# An Analysis of DDoS Attack in NIDS

Cheng Xu, Tianyi Wei, Jingyi Li, Qin Gao

Information Security Institute, Johns Hopkins University

## Abstract

*As the Internet becomes ever widespread in various industries over the past decades, more and more enterprises and organizations rely on the constant availability of Internet services to operate daily businesses. Consequently, Distributed Denial of Service (DDoS) attacks, bent on overloading domains or specific applications with massive traffic floods, cause even damaging impacts on businesses of all stripes. As Network Intrusion Detection System (INDS) plays the key role to protect a network community, it is urgent to develop such a system to defend against various DDoS attacks. In this project, we developed a network intrusion detection system based on machine learning algorithms to distinguish between benign and malicious network traffic and then detect the DDoS attacks.*

## 1. Introduction

### 1.1 Background

As the rapid development of computer technology with developed information, network attack technology has sprung out nowadays. Many attacks make use of virus cultivation and system invasion to damage the functions as well as resources of machines seriously and to leak data, which results in a tremendous loss for human beings. DDoS is one of the most popular and destructive network attacks. At present, there are many DDoS attacks such as SQL Slammer, CodeRed, and Blaster happening in the world that attracts more and more attention.

### 1.2 Principles

Confidentiality, integrity, and availability are three components of Information Security in the classic model. Each of the components strengthens different aspects of providing protection of information [1]. In DoS(Denial of Service), it aims to the target of availability. This attack makes the target system unable to maintain services by taking advantage of flaws in network service functions of the target system or consuming the system resources directly. Basically, the DoS attack utilizes reasonable service requests to occupy more server resources, which lets legitimate users cannot get available responses of service. However, with the rapid development of Internet technology, it's more and more difficult to carry out the DoS attack because the target system can defend vicious packages from the attack easily. Thus, DDoS finally came out. DDoS takes much more machines to attack the object, which results in a larger scale attack than DoS.

NIDS is used to detect the intrusion in real time within the network by monitoring traffic, in order to prevent attacks in advance by analyzing and intervening network status in time.

## 2. Project Motivation and Current Challenges

### 2.1 Motivation

Due to the convenience of the Internet transaction, a rising volume of institutions and companies have moved their operating models to online services[3]. In order to enhance competitiveness and enlarge influence in the global market, more enterprises require their systems to perform almost 100 percent uptime transactions, which means the online resources are expected to

be available all the time. As a result, once network systems are attacked, the impact is significant.

Another factor that drives us to develop a reliable DDoS detection system is the prevalence of DDoS attacks. They are relatively easy to launch and cheap to launch: with a specific IP address, an adversary can initiate an attack and hackers can even hire botnet services with low prices; besides, the DDoS code is available online for security education, which also gives hackers resources to get armed. All this convenience exposes business and organization to all kinds of DDoS attacks.

## 2.2 Challenges

### 2.2.1.Diverse attack methods and evasion techniques

Adversaries can initiate a DDoS attack through multifaceted methods designed to overwhelm domains or specific application infrastructure with traffic floods.

They can prevent normal access by consuming all network bandwidth. This type of attack includes but not limited to Smurf attack, Ping of Death. They can also exhaust resources available on the target or specific devices between the target and the Internet[10], like SYN floods, ACK floods, and RST attacks. Another way to launch the DDoS attack is to exploit vulnerabilities in the application layers. Examples of such includes but are not limited to HTTP GET floods, HTTP POST floods, DNS application attacks.

What's more, new attacking methods or mauled established methods frequently appear and are used to evade fixed defence mechanisms. Once a single evasion successfully passes through, all of the defense is nullified[10]. Therefore, heuristic approaches are required to defend yet-unknown attacking approaches.

### 2.2.2.The mixture of legitimate and malicious traffic

A tricky problem of filtering traffic is that there are still benign service requests during a DDoS attack and some of them may have significant financial consequences to their businesses. Obviously, no company wants to turn away their customers by accidentally blocking their requests. Therefore, it is crucial and challenging to choose filters. Because if the rules are set too general, the bona fide service requests may be blocked out; if the rules are too specific, malicious requests may enter into the system.

