A Proposed Technique for Simultaneously Detecting DDoS and SQL Injection Attacks

Istiaque Hashem
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka-1208, Bangladesh

Minhajul Islam
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka-1208, Bangladesh

Shazid Morshedul Haque
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka-1208, Bangladesh

Zobaidul Islam Jabed
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka-1208, Bangladesh

Nazmus Sakib
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka-1208, Bangladesh

ABSTRACT

As people's reliance on the internet grows, they reveal their data without realizing the implications of a cyberattack. A cyberattack is any form of attack against one or more computers or networks to cause damage. These attacks have the potential to compromise network access, confidentiality, and data integrity. The most popular and powerful attacks to destroy an enterprise, server, or host are distributed denial-of-service (DDoS) and Structured Query Language injection (SQLi). A distributed denial-of-service (DDoS) attack can freeze an entire website with an intention to ransomware or push viruses. On the other hand, with a successful Structured Query Language injection (SQLi) attack, hackers can access the secret information of a legit user. To deal with DDoS and SQL injection attacks, a variety of techniques have been developed. However, hackers use different techniques to breach security mechanisms, many of which are undetectable by most intrusion detection systems because of their unpredictability. In this paper proposal of a system is given that can detect DDoS and SQL injection attacks simultaneously. Right now, there is no such system that can detect both attacks at the same time. A secure way of browsing the internet and sharing information can be ensured with this system. Webservers will be more secured.

Keywords

DDoS; SQL Injection; Machine learning; Knn; Random Forest and Decision Tree; NSL-KDD Dataset; Weka tool

1. INTRODUCTION

As the importance of protecting a network, device and the privacy of personal and confidential data is a national concern, cyberattacks are a significant issue to address. A cyberattack is an attempt to gain unauthorized access to a device, operating system, or computer network to cause damage. Cyberattacks are designed to disable, interrupt, kill, or take control of computer systems and modify, block, erase, exploit, or steal the data contained within them. Cyberattacks can be broken down into two broad types: attacks where the goal is to disable the target computer, network or knock it offline, or attacks where the goal is to access the target computer's data and perhaps gain admin privileges [1][2]. To obtain these goals, different methods are being used by cybercriminals. Distributed Denial-Of-Service (DDoS) and Structured Query Language injection (SQLi) attacks are two of the most common and vicious cyberattacks. A distributed denial-of-service (DDoS) attack is known as a malicious attempt to disrupt the regular traffic of a targeted server, service or network by overwhelming the target or its surrounding infrastructure with a flood of internet traffic [3]. Structured Query Language injection (SQLi) is one of the most prevalent threats to a database system. An attacker injects a Structured Query Language (SQL) statement into an input box to access resources or modify data stored in the database[4] [5]. DDoS and Structured Query Language injection (SQLi) attacks are the most dangerous threats to a web-based application. According to NetScout threat intelligence, in 2019, 8.4 million DDoS attacks were reported. This means 670,000 attacks per month, 23,000 attacks per day and 16 attacks every minute[6].Recently, the United States web-based hosting service GitHub was hit by the largest distributed denial of service (DDoS) attack in history. It was subjected to a colossal 1.35Tbps flood of traffic: unknown hackers attempted to take the platform offline, resulting in major websites across large portions of the United States being out of action for several hours[7].On the other hand, exercise shows that Structured Query Language injection (SQLi) is now represented nearly two-thirds (65.1%) of all web application attacks. That's up sharply from the 44% web application layer attacks that Structured Query Language injection (SQLi) represented just two years ago [8]. Both of these attacks can cause unrecoverable damage to a web server or web-based organization. A successful Distributed Denial of Service (DDoS) attack can halt an entire website or web-based application where a successful Structured Query Language injection (SQLi) attack can steal or manipulate a legit user's personal and confidential data. It is crucial to protect a network, server and database from these vicious attacks. However, existing security mechanisms do not provide an effective defense system against these attacks or the defense capability of some mechanisms is only limited to specific Distributed Denial of Service (DDoS) or Structured Query Language injection (SQLi) attacks[9].

In this paper, a new defense system is proposed to detect

Distributed Denial of Service (DDoS) and Structured Query Language injection (SQLi) attacks simultaneously. A system with two layers, one for detecting Distributed Denial of Service (DDoS) attacks and another for detecting Structured Query Language injection (SQLi)attacks, can be a solution to the majority of cyberattacks directed at a website. In consequence, networks, websites, web-based organizations, database systems can be safer.

