World Academy of Science, Engineering and Technology
International Journal of Computer and Systems Engineering
Vol:8, No:10, 2014

# Predicting Application Layer DDoS Attacks Using Machine Learning Algorithms

S. Umarani, D. Sharmila

***Abstract***—A Distributed Denial of Service (DDoS) attack is a major threat to cyber security. It originates from the network layer or the application layer of compromised/attacker systems which are connected to the network. The impact of this attack ranges from the simple inconvenience to use a particular service to causing major failures at the targeted server. When there is heavy traffic flow to a target server, it is necessary to classify the legitimate access and attacks. In this paper, a novel method is proposed to detect DDoS attacks from the traces of traffic flow. An access matrix is created from the traces. As the access matrix is multi dimensional, Principle Component Analysis (PCA) is used to reduce the attributes used for detection. Two classifiers Naive Bayes and K-Nearest neighborhood are used to classify the traffic as normal or abnormal. The performance of the classifier with PCA selected attributes and actual attributes of access matrix is compared by the detection rate and False Positive Rate (FPR).

***Keywords—***Distributed Denial of Service (DDoS) attack, Application layer DDoS, DDoS Detection, K- Nearest neighborhood classifier, Naive Bayes Classifier, Principle Component Analysis.

## I. INTRODUCTION

A Distributed Denial of Service (DDoS) attack is a major threat in the internet. The attackers use this to disturb a service or a computing resource by using a coordinated effort. This effort can be made by compromising multiple numbers of computers to send a flood of traffic created intentionally for overwhelming their servers or fully utilizing their bandwidth. The main motive of DDoS attack is to disturb or make unavailable the normal service provided by a site for both internal and external users [1]. The simplest way of DDoS attack occurs, when individual users work together and reload/refresh a website continuously in their browser like Internet Explorer or Chrome. In a genuine case if a large number of users are accessing the same web site at the same time, then this can be referred as flash crowd. Presence of the flash crowd and the access patterns by this crowd can be used to determine the DDoS attack.

Recently a survey was taken in North America, Europe and Asia in 36 tier 1, tier 2 and hybrid IP network operators [3]. The survey reported that DDoS attack is the foremost among all the attacks in the large networks and around 64% of them affected the operational security in most popular servers. At the initial stage, the attacker attempts to compromise the computers connected via the Internet to make it malicious systems and then install the tools needed to attack in these compromised systems. The compromised computers are referred as “zombies”. During the next stage, the zombies receive the attack command from the attacker system via a secure channel for commencing a bandwidth attack against the target or victims server [2]. Then the attack traffic will be transmitted from the “zombies” to the blameless third-parties. Either the own or spoofed source IP addresses will be used in the attack traffic. There are two major reasons to use the spoofed IP addresses in the attack traffic. First reason is to conceal the uniqueness of the “zombies” and minimize the danger of being traced the actual attacker from these “zombies” and second is to make it very difficult or impossible to strain attack traffic by not disturbing the legitimate traffic.

DDoS attacks can be activated from a great pool of compromised systems used in homes, educational, industrial and government organizations. These compromised systems are called as bots. These bots can be connected with a remote Internet Relay Chat (IRC) server automatically for controlling remotely by the attacker from these botnets [6], [7]. Along with DDoS attacks, the Botnets may also be utilized for generating spam emails, viruses and worms. The existing design and communication patterns used in the internet also enable the DDoS attack easier. As the routers are implemented for providing increased throughput, high bandwidth pathways must be used as the intermediate network. In contradiction, the end hosts might be given as much bandwidth as they require to use for own applications and all the end hosts have less bandwidth when compared to the routers. Therefore, the attackers can use wrongly these plentiful resources in these routers to transfer/deliver large number of packets to a target server.

Based on the nature of DDoS attacks, they are classified into two types. They are end-point attacks and infrastructure attacks. In case of end-point attack, the target is an individual end-host or an entire customer sub-network which is served by an Internet Service Provider (ISP). In the infrastructure based attack, attack packets in large number are sent via a port of an ISP router to generate one or more choke-points within the ISP infrastructure by using the knowledge of the routing pattern within the domain**.** Traditionally DOS attacks are started from the network layer by flooding UDP, SYN or ICMP messages which is called as NET-DDoS. If these attacks fail, then the attacker may move to the application layer and flood HTTP GET messages and these messages are called as application layer DDoS. The attacker may send a large number of queries to a database from victim computers

S. Umarani is with the Maharaja Engineering College, Avinashi, Tamil Nadu, India – 641654 (e-mail: umashenna@ gmail.com).

