American International Journal of Computer Science and Information Technology
Volume.8, Number 1;January-March, 2023;
ISSN: 2837-388X| Impact Factor:
https://zapjournals.com/Journals/index.php/aijcsit
Published By: Zendo Academic Publishing

DATA-DRIVEN DEFENSE: ANALYZING DDOS ATTACKS WITH MULTIPLE DISCRIMINANT DATA ANALYSIS

Mehmet Ali Yılmaz[1]

Article Info

Keywords: Security, Distributed Denial of Service (DDoS), Confidentiality, Integrity, Availability

Abstract

Ensuring security remains a paramount concern in both business and public spheres. Among the myriad threats faced, hacking attacks, particularly Distributed Denial of Service (DDoS) attacks at the application and network layers, loom large. Identified vulnerabilities often grant attackers unauthorized access, allowing them to compromise web services and impede network functionality.

The foundation of data security rests on three fundamental tenets: confidentiality, integrity, and availability. Confidentiality entails safeguarding data from illicit use, involving scrutiny, restricted sharing, and controlled dissemination. Information categorized as highly sensitive is deemed secret and necessitates stringent, exclusive protection. Integrity guarantees the unaltered veracity of data, affirming its origin and authenticity. Availability ensures data accessibility to authorized users, underpinning its utility and relevance.

Introduction

Security has always been one of the major concerns in business and public affairs. Hacking attacks are big threads for firms and attackers develop different kinds of Distributed Denial of Service (DDoS) that relies on the application and network layer. Most vulnerabilities which indicated here let attackers prevent authorize and authenticate to web services and decelerate network operations.

Security in data should be considered in three different subjects: confidentiality, integrity, and availability. Confidentiality is defending data used by illegal affairs. Defending includes inspecting, sharing, and the appliance of information from the public. Highly classified information is expressed as secret and ought to be saved exclusively. Integrity is assuring the genuineness of information in such that information is not modified and the source is real and availability means that data is attainable by legitimate users.

[1] Computer Engineering, Bolu Abant Izzet Baysal University, Turkey

Background

Transmission Control Protocol/Internet Protocol (TCP/IP) is a protocol established by the United States Department of Defense (DoD) inthe 1970s. To connect computers with other same kind systems (routing), this protocol was intended and called ARPANET (Advanced Research Project Agency Internetwork (Forouzan&Fegan, 2006). It comes before the ancestor of the Internet that we use today. TCP/IP is consists of different other protocols thatare designed to carry information through interconnections.

Internet Protocol (IP) especially deals with the destination of data. For such purpose, each packet contains the source and destination data. Many attacks at the internet protocol layer maneuver this packet pattern. Transmission Control Protocol (TCP) ensures guarded distribution of information to the destination defined in the internet protocol. Most attacks occur on deficiency in TCP finite state machines. User Datagram Protocol (UDP) can be used as a substitute for TCP. It is connectionless and not assures that packet arrives at the destination. Also, it does not have a loss recovery system. Moreover, it is faster. UDP packets are more often used in performing flood attacks. Internet Control Message Protocol (ICMP) delivers check and error messages related to network conditions between entities. ICMP ECHO_REQUEST and ICMP ECHO_RESPONSE are the two types of ICMP (Jinhua, et. al, 2013). These two datagrams can be used to understand if a distant system is available on the network. This is mostly done by the "ping" command. In some circumstances, overfilling ICMP packets should disallowance services.

Transmission on a channel using TCP/IP or UDP/IP will mostly by definite packages. Every package has a sender and receiver address, data, and extra control information. Additionally, TCP and UDP use a defined port number for the connection. These port numbers specify the type of service. On the other hand, ICMP does not include TCP block. All compulsory information is stored in itself.

Distributed Denial of Service Attacks

The main objective of attacks is to discompose the victim's service. Distributed denial of service (DDoS) attacks are mostly decomposition of internet services by using deficiencies in IP rather than defacing of service. There are a lot of different types of DDoS attacks depends on their parameters. Normal attacks are proceeded by an individual host (or a few amount of hosts located at the same place). The ordinary method for an attack is accomplishing software and architecture pitfall. Such a bug can be the wrong implementation of IP stack which can blast the host while acquiring a nonstandard IP packet (such as ping-of-death). That kind of attack would mostly have diminished the size of data. Unless some unfixed exploits occur on victim hosts, most DDoS attacks do not impend a certain threat to quality services at present internet technology(Mirkovic, et. al, 2004). DDoS attacks are mostly accomplished by a huge amount of hosts. These servers might be amplifiers, reflectors, or zombies who were placed on remote hosts and have been looking for attack command. Most of the attacks done by hundreds of servers, producing hundreds of megabits per second floods. Bulk flooding is essential for the appliance used in DDoS attacks where assailants flood the subject with the maximum available number of packets to defeat the victim. DDoS attacks performed by lots of hosts attacking simultaneously. This can be executed by affecting internet servers with a "zombie". In this way, an assailant can be anybody with assured information and entry granted to the main server. By entering a few commands, all zombie colonies would be become active and arise a substantial attack on the victim.