The rest of the paper follows in section 2 about the literature review, then our proposal is in section 3. In section 4, we describe our experiment and study of the system's DDoS part and so on.

2. LITERATURE REVIEW

2.1 Distributed Denial of Service (DDoS)

Distributed Denial of Service attack is one of the most frightening attacks. It mainly targets a web server, network or website. A Distributed Denial of Service (DDoS) attack can easily overwhelm an entire site by sending much more requests than the server can handle. Distributed Denial of Service (DDoS) attacks can be categorized into three groups[10].

i)Volume-based attacks: In volume-based Distributed Denial of Service (DDoS) attacks, the abnormality is generated by absorbing the target's usable bandwidth between the target and the internet. It is mainly accomplished by sending a very large amount of traffic to a target. Examples include UPD Floods, ICMP Floods, and Spoofed-Packet Floods[11]

ii)Protocol-based attacks: Protocol-based attacks may bring a service to a halt by using all the capacity of the server. Additionally, it is referred to as a state-exhaustion attack. SYN Flood, Ping-Of-Death, Smurf DDoS are some of the examples [11]

iii)Application layer attacks: In this case, the attacker establishes a link with the target and then monopolizes processes on the server, exhausting its resources. Slow Loris, Apache are examples of this attack [11].

Every Distributed Denial of Service (DDoS) attack has an attacker, several handlers, a large number of zombies and a victim. They are named zombies because they don't have control over themselves. Attackers take complete control of the zombies. A group of zombies is also known as a botnet. These zombies are computers, laptops or any other device that can access the internet. The zombies can be manipulated by the handlers. A special program is run by the zombies that can generate a flood of packets. First, the attacker gains full authorization of the zombies and plants the code that causes them to create powerful attack streams. The attackers also try to hide the code to avoid detection. The attackers contact the handlers to command the zombies. The communication is done via TCP, UDP, ICMP protocols. The attacker commands the zombies to attack the target at the scheduled time set by him. The duration, attack type, protocol number and other features can be adjusted by the attacker to make the requests legit. As zombies are legit internet devices, it is very difficult to detect Distributed Denial of Service (DDoS) attacks. Several tools help to generate Distributed Denial of Service (DDoS) attacks [12]. Tools like Ramen worm[13] and Code Red[14] helps the zombies to hide the codes that can create the attacks. There are also tools like Trinoo[15] Tribe Flood Network (TFN)[16] to help the attackers to generate Distributed Denial of Service (DDoS) attacks.

2.2 DDoS Defense Mechanism

Distributed denial of service (DDoS) attacks poses a significant threat and are extremely difficult to protect against. Distributed denial of service (DDoS) defense mechanism can be divided based on activity deployed or location deployment [17]. Based on activity, Distributed denial of service (DDoS) defense system can be further categorized.

Intrusion prevention: The best way to defend against an attack is to prevent it from happening. This technique can be implemented in many ways. One way is changing the IP address of the victim. Another way is disabling IP broadcasts. As a result, attacks like ICMP flood and Smurf can be prevented [17]

Intrusion detection: The intrusion detection systems detect Distributed denial of service (DDoS) and inform the admins to take necessary measures. These systems are developed by training and testing existing datasets of distributed denial of service (DDoS) attacks. There are a few existing datasets like KddCUP'99, NSL-KDD, UCLA, CARDA. These datasets contain the features of various Distributed denial of service (DDoS) attacks. Classification algorithms are used to train and test intrusion detection systems to classify whether a request is legit or an anomaly. Random Forest, KNN, J48 decision tree, Support Vector Machine are some of the classification algorithms that are frequently used [18] [19] [20] [21] [22]

Intrusion Response: The intrusion response is referred to as the traceback to the source of the Distributed denial of service (DDoS) attack, which is then blocked. There are many methods to traceback the source. IP traceback, ICMP traceback are some methods to respond against Distributed denial of service (DDoS) [17].

In this study, classification algorithms are used to detect Distributed denial of service (DDoS) attacks for the proposed system.

