# DDOS Attack Detection Using Machine Learning

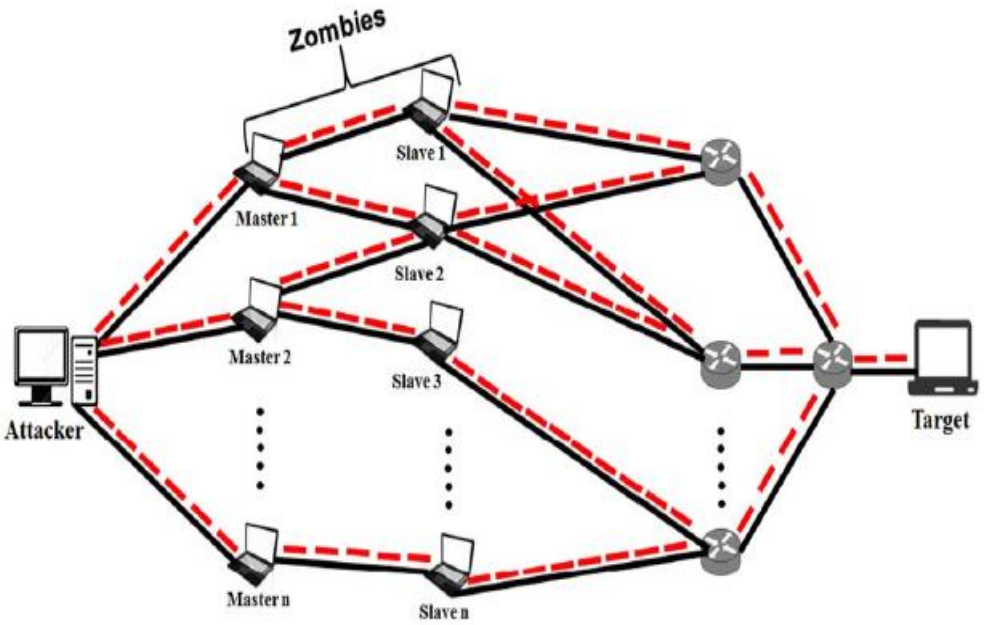
[1]Ashutosh Nath Rimal, [2]Dr. Raja Praveen
[1]M. Tech Cyber Security Student, [2]Assistant Professor
[1]Department of Computer Science and Technology,
[1]Jain (Deemed to be University), Bangalore, India

***Abstract:*** ***Over the last few decades, the Internet has absolutely changed the world. However, the tremendous growth of Internet brought many cyber-attacks. Distributed denial service attack is considered one of the most dangerous attacks. Distributed denial of service is a critical threat that is responsible for halting the normal functionality of services in cloud computing environments. Distributing Denial of Service attacks is categorized in the level of crucial attacks that undermine the network's functionality. These attacks have become sophisticated and continue to grow rapidly, and it has become a challenging task to detect and address these attacks contains.***

***IndexTerms*** **– DDOS Attack, Machine Learning, Cyber Security.**

## I. INTRODUCTION

Over the last few decades, the Internet has absolutely changed the world. However, the tremendous growth of Internet brought many cyber-attacks. Cyber security constitutes one of the most serious threats to the current society, costing hundreds of billions of dollars each year [1]. The first recorded Denial of Service (DoS) attack dates back to the year 1974 with the help of a 13-year-old high school student who had recently heard about a command that could be executed on CERL 'S PLATO. DoS has since risen to become distributed DoS or DoS, and has become infamous for the most destructive cyber-attack type. PLATO was first of its kind computerized shared learning system. The command 'ext' short for external was used to communicate with external devices but if a system was not connected to an external device, then the command 'ext' would force the system to shut down. Denial-of-service (DDoS) attack refers to the utilization of client/server technology to mix multiple computers as an attack platform to launch attacks on one or more targets to extend the facility of the attack [2]. Distributed denial-of-service attack has changed the normal peer-to-peer attack mode, so there's no statistical rule for attack behavior, additionally, common protocols and services are utilized in the attack. it's difficult to differentiate attack or normal behavior only through the kinds of protocols and services. The distributed denial-of-service attack is not easy to detect [3].