There are many ways to place zombie programs on the subject host like an email attachment, flash animation, patch to a game. Transmission between a zombie and its master can be buried even by a standard protocol. DDoS attacks are very endemic and mainly targets to public services. Since a spread attack is performed through the net protocol, it is difficult to stop hundreds of flood serving hosts. If the packets are authorized requests, it will be very sorrowful. Because they cannot be correlated with a DDoS attack. On the other hand, DDoS attacks use

a huge number of assets from a framework like ISP's and interconnection devices. Such attacks should be more dangerous therefore a particular attack across a lightweight web server can destroy all ISP's infrastructure and many users can be influenced by out of services.

Most of the denial service attempts are distributed attacks and fundamentally depends on sending a mass amount of packets. Furthermore, reused packets should be transformed to increment the damage. To make an efficient DDoS attack, a few numbers of hosts connected via a T1 leased line would not be enough. Essentially for an extensive web server, a DS3 link might not be competent either. For generating a heavy load on the server, the attacker would use an enormous number of hosts with a fast connection. Collecting flood attacks from all hosts should perform a flood of large Mbps all conducted to a unique server. Such a huge amount of data can crush any organization within network infrastructures, circuits, and certainly servers although prevented by load balancers. These attacks affect not only the victim itself but others also nearby in-network range and defuse the routers serving to other clients. Most of these assaults require an "army" of zombies that are distributed around the internet. This army contains servers that have an exploit installed on them(Ioannidis, et. al, 2002). Posterior software installation, these hosts would communicate with the centric unit for receiving commands or patches. This communication canal enables the attacker to start DDoS floods. There are many ways to distribute zombie programs on the internet. The basic approach is to use security holes and installing by hand. Other common methods are putting Trojan inside a game installer, an MP3 song, or other media files. Also, some worms like Code Red worm, use vulnerabilities in Microsoft Internet Information Service (IIS) servers to spread out rapidly. The basic way for a Trojan to connect with a master is using TCP communication and get instructions from this channel. But this way of connection is traceable and so it makes Trojan weak. Actuating "netstat" command on the influenced host would show the association, existence of Trojan, and the IP of the attacker. For hiding, Trojan should use more safety protocols like HTTP or IRC. In an HTTP connection, Trojan can use a specific CGI page to communicate with the master. This contact would be cast away between further HTTP connections. Additionally, the connection can be comparatively stanchly and confirm that the Trojan is updated but not seen from the victim(Lee, Keunsoo, et al, 2008). One more usual way is using internet relay chat protocol. A trojan horse would connect to a specified IRC channel. Anyone who uses this station to text messages to Trojans and then it attacks back. Sub7 is the most known Trojan in this manner. As a protection method for master, a common way is to encrypt the full session. The class of attack can differ related to Trojan appliances. Most often, it facilitates minimal kinds of attacks.

The attacker determines the kind of attack, packet size, target IP, and more other features. In essence, any type of common attack can be done by those zombie Trojans. Defending from such attacks is notably cumbersome because the massof attack may be very colossal and block complete transmission capacity. The devices in the back of ISP cannot handle such network traffic. Other than zombie hosts, attackers use spoofed IP addresses or imitate the master IP address to hide. Also using reflectors is someway else. The attacker reflects the assault from disparate hosts and it causes the client to see them as attackers. Reflectors should be applied to many different attacks. Both a single host and many zombies can use reflectors to undercover the master source. Nearly any computer which servesthe internet can be the potential to be a reflector because it conforms IP standards. The main aim is to accomplish common protocols that have a request-response arrangement. An attacker sends the request with source IP arranged to the subject's address. It responds to the victim by effectually reflecting the attack.

HTTP Flooding Attack

HTTP flood uses HTTP GET or POST exploit requests to hit a system by using mostly botnet (zombie army) online computers. When an HTTP client establishes a connection with an application server, it sends either GET

for fetching mostly static content or a POST request for dynamically formed assets. Maximum damage is caused by a single request occupying the most resources. Therefore, the attacker constantly sends a large number of requests. Therefore, attackers mostly prefer POST requests as they activate server-side operations(Lu, Wei-Zhou, et. al, 2006). However, GET request is much simpler and adequate for botnet systems. These are hard to distinguish from the normal request as they use standard URLs. Conventional ret-based catching solutions are incapable because HTTP floods are mostly below the threshold evaluation limit. Mitigating HTTP flood attacks is complicated and versatile. One way is using a captcha like compulsive mechanism to identify whether it is a bot or not. Another solution is using a web application firewall and limiting the clients. Figure 1 shows a typical HTTP flood attack.