2.3 SQL Injection Attack

Structured Query Language injection (SQLi) is a form of web hacking technique in which an attacker inserts code to manipulate a SQL query to obtain unauthorized access to a database. As exchanging information over the Internet through various channels and web applications became a fairly widespread phenomenon, these applications and their related databases are vulnerable to all sorts of threatsto information security since they can be accessed through the Internet. To better understand SQL injection attacks, a website that allows users to log in by entering their username and password can be used as an example. Following is the query constructed in case of an authorized login attempt where the username is 'user' and password '123'.

SELECT * FROM users WHERE name = 'user' and password = '123'

However, it is also possible to type the following input into the website's username with 'user' and password field with" or '1'='1' by a user with maliciousintent [23] *SELECT * FROM users WHERE name = 'user' and password = "or '1'='1'.*

Here, this user will always be logged into the website because 1=1 will always be valid. Hence, unauthorized access to the account details of authentic users is gained by the attacker and ownership of this data may have significant implications for the individual who is the authentic account owner. This is known as fraud and data privacy infringement. This is a straightforward example of a SQL injection attack just for understanding [23].But there are various and more complex forms of SQL injections. Even though many methods for dealing with these attacks have been developed, hackers continue to succeed in getting through the numerous security mechanisms to deal with SQL injection attacks. The main cause of this issue is the unpredictability of multiple attacks since it is impossible to foresee the exact form of attack that would be used. One of the earliest and most widely used methods for detecting SQL Injection attacks is pattern matching. AMNESIA is the most known tool which uses a pattern-matching mechanism. To classify malicious queries, the user input is matched with the registered patterns [24]. Dynamic tainting is also a similar method-based pattern matching mechanism. All user inputs are treated as tainted data in dynamic tainting. It is compared to previously registered trends by dynamic tainting [25].SQL Check is a well-known parsing technique in which each user input is treated as an argument query and the syntax of SQL queries is analyzed by the SQL parser. There is also a hybrid that combines pattern matching and a role-based access control mechanism. Machine learning algorithms are also used to detect SQL injection attacks. Where tokenization process is used One benefit of using machine learning algorithms is that it allows a wider variety of SQL queries to be leveraged. In addition, identification accuracy is improved, and the rate of false positives is decreased. Tokenization: Tokenization is a very common method used while working with Natural Language Processing (NLP). This method is also very useful for detecting SQL injection attacks.

2.4 Tokenization

Tokenization is essentially splitting a phrase, sentence, paragraph, or entire text document into smaller units, such as individual words or terms. Each of these smaller units is called tokens [26].In the query parser method, the tokenization technique is also implemented. In this method, the original query and query with injections are considered differently [27]. Combining the tokenization method with machine learning algorithms can also show promising results. As the pattern of these attacks is not easy to predict so, machine learning methods can work with tokens as features.

Figure 1 shows a simple example of tokenization, where thesentence is separated into multiple tokens. Each of these tokens can work like features. These features are very useful while working with different machine learning techniques. As the pattern of these kinds of attacks is not easy to predict, tokenization is a very useful method to split the query into multiple numbers of tokens to use these as features for classification. The model can be trained with a training dataset consisting of both malicious and non-malicious queries after tokenizing each of these queries. While external queries are pushed into the server, these queries also go through tokenization method while both the queries are compared with each other using machine learning algorithms.

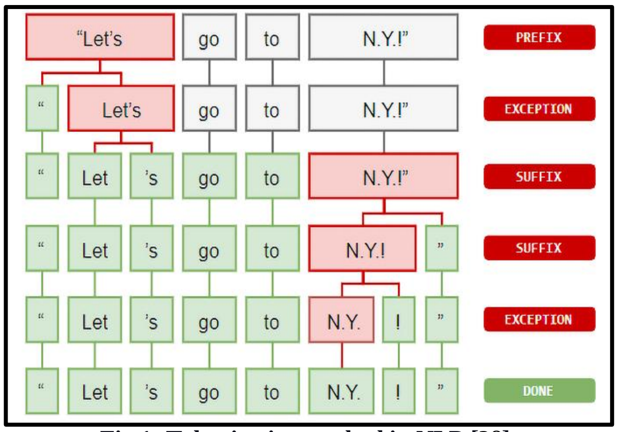

Fig 1: Tokenization method in NLP [28]

3. PROPOSED IDEA

In the study, a new mechanism is proposed which capable of detecting DDoS attacks and SQL injection simultaneously. It is a two-part system that's the first part is competent to detect DDoS attack consisted with network layer, transport layer and application layer of the OSI model. Its second part is competent to detect SQL injection which is consisted with application-layer functionality.Though the purposes of those two tiers are same which is to prevent suspicious activity on the server-side their main functionalities of detecting attacks are not equivalent.