### 2.2.3.A wide range of attack sources

Since in DDoS attacks, traffic can be sent from a wide range of levels and channels, it is infeasible to block malicious packages from every source. In order to detect sudden influx of requests, organizations must be able to route traffic across different levels of network architecture, from level 3 to level 7, from front-end applications to back-end applications, and set up different mechanisms to detect them. This means additional devices and specific knowledge is required to carry out each scheme and configure the whole infrastructure. Based on the large scale of DDoS attacks and limited resources, it is impossible to consume such energy and assets to inspect every aspect of the system. However, if a critical part of a system is overwhelmed by the flood of data, it may be sufficient to ruin the whole business.

### 2.2.4. Uprush traffic

A sudden flood of requests may not always be a DDoS attack. Some websites suffer overloading due to a specific market campaign. For example, on the Black Friday, online shopping websites are likely to be overwhelmed by a huge number of order requests as a large number of customers scramble for low price products. If the server gains a wide range of public attention, it may also suffer overloading situations if countermeasures are not deployed to precaution such circumstances. Obviously, a DDoS detection system is not expected to block the flash packets in such a situation as long as the servers can still handle the load[3].

# 3. DDoS Detection Methods with Machine Learning (Related Work)

## 3.1 DDoS Detection Based on Improved Logistic Regression

Web DDoS is one of the most common attacks. In order to improve the speed and accuracy of detecting Web DDoS efficiently, a new detection algorithm has been put forward, which combines quantum particle swarm optimization method with logistic regression [5]. The algorithm increases the efficiency and accuracy of solving the logistic coefficient.

## 3.2 DDoS Detection Based on Random Forests

To make data flow information as the classification, the source IP, destination IP, destination Port represent the source address, destination address and destination port of data flow separately. The method uses three information entropies such as SIDI, SIDP, and DPDI to represent three kinds of many-to-one features, and

analyzes the features of TCP attack, UDP attack as well as ICMP attack. Also, it detects the classification of three kinds of DDoS attacks using the classification model based on random forests. Finally, the model can classify network traffic flow from the normal state and vulnerable state accurately.

### 3.3 DDoS Detection Based on Network Traffic and IP Entropy

According to the features of network traffic and IP entropy during the DDoS attack, the detection is put forward. First, we need to judge whether the network traffic is normal, and then to identify if there is a DDoS attack. The method can detect the DDoS attack accurately with high efficiency because we can not detect the DDoS attack relying solely on neither network traffic or IP entropy [9].

## 4. Experiments

### 4.1 Dataset Description

For this experiment, we used a modified and labeled dataset provided by [2]. There are two reasons why we adjusted the original dataset.

The first reason is that the whole dataset contains about more than 570,000 Network traffic streams, all of which being labeled either as “normal” or “attack”. However, only 37460 files are labeled as “attack”, which occupy only 6% of the entire dataset. So, if we use the original dataset to train our model, the model would show a bias in determining whether a packet is “normal” or “attack” that very likely, it’ll judge a packet to be “normal” to get a high prediction accuracy. However, the accuracy is meaningless.

The second reason is that it’ll take us too much time to use the entire dataset to train a model, like SVM or Neural Networks, since our laptops couldn’t provide such huge computational power.

So, to tackle those problems, we use the Under-sampling method to modify the original dataset. Instead of using the whole dataset as the input, we decided to use only a part of the data in order to obtain a smaller and well-balanced dataset. To achieve this, we first extracted all files labeled as “attack” and then randomly extracted some files labeled as “normal” of the same size as the “attack” files. The total size of the dataset used for training and testing is 70,000, which consists of the “attack” data and “normal” data proportionally.

Based on the observation of the original dataset, we removed several features that are unnecessary in this experiment. For example, the “startDateTime” and “stopDateTime”. Although the DDoS attacks usually appear during a specific period of time, we couldn’t use a previous time to predict the time in the future. In addition, we used LabelEncoder to change some other features from string to float format to fit the requirement of our training models, such as “TCPFlags”, “sourcePayload” and “destinationPayload”.

### 4.2 Software

We used Jupyter Notebook to implement the whole experiment. We applied four training models of Scikit-learn library to our dataset. The training models will be described in detail in the later sections.

### 4.3 Hardware

The experiments were run on two laptops with the Intel Core i7 3.1 GHz processor with 16GB of memory and Intel Core i5 2.3 GHz processor with 8GB of memory separately.

### 4.4 Approach

A series of models were applied in this experiment. They are introduced briefly as follows.