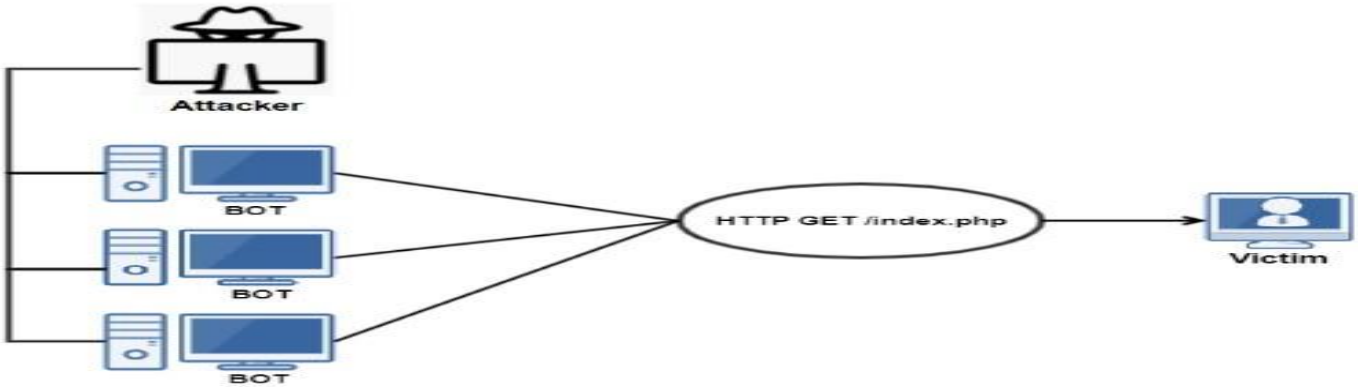

Figure 1. Http Flood Attack

UDP Flooding Attack

UDP flood is useda huge number of User Datagram Protocol (UDP) packages to server to crash. The firewall on a server should have damage after a UDP attack. When clients a UDP packet a server's specific port, server checks if any program runs using that port. If there are no applications, the server sends back an ICMP (ping) packet to tell that target is unachievable. Because the targeted server uses resources to control and then respond to each received UDP packet, its resources can quickly deplete when it receives too many UDP packet floods causing a denial of service(Carl, Glenn, et al, 2006). UDP attack scenario is shown in Fig. 2. To prevent UDP attacks, most servers restrict the number of ICMP responses.

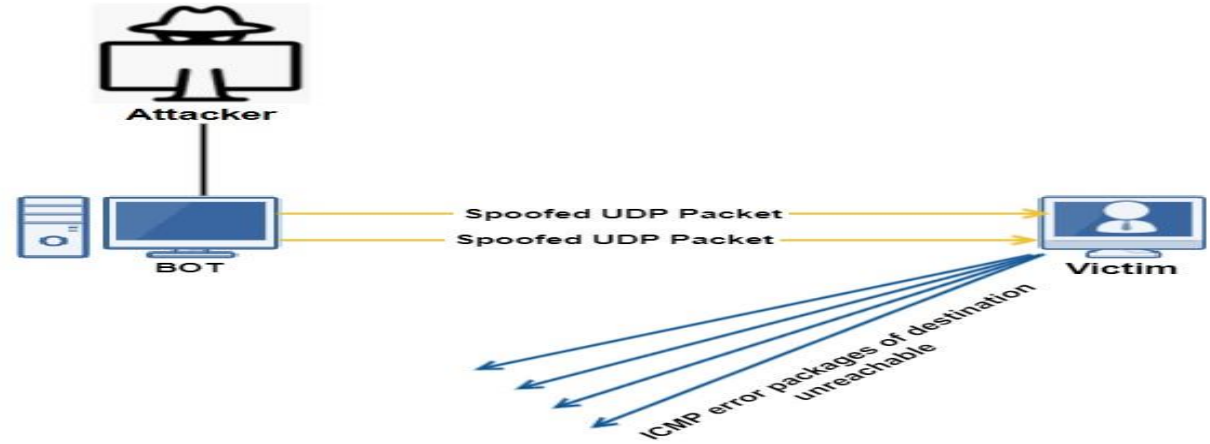

Figure 2. UDPFlood Attack

Smurf Attack

In this kind of attack, ICMP echo packets are transmitted to internet protocol broadcast addresses from distant places to perform DDOS attacks. They look the same as ping floods as they transmit a huge amount of ICMP