Dr. D. Sharmila is with the Bannari Amman Institue of Technology, Sathyamangalam, Tamil Nadu, India – 638402 (e-mail: sharmiramesh@rediffmail.com).

or may try to download a large number of images to make the server down [4].

The impact of the attack varies from a simple inconvenience to the user to major failures at the server. Based on the impact of a DDoS attack at target, the attacks can be categorized as disruptive and degrading attacks. The main aim of the disruptive attacks is to entirely cut out or break down the service of the victim. For instance, the network of the target is totally congested because of the attack, or the victim server breaks down/halts under the attack. Therefore, the users/clients cannot use/access the service further. The motivation of degrading attacks is to munch through some part of the resources of victim's resources to downgrade performance of the service provided by the victim. For example, an attack may transmit huge number of authentication requests intentionally, to utilize computing resource of the target server. This will increase the response time for the legitimate users. So the legitimate user may get dissatisfaction with the Quality of Service and may change the service provider [5].

Many number of filtering methods is used to identify the legitimate access and the flooding created by the attackers. To avoid these types of DoS attacks, the upstream to the target server must be categorized and the flooding created by the attacker must be stopped by pushback mechanism. Without proper characterization, both the legitimate traffic and the suspicious flows will be affected equally.

Many methods used for detecting and responding to DDoS attacks use the attack traces for finding the attack signatures. But getting the actual attack traces is harder especially in the newer attacks because many organizations which are mostly attacked will not give the data monitored. The monitored details may have very sensitive information or these organizations may not want to publicly admit to being attacked as this can damage their reputation. By studying the behavior of attacker by means of the types of attacks they can create, how they create data for an attack message, the target port addresses these attacks by generating legitimate or spoofed source addresses which could be used to formulate attack signatures and anomaly detection algorithms [8]. While there has been recent work by the data-mining research community to recognize intrusion patterns using offline machine-learning approaches.

Defences to the DDoS attacks can be created in many ways. The development of intrusion detection systems (IDSs), firewalls and security enhanced routers will guard against the attack traffic. Defences at the place of the victim can be implemented by monitoring the measures of hop count and Time To Live (TTL) for messages arrived at the victim. In the transit networks, the routers can monitor collaboratively when exchanging certain information to guard against DDoS attacks. Pushback mechanism is a method used to stop attack traffic flow. In this method each host computes the limit for each contributing neighbor by sending a pushback request message. The recipients start rate limiting the aggregate in the specified limit. This pushback saves the upstream bandwidth by using the early dropping of packets and useful for focusing the rate-limiting on the attack traffic within the aggregate.

The characteristics of the traffic in the low layers are not sufficient to differentiate the App-DDoS attacks when the normal flash crowd event occurs. Therefore different approach should be adapted to detect whether the huge traffic is created by the App-DDoS attackers or by legitimate web users. In this paper, a novel method is proposed to detect the denial of service attacks from the HTTP traces. From the traces an access matrix is generated, useful attributes are selected and used in the classifier to detect the attack.

The rest of this paper is organized as follows. In Section II, prior research work related to DDoS attacks are discussed. Section III explains the algorithms/methods used in this work and Section IV describes the experiments conducted by using the 1998 world cup website data set and the results obtained. Section V concludes this paper.

## II. Related Works

As many DDoS attacks were done by using open-source bot-based powerful tools and many organizations did not want to share the traces, [8] studied the source code used in the popular DDoS attack bots, Agobot, SDBot, RBot and Spybot to get the complete knowledge about the DDoS attacks which might be useful for designing efficient approaches for the detection and mitigation of DDoS attacks. Nazario [9] examined the history of the DDoS attacks, various designs used previously to reduce the punitive damage in the victim and the tools used to compromise the servers were studied. The attacks were measured based on the backbone traffic of the internet, activities in the botnets and changes in the routing. Based on the analysis, the author concluded that in many attacks, the attackers were acted in non-state, and they were able to use botnet population efficiently to begin massive DoS attacks.