As a part of this study a Machine Learning approach is used to predict a DDoS in a network with a maximum accuracy of 99.68%, if the recommended combination of feature selection and classification algorithm is chosen. The user is left with the choice for both feature selection and classification algorithm. In this case the classification algorithm being used are Naïve Bayes and SVM algorithm out of which SVM was found to give maximum accuracy.

## II. LITERATURE REVIEW

This chapter aims to provide detailed information about different DDOS detection techniques and mechanisms in recent times. This chapter includes the various works and studies to support directly or indirectly helps to carry out the present this work. Many researches in the literature focus on DDOS detection techniques separately. Many of these techniques are focused only on specific DDOS attack type. DDOS attack detection can be classified into low rate detection and high rate detection. The description of some of the previous researches that propose different solutions to detect DDOS are described in this chapter.

### 2.1 Low Rate DDOS Attack Detection

Low-rate Distributed Denial-of - Service attacks (low-rate DDoS) represent a new threat to cyberspace, as attackers send a vast amount of similar traffic-like attack packets to bypass legitimate flows. Zhang et al. [4] Proposed a congestion-participation (CPR) metric and a CPR-based approach for detecting and filtering DDoS attacks at low rates.. They found that low-rate DDoS flows actively induce congestion in the network while normal TCP flows actively prevent congestion in the network. The proposed method was conceived to distinguish flows of attack from legitimate flows. However, the testing of its effectiveness requires more experiments and analyzes using real datasets. Du and Abe[6] have proposed an entropy metric for the size of an IP packet to detect both long-term low-rate DDoS attacks and short-term high-rate DDoS attacks. Based on the assumption that many applications have typical packet sizes depending on requests for and responses to data and acknowledgments, they claimed that the distribution of the packet size changes under attacks; this can be used to identify attacks to some degree. However, because the proposed method relies heavily on the packets in the observation window, this approach is constrained in its scalability, and it takes a long detection period to achieve a high probability of detection while suffering from a low-rate DDoS attack.
Jadhav and Patil[5] suggested an efficient, unbiased method of entropy-based detection of DDoS attacks at low levels. This approach is a major improvement over conventional metric entropy. The distance value between regular traffic and attack traffic, however, is very small and therefore the false positive rate is increasing.

### 2.2 High Rate DDOS Attack Detection

A comprehensive research on the protection schemes for the spoofed DDoS attacks has been performed. Each scheme has its advantages and its limitations. Each of them is descripted as follows. A filtering technique for hop-counting is proposed at [7]. The attacker forges TCP / IP header fields for a spoofed DDoS attack to launch. Any TCP / IP header field can however be forged, but its Time-To-Live (TTL) field cannot be forged. Hence, TTL value is used to identify the spoofed IP packets. The challenge in this computation is since the header contains only final TTL value. All Operating Systems (OSs) have different initial TTL value and for a particular IP address its OS may get changed with time. The approach suffers from false positive if the legitimate packet is coming from the unlisted OS and false negative if the attacker correctly predicts the hop-count between source and victim. To differentiate attack traffic from the legitimate traffic. A path fingerprint approach is proposed in [8], where a unique path fingerprint is embedded in each packet. Path fingerprint represents the route traverses by a packet to reach its destination. The incorrect path fingerprint declares the packet as spoofed. The scheme is not able to detect subnet spoofing, as the packet reaches the same subnet and it requires calculation at the intermediate nodes. The TCP/IP header fields' values such as TTL, IP Don't Fragment (DF), window size, and total length are used to identify the OS of a packet in [9]. A fingerprint is created using these values and attached with the packet at the source side. If the fingerprint matches at the receiver side, then the packet is declared as legitimate; otherwise, it is treated as a spoofed packet.