Echo request packages. Apart from the known ping flood, smurf is a boosting attack that expands loss by taking advantage of widespread networks. In the IP broadcast network, a ping is addressed to every client and waits for a response. In Smurf attacks, attackers exploit this property to boost assault traffic(Kumar, Sanjeev, 2007).
A typical Smurf attack starts with accomplishing a false request including a spoofed origin IP which is the target server address. The request is broadcast to the entire network. After receiving the request, each host sends an ICMP response to the actual source address. At soon, a server fails down because of overflowing its capacity. To prevent the system from attacks, IP oriented broadcasting should be disabled, and sending ICMP responses to IP broadcast networks should be disallowed (Gil, Thomer M, et al, 2001). The realization process of the smurf attack is shown in Figure 3. If a Smurf DDoS attack succeeds, it will paralyze the company's servers for hours or days, causing both losses of revenue and anger of customers. Also, such attacks may cover a worse activity, such as theft of files or other intellectual property theft. Smurf and similar DDoS attacks require a robust protection strategy that can monitor network traffic and detect oddities such as packet volume, behavior, and signature.

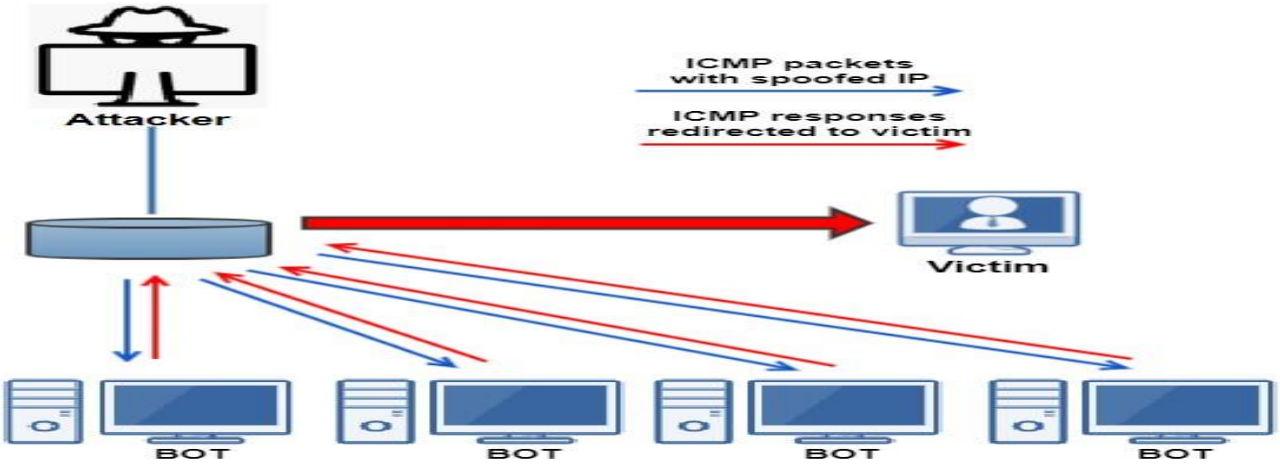

Figure 3. Smurf Attack

SQL injection (SiDDoS) Attack

SQL injection is a kind of attack that apply malevolent SQL code for accessing system database. These databases can contain highly important information about private customer data, private company data, and others. Frequently, the main target is the databases behind a website. In some cases, SQL commands can also include operating system calls. Thus, an effective SQL injection attack can cause very considerable outcomes. SQL Injection is one of the most serious vulnerabilities in web applications (Boyd, et al., 2004). Especially with the popularization of extra database layers such as frameworks and ORM (Object Relational Mapping), they are seen a little less nowadays than before but still show their effect. Web application developers make some fatal errors because they do not fully understand SQL Injection. SQL injection is used to insert malicious code into SQL commands through a web page. SQL injection is usually a SQL command that runs unannounced in the database when the user is asked for an entry, such as a user name.

Materials and Methods

Discriminant analysis is used to establish discriminant functions which are linearly or nonlinearly composite of independent variables that will separate the classes of the dependent variable. It is applicable when the dependent y variable is categorical and independent x variables are interval. It enables us to check if meaningful deviations occur in groups among predictors. It also criticizes the accuracy of a classification and it is briefed as to the

number of classes consumed by dependent variables. Discriminant Analysis is widely used to create perceptual mapping and frequently used with cluster analysis together.