This proposed system should be present on a network server where those attacks have occurred. Whenever the system is deployed on the server, it's start monitoring. So, in the first part to detect a DDoS attack a machine learning-based solution is used to find network traffic in normal and attack state, TCP-SYN or ICMP flood attack, how many times a user sends a pdf get request, HTTP get request or HTTP post request and their request time. In the network layer for finding TCP-SYN or ICMP flood attack there is an SDN-based gateway that receives traffic coming in and out. When a new incoming data flow arriving at the network server it performs an ML-baseddetection algorithm to decide if the flow is incorrect or attack based on the port's entropy for TCP traffic or logarithm of the number of packets for ICMP traffic. For abnormal flow it blocks the attack flow. At the same time the information of each IP address that arrives in the server network is collected to calculate their entropy [29] [10]. In the application layer the system checks how many requests a server gets from a specific IP address. If the number of requests exceeds the threshold value set by the system, it detects that a specific source is being used to attack the server. In addition, when a request is coming from an address system checks the requested time, when it found any anomaly then the system blocks that IP address for a while [29].

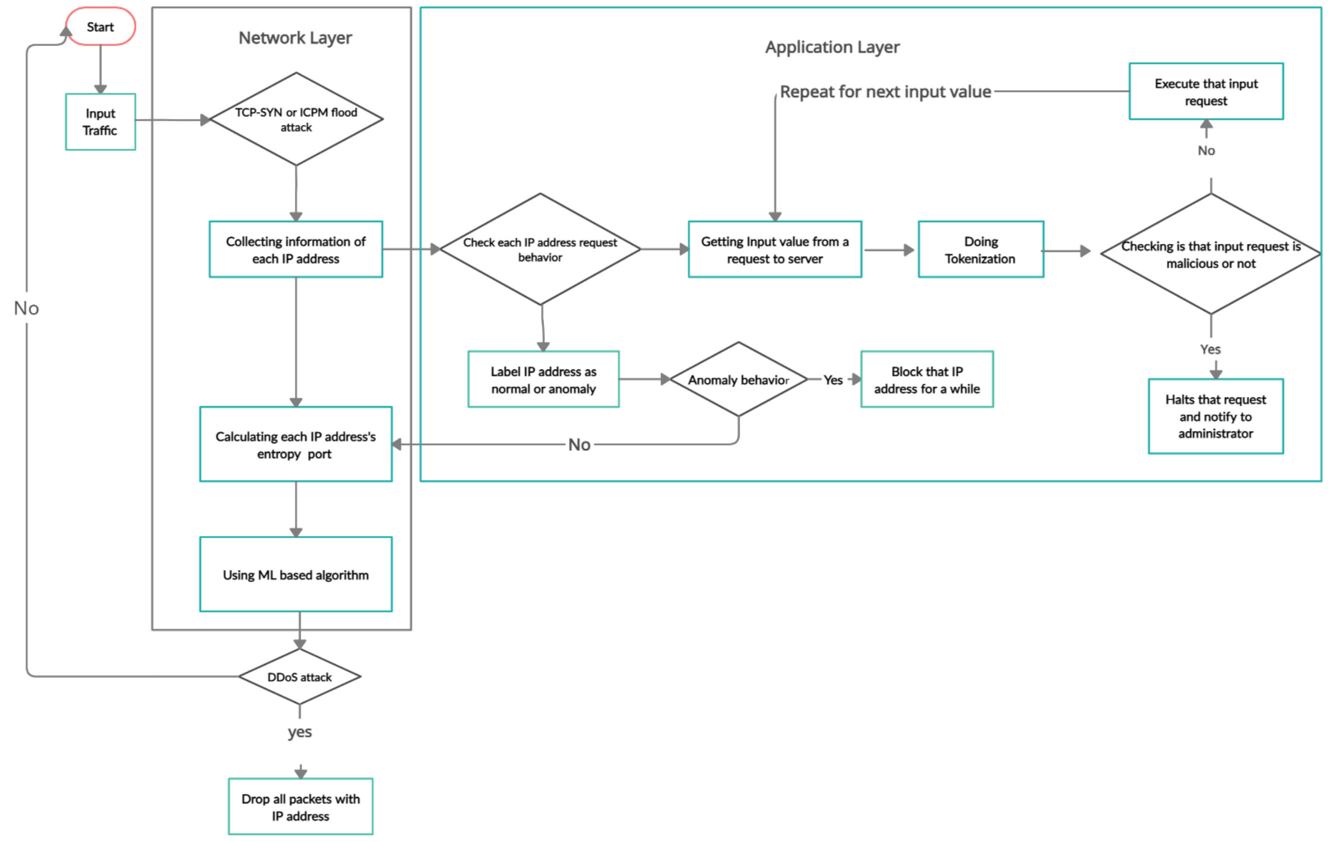
Start
Input Traffic
Network Layer
TCP-SYN or ICPM flood attack
Collecting information of each IP address
Calculating each IP address's entropy port
Using ML based algorithm
DDoS attack
yes
Drop all packets with IP address
No
Application Layer
Repeat for next input value
Execute that input request
No
Check each IP address request behavior
Getting Input value from a request to server
Doing Tokenization
Checking is that input request is malicious or not
Yes
Halts that request and notify to administrator
Label IP address as normal or anomaly
Anomaly behavior
Yes
Block that IP address for a while
No