## III. METHODOLOGY

This study uses CCIDS2017 data set in order to train the algorithm and classify DDOS Attacks based on its unique features. The data set contains destination port, different packet characteristics such as packet length, flow duration, header length etc, attack label if the traffic is DDOS traffic and various TCP flags. Features such as packet size, packet length, flow duration, forward packet, backward packet and other various packet attributes are used in order to determine DDOS attack. DDOS attack is an attempt to interrupt a targeted server, service or network's normal traffic by overwhelming the target or its surrounding infrastructure with a flood of Internet traffic. DDOS traffic generally have high flow rate, bigger packet size then normal packet size, more packet length than the normal packet length etc. Dataset is divided into Training and Testing data in order to train the algorithm. 85% of the data set is used in training the algorithm and 15% of the remaining dataset is used for testing. SVM classification algorithm is used in order get an accuracy of 99.68%.

### 3.1 Data Selection

CCIDS2017 data set was used in the study to test and train the algorithm. CICIDS2017 dataset contains benign and the most up-to-date common attacks, which resembles the true real-world data (PCAPs). It also includes network traffic analysis results using CICFlowMeter with time stamp, source and destination IPs, source and destination ports, protocols and attack-based flows [10]. The dataset contains more than 80 network flow features from the generated network traffic.

### 3.2 Data Pre-processing

Data Pre-processing is a technique that is used to transform the raw data into a clean collection of data. In other words, it is obtained in raw format as the data is collected from various sources which is not feasible for the study. The data format must be in a proper manner to achieve better results from the applied model in Machine Learning projects. Any defined Machine Learning model requires information in a defined format, for example, Random Forest algorithm does not accept null values, so null values have to be handled from the original raw data set to execute random forest algorithms. Another aspect is that the data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in a single set of data, and the best out of them is selected.

### 3.3 Feature Selection

Feature Selection is one of the key machine learning principles that greatly impacts the model 's efficiency. The data characteristics you use to train your machine learning models will have a huge impact on the performance you can achieve. Collection of features and Data Cleaning are the most important step in developing your layout. Feature Selection is the process where you select those features that most contribute to your predictive variable or output you are interested in, automatically or manually. With irrelevant features in your data, the model 's accuracy can be reduced and your model learned based on irrelevant features.

### 3.4 Machine Learning

It is a branch of artificial intelligence (AI) that helps to learn from data automatically to improve from experience without explicit programming. Machine learning is more of an advanced AI, which uses statistics, pattern recognition and information discovery and data mining to great advantage. Techniques in machine learning are usually divided into two forms: supervised learning and unsupervised learning. Supervised learning is a data set that includes both the inputs and the desired outputs. Using labeled examples to predict future events, supervised machine learning algorithms can apply what has been learnt in the past to new data. Starting with the study of a established training dataset, the learning algorithm generates an inferred function to make performance values predictions. After appropriate training the program is able to provide expectations for any new data. The learning algorithm will also equate its output with the expected output correctly, and will find errors to adjust the pattern. On the other hand, unsupervised learning takes a set of data that only contains inputs, and finds structure in the data. In comparison, unsupervised algorithms for machine learning are used where the knowledge used to train is neither identified nor named. Unsupervised learning explores how systems can infer a function from unlabeled data to define a hidden structure. The system doesn't find out the correct performance, but explores the data and can draw inferences from datasets to explain hidden structures from unlabeled details. Reinforcement learning means teaching machine learning models to make a range of choices. The agent learns to attain a target in an unpredictable, potentially complex environment. An artificial intelligence in terms of reinforcement learning faces a game like scenario. The computer makes use of trial and error to find a solution to the problem. In this study the proposed strategy is the use of supervised learning to detect DDOS attacks. Two learning algorithms with Naïve Bayes Machine and the Support Vector Machine were investigated and tested for classification of data. These algorithms have been selected based on their effective performance and Network Security implementation.