Mixture Discriminant Analysis

In the mixture discriminant analysis, assume that there exists a training set n_j from group j for $j = 1 \ldots G$. Every type of j is sectioned into R_j abstract subclasses represented as c_j. In conformity with clustering manner, every subclass possesses a multivariate normal distribution $x_i \sim N(\mu_{jr}, \Sigma_{jr})$ where μ_{jr} is mean vector and Σ_{jr} is covariance matrix for the r^{th} subclass nj^{th} class(Bashir, ^{R} Shaheena, and E. M. Carter, 2005). The prior probability for class j is μj and μjr is the mixing probability for the r^{th}subclass in jth class, such that $\sum_{r=1}^{j} \mu_{jr} = 1$. Then mixture density for class j is

$$m_j = P(X = x|G = j) = |2\mu\Sigma_j| x^{-\frac{1}{2}} \sum_{r=1}^{R_j} \mu_{jr} \exp[-D(-\mu_{jr})/2] \quad (1)$$

Where Mahalanobis distance is computed for$D(-\mu_{jr})$. Soon, posterior probabilities are achieved stand on Bayes rule

$$P(X = x|G = j) \sim \pi_j Prob(x|j) \sim \pi_j \sum_{r=1}^{R_j} \mu_{jr} \exp[-D(-\mu_{jr})/2] \quad (2)$$

Whereπ_j denotes prior probability for class j. The estimation is arranged to j that owns the highest posterior probability. The differentiation criteria rely on anonymous features that should be predicted from training data.

Quadratic Discriminant Analysis

Quadratic discriminant analysis (QDA) is varying from the linear discriminant analysis where a characteristic covariance matrix is figured for each category of records. QDA is especially essential if there exists precedent information about distinct types that shows different covariance. A drawback here is that it is not possible to use dimensionality reduction (Srivastava,et al., 2007). The discriminant function of linear discriminant analysis is quadratic in x:

$$\delta_x = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x - \mu_k)^T \sum_k^{-1}(x - \mu_k) + log\pi_k \quad (3)$$

Considering that QDA evaluates the covariance matrix for every taxon, it has much more efficient arguments than linear discriminant analysis. As there exist K centroids, μk, with p inputs particular, hence $K_p$parameters will be associated with means. As$\sum_{i=1}^{K} \mu_k = 1$, no argument is necessary for one of the preceding. So, there will be $K-1$ unbound parameters for preceding. From the covariance matrix, Σ_k the diagonal and the upper right triangles are regarded. There are $p(p+1)/2$ elements in this area.

Because K matrices supposed to be predicted, there will $Kp(p+1)/2$ arguments about covariance matrices. QDA should be applied very carefully when there isa big number of attributes because of its quadratic number of parameters in p (Lachenbruch, Peter A., and M. Goldstein, 1979).

Regularized Discriminant Analysis

Regularized discriminant analysis (RDA) is an adjustment between linear discriminant analysis and quadratic discriminant analysis. It diminishes Σ_k to a cumulative variance Σ through

$$\widehat{\Sigma_k}(\alpha) = \alpha\widehat{\Sigma_k} + (1 - \alpha)\widehat{\Sigma} \quad (4)$$

And replacing $\widehat{\Sigma_k}$with $\widehat{\Sigma_k}(\alpha)$along with discriminant intentions. In this equation, $\alpha \in [0, 1]$ is a tuning specification designates if the covariance will be calculated unbound (α=1) or will be aggregated (α=0) (Friedman, Jerome H., 1989). Besides, $\hat{\Sigma}$should be diminished along with the scalar covariance beyond needing

$$\widehat{\Sigma}(\gamma) = \gamma\widehat{\Sigma} + (1 - \gamma)\hat{\sigma}^2 I \quad (5)$$

Where γ =1 guides to cumulative covariance and γ =0 guides to scalar covariance. Channing $\widehat{\Sigma_k}$ by $\widehat{\Sigma}(\alpha, \gamma)$ moves to a deeper generic concept of covariance. RDA is especially functional when a lot of attributes are correlated as far as possible (Guo, et al., 2006).

Experimental Classification Results and Analysis

The network analysis data is collected from previous research on network analysis. The dataset contains four types of DDoS attack as follows: (HTTP Flood, SIDDOS, UDP Flood, and Smurf) without redundant and duplicate records[16].