Alomari et al. [10] studied the danger of Botnet-based DDoS attacks originating at the application layer because these attacks created the revenue losses for many business sites and government web sites. The possible solutions and the research directions for the future to resolve application layer DDoS attacks were discussed.

Pushback is one of the widely used methods to defend against DDoS attacks and uses the concept of the congestion-control problem. The detection and selective drop are the major steps used in this approach. Kumarasamy and Asokan [11] used puzzle solving mechanism to push back to the core routers rather than sending to the target server. In order to avoid the attackers, the victim server gave a puzzle to the client that sent the traffic. If the client is able to solve the puzzle, then client was considered as authentic, and traffic from the client was allowed into the server. If the puzzle solving client was suspected as malicious, then the target gave complicated puzzle. The combined puzzle sending and push back mechanism was used by the authors to identify the DDoS attack. As the DDOS is a coordinated attack which is done on a massive scale, a continuous evaluation of new attacks and the number of vulnerable hosts on the Internet must be monitored. Bhuyan et al. [12] and Chiueh [17] surveyed the types of DDoS attacks, methods used for detection, and tools used in wired networks. The issues, challenges and possible solutions were analyzed.

Gu and Liu [13] reviewed existing DoS attacks and key defense technologies in wireless networks and described network based and host based DoS attack techniques to illustrate attack principles. DoS attacks were classified according to their major attack characteristics. Current counterattack technologies were also reviewed. DoS attacks and defenses in 802.11 based wireless networks were explored at physical, MAC and network layers.

As most standard applications use the well-known port numbers and ports are listening in a long time for acknowledgements makes the DDoS attacks easier. Pseudo-random port-hopping and synchronization between communicating parties are needed to reduce the attacks, but time servers for synchronizing clocks could become targets to DoS attack themselves. Fu et al. [14] proposed an algorithm called as BIGWHEEL for the servers. These servers communicated with multiple clients in a port-hopping manner which also made to support multi-party applications. In this algorithm, the server did not use a fixed port before judging the client. An algorithm called as HOPERAA was used to solve the problem of clock drift. Based on the type and interaction of the client, the acknowledgement and time server was used. The proposed method avoids eavesdropping adversary attack directed to the application's open ports.

Zargar et al. [15] analyzed the scope of the DDoS flooding attacks and categorized the attacks and available countermeasures based on where and when these algorithms could prevent, detect, and respond to the DDoS flooding attacks. Yau et al. [16] proposed a new method by using level-max-min fairness. A control-theoretic model was used for finding the convergence of algorithm based on different system parameters. Different models to represent a good user and attacker were used to explain the distributions and behavior. The study concluded that the throttle mechanism is highly effective for aggressive attackers in dropping attacker traffic over good user traffic. The level- max-min fairness gave better good-user protection than techniques proposed in the literature. Simulations were conducted, and the results proved that router throttling had low deployment overhead in time and memory.

Mirkovic et al. [18] proposed a defense system called as DefCOM. Selected nodes in the network called DefCOM nodes are situated with some distance from the source, victim and core networks and operate as an overlay for finding and cut the attacks. These defense nodes strained the attack traffic and protected the resources of the victim, and also helped to identify the legitimate traffic along the suspicious stream and make its correct delivery to the victim. DefCOM also defined a framework to the security systems to join the overlay and cooperated in the defense.

Moore et al. [19] presented a new method called backscatter analysis to estimate the activities of denial-of service. Experiments were conducted with data traces of three week-long to get the number, duration and focus of attacks, and to characterize their behavior. From the experiments, the authors identified more than 12,000 attacks against more than 5,000 distinct targets, ranging from well-known ecommerce companies like Amazon and Hotmail to small foreign ISPs and dial-up connections. Weiler [20] presented a honeypot to defense a network from the DDoS attacks. The potential activities of the attacker are simulated, and the passive defense mechanisms were analyzed.

## III. Methodology

In this work, the HTTP traces from the 1998 world cup website was used to detect the DDoS attacks. From these traces, the access matrix is created to represent the spatial and temporal patterns of access and two classifiers Naive Bayes and KNN are used to find the attacks.