Fig 2: - Flow Chat of Proposed System

In the second part to detect SQL injection the system checks whether there is any malicious query is sent as a request from a user. The proposed system is trained with a large number of SQL injection attack query patterns. So, when a user sent their input to the server first of all, that input is going to be broken down into a sequence of characters then each of the sequences is passed to the system model to check whether there is any suspicious pattern found or not. If any suspicious input pattern is found then the system halts the input request of that user and notify the administrator [23].

In fig-2 it is shown that when our system starts monitoring it checks incoming traffic in the network layer collecting information of each IP address to calculate their port entropy. After that using an ML-based algorithm checking a TCP-SYN

or ICPM flood attack occurs or not. At the same time when collecting information of each IP address it also monitors the input request behavior means what types of request a user send, how many times network server getting that same user request, determining the time of request between host and server. If the system found any suspicious activity, then that specific IP address inserted into the block list. In addition, with the help of tokenization each input value that comes to the server is compared with a malicious query pattern.

4. EXPERIMENT & STUDY

4.1 Dataset Collection and analysis

For our model training and testing to detect DDoS attacks we select a benchmark dataset which is called NSL-KDD dataset. The NSL-KDD dataset is a variant of the KDDCup'99 dataset that has been configured. In the KDDCup'99 dataset, there are many redundant and duplicate records. For this issue, the NSL-KDD dataset is a new version of the KDDCup'99 dataset that is redundant and duplicate-free, allowing the classifiers to produce better results. This dataset consists of four files. In the NSL-KDD dataset, the attack types are grouped into four categories: DoS, Probe, U2R and R2L. There are 41 features in the NSD-KDD dataset. The dataset can be divided into four categories based on these features [30].

- 4 categorical (Features:2,3,4,42),
- 6 Binary (Features:7,12,14,20,21,22),
- 23 discrete (Features:8,9,15,23-41,43),
- 10 continuous (Features:1,5,6,10,11,13,16,17,18,19)

41 features can also be divided into four different classes, which are Basic, Content, Traffic and Host [31]. The training set of NSL-KDD has a total of 125973 instances and in the test set, it has a total of 22544 instances. NSL-KDD has a version of the dataset with 20% of the training data identified as KDDTrain+_20Percent with a total number of 25192 instances. For the model, NSLKDDTrain+ is used to train the dataset and NSLKDDTest+ is used to test the dataset.