Table 1. Number of observations in each class

Attack Name	Number of Records
Smurf	12590
UDP Flood	201344
SIDDOS	6665
HTTP Flood	4110
TOTAL	224709

R programming language is used for data analysis. R Project is an open-source programming language specialized in statistical calculation and graphs. It is a programming language developed by Ross Ihaka and Robert Gentleman in 1993 and has a comprehensive catalog of statistical and graphical methods. Machine learning, linear regression, statistical implications for time series. Most R libraries are written in the R language, but C, C ++, and FORTRAN codes are preferred for heavy computational work. R is used not only by academic research organizations but also by many large companies. R, supported by the R Foundation and part of the GNU, can be considered an adaptation of the S language. Despite some important differences, the codes written for S can also work in R. The S language exists today as R (GNU Free Software) and S + (S-PLUS, Commercial Product). It is one of the most commonly used languages in the field of data analysis (linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering, etc.), has a wide range of documents produced in many languages. In addition to the documentation, the fact that it has an application package (or easily developed by a user to suit the needs) in almost every subject is one of the issues that highlight the R programming language (Team, R. Core, 2013).

Table 2. Feature Importance Values

	HTTP Flood	SIDDOS	Smurf	UDP Flood
PKT_RATE	0.9745	0.9998	0.9745	0.9998
NUMBER_OF_PKT	0.9739	0.9987	0.9739	0.9987
PKT_TYPE	0.8514	0.9801	0.8329	0.9801
PKT_SIZE	0.9720	0.9720	0.9720	0.9376
PKT_AVG_SIZE	0.9720	0.9720	0.9720	0.9376
BYTE_RATE	0.9636	0.9636	0.9636	0.8353
PKT_DELAY	0.8139	0.9547	0.7303	0.9547
NUMBER_OF_BYTE	0.9251	0.9251	0.9495	0.8637
PKT_R	0.8738	0.8738	0.9042	0.5442
PKT_IN	0.8738	0.8738	0.9041	0.5440
PKT_OUT	0.8737	0.8737	0.9041	0.5442

Feature selection reduces the size of the feature set and increases algorithm speed, eliminates unrelated and noisy data, improves data quality, makes the data set more easily identifiable, visualized, and understandable. Also, it saves resources for data collection required to create the data set, reduces the amount of memory required to store data, and increases the success of the model obtained. In this study, Learning Vector Quantization (LVQ) is used for training and feature selection. It is an overseen kind of vector quantization that used when data is tagged (Sato, Atsushi, and Keiji Yamada, 1996). This training method uses taxon data to relocate Voronoi vectors gently to enhance the classifier identification boundary. R programming language has a generic method named "*varImp*" in the caret library that calculates variable importance for objects produced by train and method-specific methods(Kuhn, Max, 2008).

When a model is created for a classification problem, or when existing models are used, the success of that model is considered as the number of correct estimates from all predictions made. However, this information only gives the accuracy of the classification. Classification accuracy alone is often not enough information to decide whether a model is good enough or not. In this study, the complexity matrix was used for evaluation criteria. One clear way to present the estimation results of a classifier is to use a confusion matrix. A confusion matrix is a frequently used table to describe the performance of the classification model with a set of test datawhose actual values are known (Townsend, James, 1971).