### A. Dataset

Many traces are given in this data set. For this work, the requests sent to the 1998 World Cup Web site for the duration from April 30, 1998 to July 26, 1998 are taken. In this duration of time, this site received 1,352,804,107 requests [21]. The access logs in the 1998 World Cup Web site were used in the Common Log Format. It is converted to binary format to reduce the size and analysis time. All entries in the binary log are of fixed size and show one request to the site. The binary log format as follows:

```
struct request
    {
    uint32_t timestamp;
    uint32_t clientID;
    uint32_t objectID;
    uint32_t size;
    uint8_t method;
    uint8_t status;
    uint8_t type;
    uint8_t server;
    };
```

The fields used in the request structure are:

*timestamp* - the time of the request and is stored as the number of seconds since the Epoch. The timestamp information is converted to GMT for portability. For calculating the local time, each timestamp must be adjusted.

*clientID* - For each client, an unique integer identifier was given and because of some of the privacy concerns these mappings were not released;

*objectID* - a unique integer identifier for the requested URL

*size* - the number of bytes in the response

*method* - the method contained in the client's request

*status* - this field contains two pieces of information; the 2 highest order bits contain the HTTP version indicated in the client's request and the remaining 6 bits indicate the response status code

*type* - the type of file requested

*server* - gives the details of which server handled the request.

During the log collection period (April 30th, 1998 to July 26th, 1998), 33 different World Cup HTTP servers were used at four geographic locations such as Paris, France; Plano, Texas; Herndon, Virginia; and Santa Clara, California. Total of 1,352,804,107 requests were received by the World Cup site.

### B. Access Matrix Creation

Access Matrix (AM) is used to represent the spatial and temporal patterns of access of the particular site or web document at a particular instant of time. The popularity of website depends on the hit rate. The hit rate can be defined as,

$$P_{it} = \frac{b_{it}}{\sum_{i=1}^{N} b_{it}} \tag{1}$$

Here, $b_{it}$ represents the number of requests to a document i in a web server at the time unit t, and N represents the total number of documents in the entire web server. If the number of users requesting the document i at the time unit t is represented as $c_{it}$, then the average revisit of the user for the document i is calculated by $\frac{b_{it}}{c_{it}}$. The normalized revisit of the user is represented as $r_{it}$ and the number of observation time units is represented as T.

$$r_{it} = \frac{\text{average request number per user on the } i^{th} \text{ document at } t^{th} \text{ time unit}}{\text{average request amount per user at } t^{th} \text{ time unit}} \tag{2}$$

$$= \frac{b_{it} / c_{it}}{\sum_{i=1}^{N} \left( b_{it} / c_{it} \right)},\ i \in [1, N],\ t \in [1, T] \tag{3}$$

Then the AM of the dimension N x T can be constructed [22] by,

$$A_{N*T} = \left[ \bar{a}_1, \bar{a}_2, ..., \bar{a}_T \right] = \left[ a_1, a_2, ..., a_N \right]^T \tag{4}$$

where,

$\bar{a}_t = \left( a_{1t}, ...., a_{Nt} \right)^T$, $a_i = (a_{it}, \ldots, a_{iT})^T$ and $a_{it} = p_{it}\ or\ r_{it}$

The AM represents the spatio temporal pattern of access of a document i. $\bar{a}_t = \left( a_{1t}, ...., a_{Nt} \right)^T$ represents the spatial distribution of popularity at the time unit t and $a_i$ represents the usage of the document i during in the varying time unit $a_t$ and related to the interest of the user. The value of the $a_i$ depends on the number of clicks and the browsing time of the user.

### C. Feature Reduction Using PCA

PCA is also known as the Karhunen–Loeve transform. This transform is used for reducing the dimensionality of the data for analysis and compression [23]. In this transform, huge number of variables is replaced by a small number of variables which are uncorrelated. Orthogonal linear combinations of actual variables with highest variance were found. The transformation used in PCA is discussed in the following.

