Tzong-Wann Kao, Rong-Jian Chen, Jui-Lin Lai, and Citra Dwi Perkasa. A novel intrusion detection system based on hierarchical clustering and support vector machines. Expert systems with Applications, 38(1):306-313, 2011.
8. Irom Lalit Meitei, Khundrakpam Johnson Singh, and Tanmay De. Detection of ddos dns amplification attack using classification algorithm. In Proceedings of the International Conference on Informatics and Analytics, page 81. ACM, 2016.
9. Marwane Zekri, Said El Kafhali, Noureddine Aboutabit, and Youssef Saadi. Ddos attack detection using machine learning techniques in cloud computing environments. In 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech), pages 1-7. IEEE, 2017.
10. Chen Yang. Anomaly network traffic detection algorithm based on information entropy measurement under the cloud computing environment. Cluster Computing, pages 1-9, 2018.
11. scikit-learn developers. sklearn.linear_model.LogisticRegression, 2007 (accessed August 12, 2019).
12. scikit-learn developers. sklearn.linear_model.KNeighborsClassifier, 2007 (accessed August 12, 2019).
13. scikit-learn developers. sklearn.linear_model. GaussianNB, 2007 (accessed August 12, 2019).
14. Pushkar Mandot. Build your First Deep Learning Neural Network Model using Keras in Python, 2017 (accessed August 17, 2019).
15. UCI KDD. KDD Cup 1999 Data, 1999 (accessed August 12, 2018).

AUTHORS PROFILE

Aanshi Bhardwaj received her M.E. degree in Information Technology from UIET, Panjab University, India in 2014. She is currently a Ph.D. Research Scholar at UIET, Panjab University, India. Her research interests include web mining, machine learning, and security in cloud computing. She has 5 years of teaching experience in reputed university. She is diligently involved in attending seminars, workshops, conferences of her field.

Veenu Mangat received her M.E. from Punjab Engineering College (PEC) in 2004 and Ph.D. in Engineering and Technology (Computer Science) in 2016 from Panjab University, India. She is currently working as Associate Professor in Information Technology at UIET, Panjab University. She has a teaching experience of more than 15 years. Her areas of research include data mining, machine learning, privacy and security. She has co-authored more than 40 papers in reputed peer-reviewed conferences and journals. She is a Member of the IEEE since 2010.

Renu Vig received her Ph.D. degree in Engineering and Technology in the field of Artificial Intelligence and Neural Networks from Punjab Engineering College in 1997. She is ex-Director, UIET and currently working as Professor of Electronics and Communications Engineering at UIET, Panjab University, India. She has guided more than 12 PhDs and successfully completed several research projects funded by the Government of India and corporate sector. She has published more than 120 research papers in reputed journals and conferences. Her research interests include fuzzy systems, artificial intelligence, neural networks and next generation networking technologies.