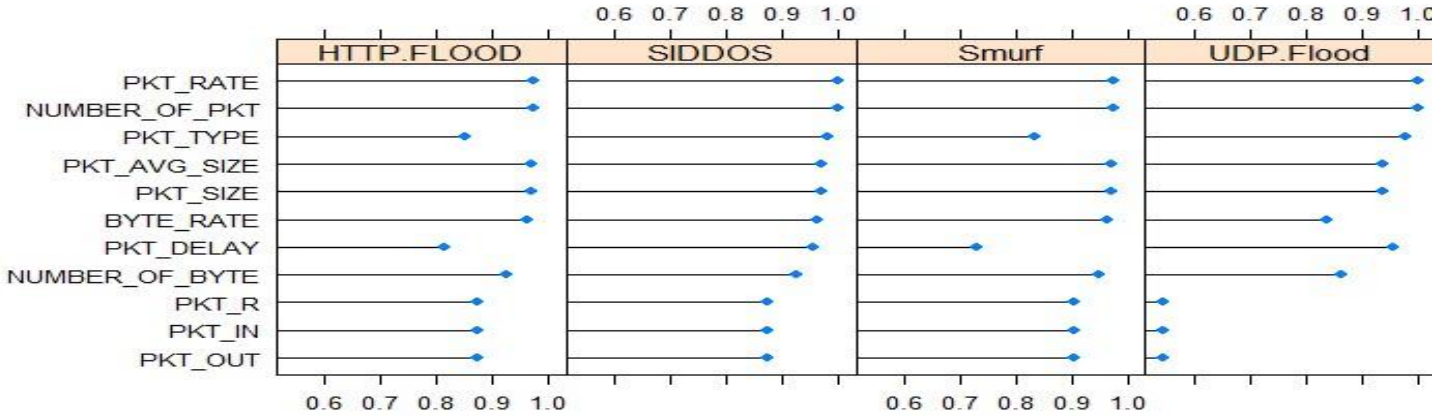

Figure 4. Feature importancer anking

Table 3. Mixture Discriminant Analysis Evaluation Values

	HTTP Flood	SIDDOS	Smurf	UDP Flood
Sensitivity	0.94647	0.97044	0.70008	0.9007
Specificity	0.99846	0.94626	0.94279	1.0000
Pos. Pred. Value	0.91962	0.35568	0.42072	1.0000
Neg. Pred. Value	0.99900	0.99905	0.98147	0.5388
Prevalence	0.01829	0.02966	0.05603	0.8960
Detection Rate	0.01731	0.02878	0.03922	0.8070
Detection Prevalence	0.01882	0.08093	0.09323	0.8070
Balanced Accuracy	0.97247	0.95835	0.82143	0.9503
Accuracy			0.8923	
Kappa			0.5997	

Accuracy is the difference between the actual value and the value displayed by the device when measuring a physical property. Accuracy is the most intuitive measure of performance and the ratio of accurately predicted

observation to total observations. It can be considered that the model is the best if the model used has high accuracy. However, in cases where the number of false-positive and false-negative values is quite different and very different from each other, other parameters should be looked at to evaluate the performance of the model.

Table 4.Quadratic Discriminant Analysis evaluation values

	HTTP Flood	SIDDOS	Smurf	UDP Flood
Sensitivity	0.94063	0.93878	0.142812	0.9791
Specificity	0.98126	0.99710	0.979785	0.7179
Pos. Pred. Value	0.48325	0.90826	0.295432	0.9676
Neg.Pred. Value	0.99887	0.99813	0.950636	0.7997
Prevalence	0.01829	0.02966	0.056028	0.8960
Detection Rate	0.01720	0.02784	0.008001	0.8773
Detection Prevalence	0.03560	0.03066	0.027084	0.9067
Balanced Accuracy	0.96095	0.96794	0.561298	0.8485
Accuracy			0.9304	
Kappa			0.6227	

Sensitivity is the ratio of accurately predicted positive observations to estimated total positive observations. This is also called Positive Predictive Value. Sensitivity can be considered as a measure of the accuracy of classifiers. Low precision may also indicate a large number of false positives. Recall is the ratio of the correct predicted results to the total number of positives. The proportion of positive observations accurately predicted for all observations in the classification. Sensitivity can be considered as a measure of the integrity of classifiers. Low sensitivity indicates a lot of false negatives.

Table 5.Regularized Discriminant Analysis evaluation values

	HTTP Flood	SIDDOS	Smurf	UDP Flood
Sensitivity	0.00000	0.97629	0.00000	0.9301
Specificity	1.00000	0.90476	1.00000	0.5649
PosPred Value	NaN	0.23858	NaN	0.9485
NegPred Value	0.98171	0.99920	0.94397	0.4839
Prevalence	0.01829	0.02966	0.05603	0.8960
Detection Rate	0.00000	0.02896	0.00000	0.8334
Detection Prevalence	0.00000	0.12137	0.00000	0.8786
Balanced Accuracy	0.50000	0.94053	0.50000	0.7475
Accuracy			0.8623	
Kappa			0.3417	

Cohen's kappa coefficient is a statistical method that measures the reliability of the comparative agreement between two evaluators (Cohen, Jacob, 1960). It measures the agreement between two evaluators, each of which separates N substances into C mutually exclusive categories. The resulting categorical variable is a non-parametric type of statistics. Since the Kappa measure also considers this agreement to be a chance, it is considered to give a stronger result than the agreement found as a simple percentage ratio.