**i. Naïve Bayes:**

Naïve Bayes is a conditional probability model that demonstrate

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Above condition gives the essential portrayal of the Bayes' hypothesis. Here A, B are two events.

P(A|B): the conditional probability that event A happens, given that B has happened. This is otherwise called the posterior probability.

P(A) and P(B): Probability of A, B without respect of one another.

P(B|A): Conditional probability that event B happens, given that A has happened.

**ii. Support Vector Machine (SVM):**

Support vector machine is supervised algorithm for machine learning that can be used both for regression and classification. In this algorithm, with an estimate of a particular coordinate, system plots every data item as a point in n-dimensional space. After that classification, the hyper-plane that separates the two classes is found. Exceptionally fine. Support Vectors are essentially the co-ordinates of individual observation. Support vectors are essentially the best segregates the two classes. SVM is a standout amongst other realized methods in pattern classification and image classification. It is intended to separate of a lot of preparing pictures two distinct classes, (x1, y1), (x2, y2), …, (xn, yn) where xi in Rd, d-dimensional element space, and yi in {−1, +1}, the class name, with i = 1…n [1]. SVM assembles the ideal isolating hyper planes dependent on a piece work (K). All images, of which feature vector lies on one side of the hyper plane, are have a place with class − 1 and the others are have a place with class + 1.

### 3.5 Proposed Work

CCIDS2017 data set was used in the study to test and train the model. Based on existing strategy classification algorithms Support Vector Machine (SVM) and Naïve Bayes algorithms are being used based on the calculation for the best accuracy result. The model is based on features such as packet flows, port numbers, traffic analysis etc. The data set is divided into training and test data. 85% of the dataset is used for training and 15% of the data set is used for testing.

The detection model is implemented in two stages:

(a) the training stage and

(b) the detection stages.

In the training stage, SVM machine learning calculations are utilized to learn the classifiers. Through the evaluation procedure, the machine learning algorithm that gives the most noteworthy in general classification accuracy will be chosen for use in the proposed detection model. During the detection phase of the model the packet flows, ports etc. are observed and classified in the training stage to decide whether the traffic is DDOS traffic.

## IV. RESULTS AND DISCUSSION

The model is experimented using two classification algorithms Naïve Bayes and SVM in order to predict DDOS attack.

```
In [22]: runfile('C:/Users/Aditya/Desktop/Honeypot/Thesis/naivebais.py', wdir='C:/Users/Aditya/Desktop/
Honeypot/Thesis')
cm is  [[ 82  79]
 [  0 159]]
             precision    recall  f1-score   support

          0       1.00      0.51      0.67       161
          1       0.67      1.00      0.80       159

avg / total       0.84      0.75      0.74       320

0.753125
```

As shown in Fig 6, it can be observed that experimental result for Naïve Bayes Detection has a low accuracy of just 75.31%.

```
F:\Study\Anaconda Tool\lib\site-packages\sklearn\utils\validation.py:578: DataConversionWarning: A column-
vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for
example using ravel().
  y = column_or_1d(y, warn=True)
[[161   1]
 [  0 158]]
99.6875
classification report
             precision    recall  f1-score   support

          0       0.99      1.00      1.00       161
          1       1.00      0.99      1.00       159

avg / total       1.00      1.00      1.00       320
```

As shown in Fig 8, it can be observed that experimental model for SVM has a low accuracy of just 99.6875%.

| Machine Learning Algorithms | True Positive (TR) | False Negative (FN) | False Positive (FP) | True Negative (TN) |
|---|---|---|---|---|
| Naïve Bayes | 82 | 79 | 0 | 159 |
| SVM | 161 | 1 | 0 | 158 |