From the samples, the average vector is calculated as,

$$\bar{\mu} = \frac{1}{T} \sum_{t=1}^{T} \bar{a}_t \tag{5}$$

where T is the total number of samples in the data set, $\bar{a}_t = \left( a_{1t}, ...., a_{Nt} \right)^T$ is the sample t, and $a_{it}$ is the popularity of document as defined in T . The deviation from the average is defined as

$$\bar{\phi}_t = \bar{a}_t - \bar{\mu} \tag{6}$$

The sample covariance matrix of the data set is defined as

$$C = \frac{1}{T} \sum_{t=1}^{T} \left( \bar{a}_t - \bar{\mu} \right) \left( \bar{a}_t - \bar{\mu} \right)^T \tag{7}$$

$$= \frac{1}{T} \sum_{t=1}^{T} \bar{\phi}_t \bar{\phi}_t^T = \frac{1}{T} \Phi \Phi^T$$

where

$$\Phi = \left[ \bar{\phi}_1, \bar{\phi}_2, ..., \bar{\phi}_T \right].$$

When using PCA to minimize the high dimensional data, the eigenvalues and corresponding eigenvectors from the covariance matrix C calculated from the samples are used in the singular value decomposition (SVD). Let $(\lambda_1, u_1), ...., (\lambda_N, u_N)$ represent the pairs of eigenvalue–eigenvector from the sample covariance matrix C. Usually, largest eigenvectors are selected that has large K eigenvalues and the (N-K) dimensions generally contain noise. K can be determined by

$$\sum_{i=1}^{K} \lambda_1 / \sum_{i=1}^{N} \lambda_1 \geq \alpha \tag{8}$$

where α is the contribution ratio of variation in the subspace to the total variation in the original space. The calculation of matrix of U with size T x K is done. The columns have the K eigenvectors. The K-dimensional subspace is represents data of principal components based on the following rules:

$$\bar{d}_t = U^T \left( \bar{a}_t - \bar{\mu} \right) = U^T \bar{\phi}_t,\ \text{t=1,..,T} \tag{9}$$

### D. Naïve Bayes Classifier

A Naive Bayes classifier is a classifier based on a probabilistic model and uses Bayes' theorem with strong (naive) independence assumptions [24]. This classifier could be trained efficiently using supervised learning and the parameters used for Naive Bayes models can be estimated by using the method of maximum likelihood. The major benefit of using Naive Bayes classifier is, a small amount of training data is enough to estimate the parameters (means and variances of the variables) needed for the classification as independent variables are assumed.

The probability model used for a classifier uses the conditional model represented as P(C/$F_1$,…...,$F_n$) over a dependent class variable C *(classes)*, conditional on several feature variables ($F_1$,…,$F_n$). Using Bayes' theorem, the conditional probability can be written as,

$$P(C/F_1,\ldots,F_n)=\frac{P(C)P(F_1,\ldots,F_n/C)}{P(F_1,\ldots,F_n)} \qquad (10)$$

Otherwise, it can be written as,

$$posterior=\frac{prior\ x\ liklihood}{evidence} \qquad (11)$$

As the denominator part is not depending on and the values of the features $F_i$ are given, the denominator is effectively constant. The numerator is equivalent to the joint probability model and it can be rewritten as,

$$P(C,F_1,\ldots F_n)=P(C)P(F_1/C)P(F_2/C)P(F_3/C)\ldots.. \qquad (12)$$

The above formula is equivalent to the following,

$$P(C,F_1,\ldots F_n)=\prod_{i=1}^{n}P(F_i/C) \qquad (13)$$

### *E. KNN Classifier*

In the nearest neighborhood classifier, the training set is represented as, ($X_1$,$Y_1$), ($X_2$,$Y_2$), …., (Xn,Yn), each $X_i$ represents the input and each $Y_i$ represents the class label in the set {1,…C} where the total number of classes are C. Let $X_i$=($Xi^{(1)}$, $Xi^{(2)}$, …, $Xi^{(d)}$) is the d dimensional input vector, then this classifier finds $Y_{new}$ in the {1,..C} for the given $X_{new}$. If the K value is 1, then find the closest point $X_j$ to $X_{new}$ with respect to the Euclidean distance and classify $X_{new}$ as $Y_j$. This 1-NN classifier is called as instant classifier [25].

The Euclidean distance is represented as,

$$dist(X_j,X_{new})=\sqrt{\sum_{i=1}^{d}(X_j^{\ d}-X_{new}^{\ d})^2} \qquad (14)$$

In KNN classifier, find the k closest training points to $X_{new}$ with respect to the Euclidean distance and classify by $Y_{new}$ as majority vote among the k closest points.