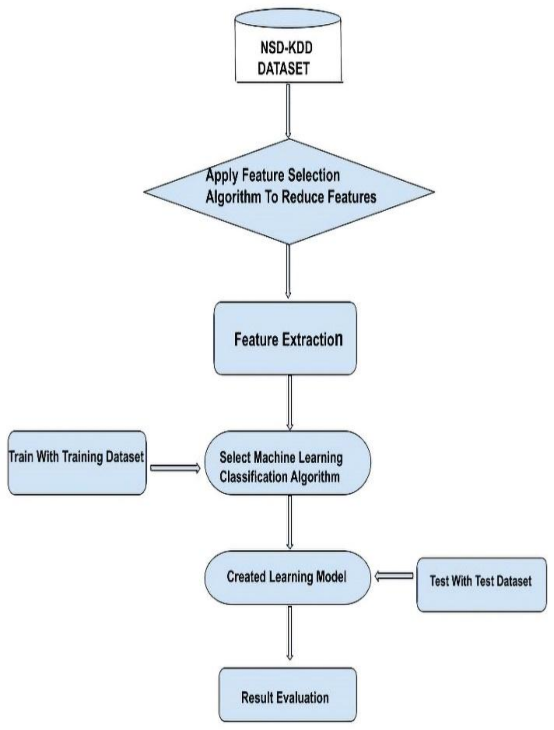
NSD-KDD DATASET
Apply Feature Selection Algorithm To Reduce Features
Feature Extraction
Train With Training Dataset
Select Machine Learning Classification Algorithm
Created Learning Model
Test With Test Dataset
Result Evaluation

Fig 3: -Flow chart of system Model using NSL-KDD Dataset

To evaluate DDoS attacks, the NSL-KDD dataset is used to train and test the model. Three classifier algorithms are used. The algorithms are Random Forest, KNN and Decision Tree. For testing and training, two different files have been used from the NSL-KDD dataset. First, the dataset is normalized and extracted by implementing various selective attribute methods like Infogain methods. Then the model will be trained using the dataset. In **figure 3,** we can see how the model will be trained and tested to be ready for detecting DDoS attacks.

4.2 Feature Extraction

In the NSL-KDD Dataset, there are 41 features. Still, to avoid overfitting of a machine learning algorithm, some features from the dataset must be reduced. It is essential to apply feature reduction techniques to find which features are most important. For selecting the features WEKA TOOL is used and performed many feature selection techniques. From all of them, InfoGainAttributeEval (also called entropy) gives the best result for feature reduction.

So InfoGainAttributeEval with Ranker Search Method is used for each attribute. The output variable varies from 0 to 1.

After the evaluation, attributes with higher information gain values are selected. The arbitrary cutoff is 0.270. All the attributes which info gain value is higher than this cutoff value is selected. So, we found a total of 16 features whose entropy value is higher than the cutoff value. TABLE1 is showing those features, including its class.

Table 1: -Selected Features from dataset including their type

Variable No	*Description*	*Type*
1	Service	Nominal
2	Flag	Nominal
3	Src_bytes	Numeric
4	Dst_bytes	Numeric
5	Logged_id	Nominal
6	Count	Numeric
7	Serror_rate	Numeric
8	Srv_serror_rate	Numeric
9	Same_srv_rate	Numeric
10	Diff_srv_rate	Numeric
11	Dst_host_srv_count	Numeric
12	Dst_host_same_srv_rate	Numeric
13	Dst_host_diff_srv_rate	Numeric
14	Dst_host_serror_rate	Numeric
15	Dst_host_srv_diff_host_rat e	Numeric
16	Dst_host_srv_serror_rate	Numeric

4.3 Result Evolution

In this section, we present the DDoS experimental results derived from different machine learning models with the help of the NSL-KDD dataset.

At first, several selective attribute methods are being implemented in the WEKA TOOL. Then those algorithms are performed on JUPYTER NOTEBOOK to build the model to detect DDoS attacks. The accuracy of three machine learning classification algorithms is compared to see which one gives the best results. The accuracy of the attributes is chosen by performing info gain methods. Cross-validation was done with the training dataset and also a test dataset is used to test the trained model. For our experiment, 16 features are selected. Two categorical features are converted into numeric values for training and testing. Those two nominal features are "service" and "flag".

As their performances are evaluated using various parameters, including accuracy, precision, recall and F1-score, therefore, their given parameters can be calculated using True positive (TP),False Negative (FN), False Positive (FP) and True Negative (TN). Precision and recall give a measure of the relevant data points. It gives a result that demonstrates how good the model is at finding true positives of all possible values. For getting a good percentage of accuracy, a tradeoff between precision and recall is needed. F1-score is the harmonic mean of precision and recall. It helps us to understand which parameters (precision, recall) are more important for our model. A good F1-score indicates a good precision and recall value.