TP refers to a positive sample that is predicted to be positive, and in this context is normal data predicted to be normal behavior.
TN refers to a negative sample that is predicted to be negative, and in this paper is the attack data predicted to be aggressive.
FP refers to a negative sample that is predicted to be positive, and in this paper is attack data predicted to be normal behavior.
FN refers to a positive sample that is predicted to be negative, and in this paper is normal data predicted to be aggressive.
Precision refers to the ratio of correctly predicted positive observations to the total predicted positive observations, that is, Precision=TP/TP+FP
Recall or also known as sensitivity is the ratio of correctly predicted positive observations to the all observations in actual class, that is, Recall=TP/TP+FN
F1 Score is the weighted average of Precision and Recall, that is,
F1=2*(Precision*Recall)/ (Precision + Recall)
Accuracy refers to the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations, that is AC =TP+TN/(TP+TN+FP+FN)

| Machine Learning Algorithms | Accuracy (%) | F1 Score (%) | Precision (%) | Recall (%) |
|---|---|---|---|---|
| Naïve Bayes | 75.3125 | 74 | 84 | 75 |
| SVM | 99.68 | 100 | 100 | 100 |

From the experimental results given in table, we can see that the Naïve Bayes algorithm produced least accuracy of 75.3125%. It was observed that the classification accuracy among the algorithm has huge difference. The Naive Bayes algorithm provides the lowest overall classification accuracy, but it has the advantages of being simple and low time and computation requirements for training and testing. It can be seen from the above table that SVM resulted in giving higher accuracy of 99.68% in compared to 75.3125 % obtained using Naïve Bayes.

## V. CONCLUSION

The study proposes a DDOS attack detection method using machine learning algorithms and packet analysis. The CCIDS2017 data set is used in the study to detect DDOS attack. Dataset has 80 features out of which 20 were selected. Then the extracted features are used as input features of machine learning, and different algorithm are used to train and obtain the DDoS attack detection model. The algorithm, which were applied to the data set, are Support Vector Machine and Naïve Bayes. The experimental results showed that the SVM algorithm has greater accuracy in compared to Naïve Bayes. Comparing all the results showed that SVM has the highest efficiency of 99.68%.

## REFERENCES

**[1]** Net losses: Estimating the global cost of cybercrime. Technical report, Intel Security Group (previously McAfee, Inc.), 2014.

**[2]** Zargar S T，Joshi J，Tipper D A survey of defense mechanisms against distributed denial of service (DDoS)flooding attacks[J]．IEEE Corn— munications Survey & Tutorials 2013，15(4)：2046—2069．

**[3]** Wang Bing，Zheng Yao，Lou Wenjing，et a1．DDoS attack protection in the era of cloud computing and software—defined networking[J]．Computer Networks，2015，81(4)：308—319．

**[4]** C. Zhang, Z. Cai, W. Chen, X. Luo, and J. Yin, "Flow level detection and filtering of low-rate DDoS," Computer Networks, vol. 56, no. 15, pp. 3417–3431, 2012.

**[5]** P.N.Jadhav and B. M. Patil, "Low-rate DDOS AttackDetection using Optimal Objective Entropy Method," International Journal of Computer Applications, vol. 78, no. 3, pp. 33–38, 2013.

**[6]** P. Du and S. Abe, "IP packet size entropy-based scheme for detection of DoS/DDoS attacks," IEICE Transaction on Information and Systems, vol. E91-D, no. 5, pp. 1274–1281, 2008.

**[7]** H. Wang, C. Jin, and K.G. Shin, "Defense Against Spoofed IP Traffic Using Hop-Count Filtering." IEEE/ACM Transactions on Networking, 15(1), 40-53, Feb. 2007.

**[8]** F.Y. Lee and S. Shieh, "Defending against spoofed DDoS attacks with path fingerprint." International Journal of Computers and Security, Elsevier, 24(7), 571-586, March 2005.

**[9]** O.A. Osanaiye and M. Dlodlo, "TCP/IP header classification for detecting spoofed DDoS attack in Cloud environment." In: IEEE International Conference

**[10]** CCIDS2017 data set, https://www.kaggle.com/cicdataset/cicids2017.