## IV. Result and Discussions

The experiments are conducted with 1998 World Cup website dataset and the results of classifier are accessed by the Detection rate and the False positive rate. Two different classifiers with the combination of actual attributes and PCA reduced attributes are compared. Table I shows the numeric results and Figs. 1 and 2 show the results graphically.

TABLE I
Detection Rate And False Positive Rate Of Proposed Learning Algorithms

| Classifier | Detection Rate | False Positive Rate |
|---|---|---|
| Naïve Bayes Model | 0.9547 | 0.1984 |
| KNN | 0.9303 | 0.1742 |
| Naïve Bayes with PCA | 0.9595 | 0.2014 |
| KNN with PCA | 0.9425 | 0.1865 |

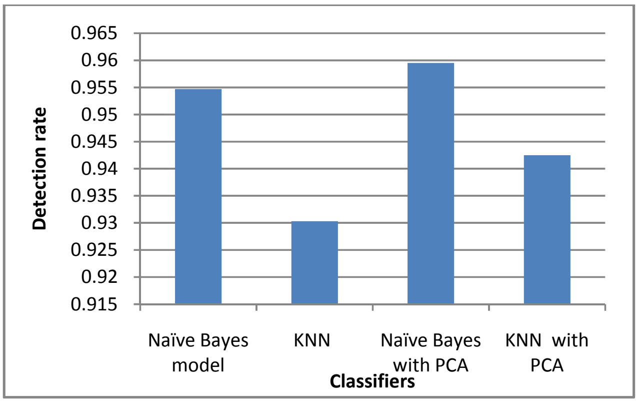

Fig. 1 Detection rate

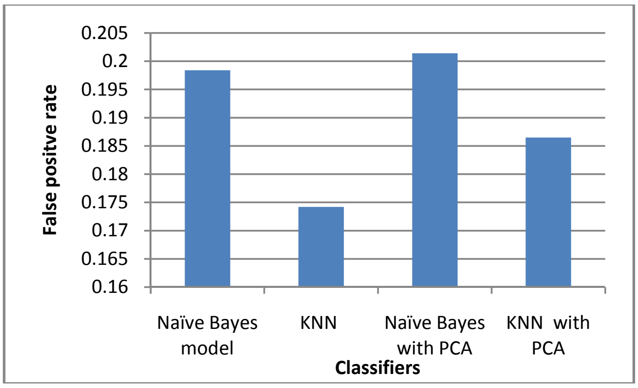

Fig. 2 False Positive Rate

From Fig. 1, it is observed that the detection rate of Naive Bayes classifier improved by 2.56% and 1.77% than KNN classifier with actual attributes and PCA reduced attributes**.** From Fig. 2, it is observed that the FPR of Naive Bayes classifier improved by 12.2% and 7.14% than KNN classifier with actual attributes and PCA reduced attributes**.** From these two figures, it is also observed PCA selected attributes, the average Detection rate and average FPR are increased by 0.9% and 4.11% respectively when comparing to usage of all the attributes in the classifiers.

## V.Conclusion

In this paper, a novel method is proposed to classify the traffic flow into denial of service attacks and legitimate access by creating the access matrix from the HTTP traces. As the access matrix is multi-dimensional, Principle component analysis is used to reduce the attributes before classification. Naive Bayes and K-Nearest neighborhood classifiers are used to classify the traffic. The performance of the classification is compared by the detection rate and False Positive Rate (FPR). From the experiments and results obtained it is proved that with the PCA selected attributes, the average Detection rate and average FPR are increased by 0.9% and 4.11% respectively when comparing to usage of all the attributes in the proposed classifiers.

## REFERENCES

[1] Mirkovic, Jelena, and Peter Reiher. "A taxonomy of DDoS attack and DDoS defense mechanisms." ACM SIGCOMM Computer Communication Review 34.2 2004, 39-53.

[2] Dietrich, Sven, Neil Long, and David Dittrich. "Analyzing Distributed Denial of Service Tools: The Shaft Case." LISA. 2000, pp. 329-339.

[3] Arbor Networks, "Worldwide ISP Security Report", Sept. 2005, pp. 1-23.