Precision = TP / (TP + FP)
Recall = TP/(TP+FN)
Accuracy = (TP + TN)/(TP+TN+FP+FN)
F1-score = 2*((precision*recall)/ (precision+recall))

Table 2shows the accuracy, precision, recall and F1-score of different machine learning algorithms to cross-validate for train tests. The value of k is set 10 during cross-validation.

Table 2: -Performance of different ML algorithms using cross validation

Algorithm	Accuracy	Precision	Recall	F1-score
KNN	0.997	0.999	1.00	1.00
Random Forest	0.95	0.95	0.94	0.95
Decision Tree	0.996	0.995	0.999	0.999

Table 3 shows the accuracy, precision, recall and F1-score of different machine learning algorithms where the NSL-KDD test dataset is used to test our model.

Table 3: -Performance of different ML algorithms using test dataset

Algorith m	Accuracy	Precision	Recall	F1-score
KNN	0.81	0.83	0.85	0.81
Random Forest	0.75	0.80	0.78	0.75
Decision Tree	0.80	0.82	0.81	0.79

Table 2 and Table 3 shows that KNN gives the best result after the model evaluation. But inTable 2, the performances of the algorithms are high. The accuracy is over 99% of all the algorithms during cross-validation.

But when the model is tested with the NSL-KDD test dataset, the performance and accuracy rate are low compared with the result of table 2. The confusion metrics of all the models are shown in the below figures. Those confusion metrics between testing target output and testing predicted outcome. TheX-axis shows predicted values and Y-axis shows actual values of the dataset. In the given heatmap's value of the top left box is True Negative (TP), the value of the top right box is False Positive (FP), the value of the bottom left box is False Negative (FN), the value of the bottom right box is True Positive (TP).