#### 4.4.1. Logistic Regression

Logistic Regression is used to deal with the problem of a dependent variable for classification. There are only two values for the dependent variable with labeled “0” and “1”. However, the independent variables can be continuous or binary. According to the analysis using logistic regression, the weight of independent variables will come out, so that it can find what factors result in the dependent variable with labeled “0” and “1”. At the same time, it can predict the probability of the dependent variable with labeled “0” and “1” due to the weight of independent variables.

#### 4.4.2. Random Forest

Random Forest is a classifier to train samples and predict results taking advantage of multiple decision trees. The class of outputs depends on the mode of the classification of the individual trees. The basic unit of the Random Forest is the decision tree, and its essence is ensemble learning which belongs to machine learning.

#### 4.4.3. K-Nearest Neighbours

K-Nearest Neighbours algorithm (k-NN) is one of the simplest classification algorithms. It is a non-parametric, lazy machine learning algorithm for classification and regression[6]. Its purpose is to use a database in which the data points are separated into several classes to predict the classification of a new sample point[7].

### 4.4.4. Neural Network

Neural Networks is a machine learning approach inspired by the way in which the brain performs a particular learning task. The learning procedure occurs by repeatedly activating and reinforcing certain neural connections over others. Different from other supervised algorithms, neural networks provide feedback for the result: when the desired outcome occurs, the neural connections producing that outcome become strengthened.

### 4.4.5. Work Flow

To ensure that we obtain the statistic analysis results, we used cross-validation method train and test four models. In addition, we also implemented the grid search method to[1] automatically adjust parameters to get their optimal value. However, since to apply grid search on the K-Nearest Neighbours and Neural Networks is so time-consuming, we only applied grid search on the first two algorithms. We obtained two prediction accuracy rates before and also after parameter optimization. By comparison, the latter one is better than the former. Not only did we calculate our model's accuracy rate, but also we generated ROC (Receiver Operating Characteristics) curve and the corresponding AUC (Area Under the Curves) to intuitively evaluate the performance of our models. The result showed that by adjusting parameters, the performance of each model could be further improved. Visually, we used ROC to present the change of accuracy.

1) Logistic Regression

In logistic regression, we first used the default value for each parameter of the model to get a prediction accuracy rate that is 0.984557. After optimizing two parameters----attributing '10.0' to 'C', which means the inverse of regularization strength, and 'l1' to 'penalty', which is used to specify the norm used in the penalization, we got an accuracy rate of 0.991791. It seems that only a little improvement was made than before. However, since the original rate was quite high, there was barely space left for evident improvement. So parameter adjustment was worthy and the prediction result was good.

To evaluate the performance of our adjusted logistic regression model, we first drew a ROC curve and calculated its corresponding AUC. The ROC curve almost coincided with the top and left boundary lines of the graph. Similarly, the AUC was 0.997819. Those two evaluation metrics indicated that our model was optimal and the prediction result was pretty good.

2) Random Forest

In the random forest, we first used the default value for each parameter of the model to get a prediction accuracy rate that is 0.999800. Then we changed the value of four parameters by attributing 'gini' to 'criterion', which measures the quality of a split, 'None' to 'max_depth', which means the maximum depth of the tree, 'auto' to 'max_features', which represents the number of features to consider when looking for the best split, and '200' to 'n_estimators', which is the number of trees in the forest. And we got an accuracy rate of 0.999853. The improvement is quite obvious considering that the original accuracy is already very high. As a result, parameter adjustment is well worthy and the prediction result is good.

To evaluate the performance of our adjusted random forest model, we first drew a ROC curve and calculated its corresponding AUC. Similarly, the ROC curve nearly coincided with the top and left boundary lines of the graph. Likewise, the AUC was perfect----1.0. Those two evaluation metrics indicated our model was optimal and the performance in prediction is excellent.