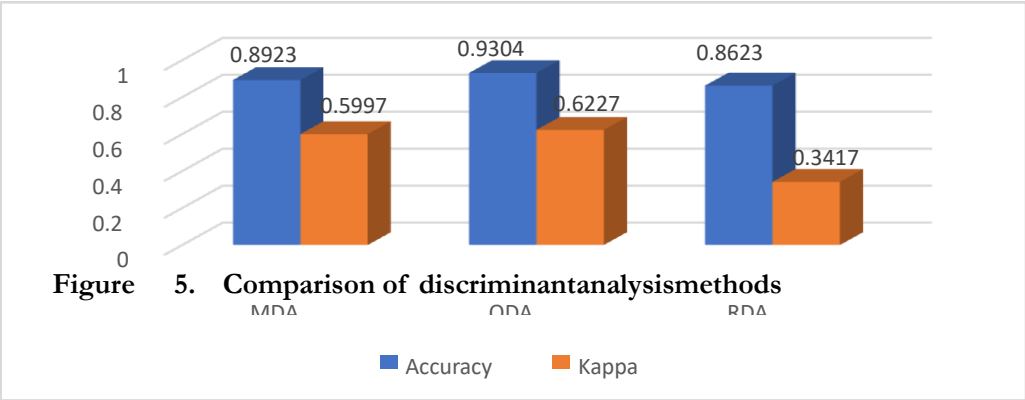

Figure 5. Comparison of discriminantanalysismethods

Conclusion

Denial of Service attacks are performed by restraints of transmission protocols and using a lack of security in applications. As these attacks are consistently expanding, they are bringing new difficulties on how to struggle with their influences. To preserve the system from these attacks, some basic protection procedures should be applied. Setting up a firewall with a router filter application and watching a network permanently for unusual packet transmission are the preliminary rules. Also, network administrators should keep themselves up to date for continuously renewed attack techniques. DDoS attacks are planned to demolish affairs by flooding using fake communication and transactions to finish successfully.

In this study, it is seen that PACKET_RATE is the main factor through all analysis as it has the highest importance value. Also, NUMBER_OF_PACKETS and PACKET_SIZE have a significant effect whereas PACKET _TYPE has lagged. Quadratic Discriminant Analysis has the maximum accuracy and kappa value over Mixture Discriminant Analysis and Regularized Discriminant Analysis. To protect from attacks, internet firms need a nextgeneration architecture, state of art techniques to catch and vanquish these assaults by considering these discriminant analysis related to attack types.

References

Alkasassbeh, Mohammed, et al. "Detecting distributed denial of service attacks using data mining techniques." *International Journal of Advanced Computer Science and Applications* 7.1 (2016): 436-445.

Bashir, Shaheena, and E. M. Carter. "Robust reduced rank mixture discriminant analysis." *Communications in StatisticsTheory and Methods* 34.1 (2005): 135-145.

Boyd, Stephen W., and Angelos D. Keromytis. "SQLrand: Preventing SQL injection attacks." *International Conference on Applied Cryptography and Network Security. Springer*, Berlin, Heidelberg, 2004.

Carl, Glenn, et al. "Denial-of-service attack-detection techniques." *IEEE Internet Computing* 10.1 (2006): 82-89.

Cohen, Jacob. "A coefficient of agreement for nominal scales." *Educational and psychological measurement* 20.1 (1960): 3746.

Forouzan, Behrouz A., and Sophia Chung Fegan. *TCP/IP protocol suite*. Vol. 2. McGraw-Hill, 2006.

Friedman, Jerome H. "Regularized discriminant analysis." *Journal of the American statistical association* 84.405 (1989): 165175.

Gil, Thomer M., and Massimiliano Poletto. "MULTOPS: A Data-Structure for Bandwidth Attack Detection." *USENIX Security Symposium*. 2001.

Guo, Yaqian, Trevor Hastie, and Robert Tibshirani. "Regularized linear discriminant analysis and its application in microarrays." *Biostatistics* 8.1 (2006): 86-100.

Jinhua, Gao, and Xia Kejian. "ARP spoofing detection algorithm using ICMP protocol." 2013 *International*

Ioannidis, John, and Steven Michael Bellovin. "*Implementing pushback: Router-based defense against DDoS attacks*." (2002).

Kuhn, Max. "Building predictive models in R using the caret package." *Journal of statistical software* 28.5 (2008): 1-26

Kumar, Sanjeev. "Smurf-based distributed denial of service (DDOS) attack amplification in internet." *Second International Conference on Internet Monitoring and Protection (ICIMP 20070)IEEE*, 2007.

Lachenbruch, Peter A., and M. Goldstein. "Discriminant analysis." Biometrics (1979): 69-85.

Lee, Keunsoo, et al. "DDoS attack detection method using cluster analysis." *Expert systems with applications* 34.3 (2008): 1659-1665.

Lu, Wei-Zhou, and Shun-Zheng Yu. "An HTTP flooding detection method based on browser behavior." *2006 international conference on computational intelligence and security*. Vol. 2. IEEE, 2006.

Mirkovic, Jelena, and Peter Reiher. "A taxonomy of DDoS attack and DDoS defense mechanisms." *ACM SIGCOMM Computer Communication Review* 34.2 (2004): 39-53.

Sato, Atsushi, and Keiji Yamada. "Generalized learning vector quantization." *Advances in neural information processing systems*. 1996.

Srivastava, Santosh, Maya R. Gupta, and Béla A. Frigyik. "Bayesian quadratic discriminant analysis." *Journal of Machine Learning Research* 8.Jun (2007): 1277-1305.

Team, R. Core. "R: A language and environment for statistical computing." (2013): 201. Townsend, James T. "Theoretical analysis of an alphabetic confusion matrix." *Perception & Psychophysics* 9.1 (1971): 4050.