[4] Lee, Wenke, and Salvatore J. Stolfo. "Data mining approaches for intrusion detection." Usenix Security. 1998, pp. 1-10.

[5] Gu, Qijun, Peng Liu, and Chao-Hsien Chu. "Analysis of area-congestion-based DDoS attacks in ad hoc networks." Ad Hoc Networks 5.5, 2007, 613-625.

[6] Li, Chao, Wei Jiang, and Xin Zou. "Botnet: Survey and case study." Innovative Computing, Information and Control (ICICIC), 2009 Fourth International Conference on. IEEE, 2009, pp. 1-20.

[7] McLaughlin, Laurianne. "Bot software spreads, causes new worries." Distributed Systems Online, IEEE 5.6 (2004): pp. 1-5.

[8] Thing, Vrizlynn L., Morris Sloman, and Naranker Dulay. "A survey of bots used for distributed denial of service attacks." New Approaches for Security, Privacy and Trust in Complex Environments. Springer US, 2007, pp. 229-240.

[9] Nazario, Jose. "Politically motivated denial of service attacks." The Virtual Battlefield: Perspectives on Cyber Warfare (2009): pp. 163-181.

[10] Alomari, Esraa, et al. "Botnet-based distributed denial of service (DDoS) attacks on web servers: classification and art." arXiv preprint arXiv:1208.0403 2012, pp. 24-32 .

[11] Kumarasamy, S., & Asokan, R. (2012). Distributed Denial of Service (DDoS) Attacks Detection Mechanism. arXiv preprint arXiv:1201.2007, pp. 41-49.

[12] Bhuyan, Monowar H., et al. "Detecting Distributed Denial of Service Attacks: Methods, Tools and Future Directions." The Computer Journal 2013, pp. 1-20.

[13] Gu, Q., & Liu, P. Denial of service attacks. Handbook of Computer Networks: Distributed Networks, Network Planning, Control, Management, and New Trends and Applications, Volume 3, 2007, pp. 454-468.

[14] Fu, Z., Papatriantafilou, M., & Tsigas, P. (2008, October). Mitigating distributed denial of service attacks in multiparty applications in the presence of clock drifts. In Reliable Distributed Systems, 2008. SRDS'08 pp. 63-72.

[15] Zargar, Saman Taghavi, James Joshi, and David Tipper. "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks." Communications Surveys & Tutorials, IEEE 15.4 2013, pp. 2046-2069.

[16] Yau, David KY, et al. "Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles." IEEE/ACM Transactions on Networking (TON) 13.1 2005, pp. 29-42.

[17] Chiueh, Shibiao Lin Tzi-cker. "A Survey on Solutions to Distributed Denial of Service Attacks." Department of Computer Science Stony Brook University 2006, pp. 1-38.

[18] Mirkovic, Jelena, et al. "Distributed defense against DDOS attacks." University of Delaware CIS Department Technical Report CIS-TR-2005-02, 2005, pp. 1-12.

[19] Moore, David, et al. "Inferring internet denial-of-service activity." ACM Transactions on Computer Systems (TOCS) 24.2, 2006, pp. 115-139.

[20] Weiler, Nathalie. "Honeypots for distributed denial-of-service attacks." Enabling Technologies: Infrastructure for Collaborative Enterprises, 2002. WET ICE 2002. Proceedings. Eleventh IEEE International Workshops on. IEEE, 2002, pp. 109-114.

[21] (Online). Available: http://ita.ee.lbl.gov/html/traces.html.

[22] Xie, Yi, and Shun-Zheng Yu. "Monitoring the application-layer DDoS attacks for popular websites." Networking, IEEE/AcM Transactions on 17.1, 2009, pp. 15-25.

[23] L. I. Smith, A Tutorial On Principal Components Analysis (EB/OL), 2003 (Online). Available: http://www.snl.salk.edu/~shlens/pub/ notes/ pca.pdf.

[24] Jiawei Han and MichelineKamber, “Data Mining Concepts and Techniques”, Second Edition, Elsevier, 2006, pp 512-513.

[25] Zhu, Xiaojin, and Andrew B. Goldberg. "Introduction to semi-supervised learning." Synthesis lectures on artificial intelligence and machine learning 3.1, 2009, pp. 1-130.