## 5. Conclusion

| Model | Accuracy (without cross-validation) | Sensitivity | Specificity | False Positive Rate | Precision | Time (1-shortest 4-longest) | Accuracy (with cross-validation) | AUC |
|---|---|---|---|---|---|---|---|---|
| Logistic Regression | 0.991725 | 0.985341 | 0.998128 | 0.001872 | 0.998110 | 2 | 0.991791 | 0.997819 |
| Random For | 1.0 | 1.0 | 1.0 | 0.0 | 1.0 | 1 | 0.999853 | 1.0 |

[1] Malik, U. (2018, July 26). Cross-Validation and Grid Search for Model Selection in Python. Retrieved from https://stackabuse.com/cross-validation-and-grid-search- for-model-selection-in-python/

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| est | | | | | | | | |
| K-NN | 0.997864 | 0.996802 | 0.998930 | 0.001070 | 0.998932 | 3 | 0.997517 | 0.999191 |
| Neural Network (Perceptron) | 0.990256 | 0.985341 | 0.995187 | 0.004813 | 0.995155 | 4 | 0.988441 | 0.992798 |

In this project, we successfully applied different machine learning algorithms to DDoS detection and proved them to be effective measures. After adjusting the parameters of each classifier, we found that all of them have a convincing performance, with an accuracy of over 99.4 percent. Random forest classifier has the highest accuracy, 0.999853.

In addition, we used AUC to evaluate each classifier, which also indicates that all of them are good enough to be selected as the classifier.

Furthermore, we also evaluated each attributes to find out their meanings and usages respectively. We used information gain as our metric. Information gain can directly reflect the contribution of a certain feature to a classifier. Features with a bigger information gain can be considered more “useful” for a classifier. We calculated the information gain of each feature, and generate a histogram to visualize the result.

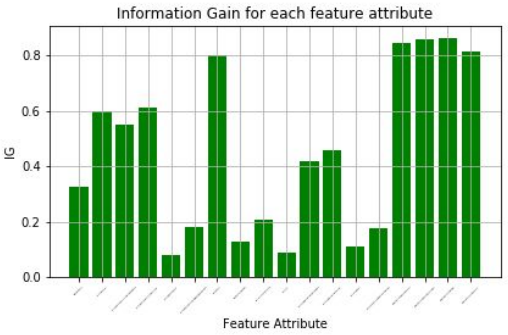


Figure 1. Information Gain for Each Feature Attribute

In addition, we considered the relevance between each feature, as features with high relevance should not be selected together. We generated the correlation matrix and used the heat map to visualize the matrix.

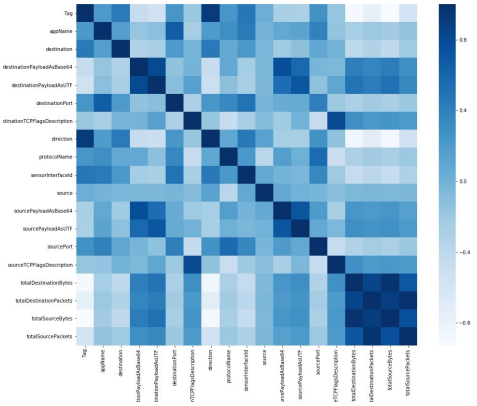

Figure 2. Correlation of Attributes

## 6. References


[1]Karen Scarfone, Wayne Jansen, and Miles Tracy. 2008. NIST Special Publication 800-123, Guide to General Server Security.

[2] Search UNB. (n.d.). Retrieved from https://www.unb.ca/cic/datasets/index.html

[3]Andrew Braunberg, Mike Spanbauer. Retrieved from https://www.nsslabs.com/research-advisory/library/infrastructure-security/distributed-denial-of-service-prevention/why-is-ddos-prevention-a-challenge/why-is-ddos-prevention-a-challenge/

[4]R. Thomas, B. Mark, T.Johnson, and J. Croall, "Netbouncer: client-legitimacy-based high-performance ddos filtering," in DARPA Information Survivability Conference and Exposition, 2003. Proceedings, vol. 1. IEEE Press,2003,pp.14-15.

[5]ZHANG Xuebo, LIU Jinghao, FU Xiaomei. Design and Implementation of Anti Web DDoS Attack Model Based on Improved Logistic Regression Algorithm[J]. Netinfo.Security, 2017(6):62-57.

[6]Retrieved from https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

[7]Retrieved from https://medium.com/@adi.bronshtein/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7

[8]Yu Pengcheng, Qi Yong, Li Qianmu. 2017. Application Research of Computers 10(34). DDoS attack detection method based on random forest.

[9]Yang Jungang, Wang Xintong, Liu Guqing. 2016. Application Research of Computers 4(33). DDoS attack detection method based on network traffic and IP entropy.

[10]Retrieved from https://www.nsslabs.com/research-advisory/library/infrastructure-security/distributed-denial-of-service-prevention/why-is-ddos-prevention-a-challenge/why-is-ddos-prevention-a-challenge/