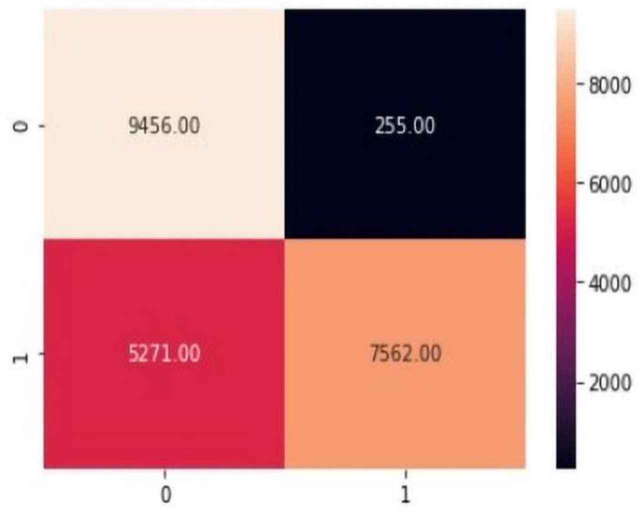

Fig 4: Confusion metrics of Random Forest

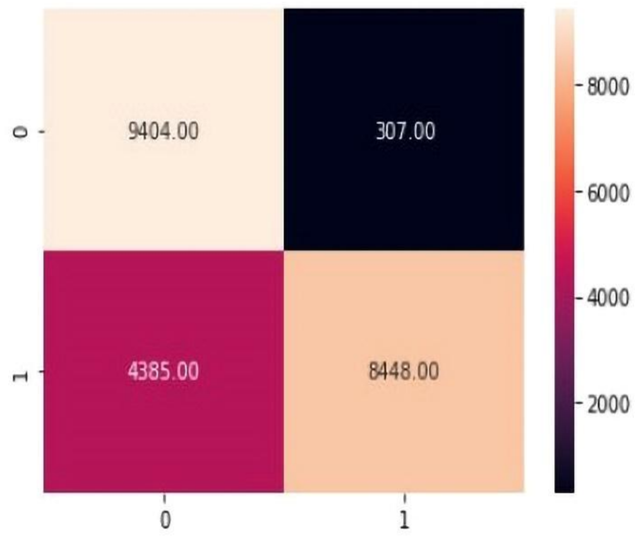

Fig 5: Confusion metrics of Decision tree

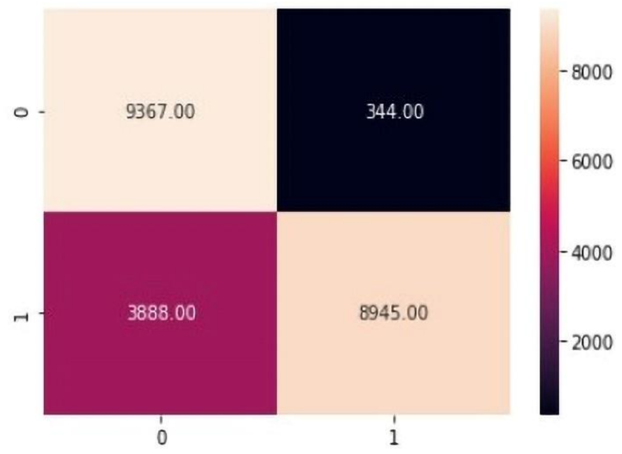

Fig 6: Confusion metrics of Knn

From the above heatmaps, 100% of the total testing dataset model can predict 41% true positive value, 2% false positive value, 40% true positive value and 17% false-negative value.

From the above calculation, this can be observed that the False-negative rate is high when the model is tested with the test dataset. For this, the performance rate is very poor. The performance ratio and accuracy can be higher if we minimize the value of false negative. By minimizing false-negative, the performance ratio and accuracy along with other parameters can be maximized.

4.4 System Properties

Those experiments are done on Intel(R) Pentium(R) CPU G4560 @ 3.50GHz 3.50 GHz with 8GB memory and Intel(R) Core (TM) i5-6200U CPU @ 2.30GHz 2.40 GHz with 12GB memory.

5. FUTURE WORK

A concept of SQL Injection attack detection model will be considered in conjunction with this model in the future. Where both attacks (SQL Injection and DDoS) can be prevented by the system. As a result, tests will be run on the SQL datasets consisting of both malicious and non-malicious queries. Where tokenization method will be used to split queries into tokens offeatures, which is the process of breaking the queries into meaningful elements called tokens. Multiple Machine learning algorithms will be used to train the model with these datasets for comparing the results to find the best possible algorithm to combine with role-based access control for classification for the most accuracy and precision while detecting SQL injection attacks.

6. ACKNOWLEDGMENTS

This paper is guided and supported by NazmusSakib. Assistant professor of the Department of Computer Science and Engineering, Ahsanullah University of Science & Technology.

7. CONCLUSION

The rate of cybercrime is increasingly rising. As a result, cybersecurity must be improved. In the majority of instances, the IDS fails to detect these attacks. On the other hand, cyber hackers seem to get through numerous security systems to combat these assaults. Moreover, not many IDS can simultaneously detect multiple types of attacks like DDoS and SQL injection attacks. This suggested system is one of the ways to detect DDoS and SQL injection attacks simultaneously, SQL injection attacks. This suggested system is one of the ways to detect DDoS and SQL injection attacks simultaneously, which will be very helpful to tackle cyberattacks. In this paper, several machine learning techniques are used to detect DDoS attacks. In our experiment, the Knn algorithm, which given us higher accuracy, will be implemented in our system.

8. REFERENCES

[1] M. K. Pratt, 2021. [Online]. Available: https://searchsecurity.techtarget.com/definition/cyber-attack.

[2] J. Fruhlinger, 27 February 2020. [Online]. Available: https://www.csoonline.com/article/3237324/what-is-a-cyber-attack-recent-examples-show-disturbing-trends.html.

[3] "CLOUDFLARE," [Online]. Available: https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/.

[4] M. F. Y. Zainab S. Alwan, "Detection and Prevention of SQL Injection Attack: A Survey," vol. 6, no. 8, 2017.

[5] "OWASP," [Online]. Available: https://owasp.org/www-community/attacks/SQL_Injection.

[6] J. Thakkar. [Online]. Available: https://sectigostore.com/blog/ddos-attack-statistics-a-look-at-the-most-recent-and-largest-ddos-attacks/.

[7] N. Anthony Chadd, "DDoS attacks: past, present and future".

[8] J. Vijayan. [Online]. Available: https://www.darkreading.com/attacks-breaches/sql-injection-attacks-represent-two-third-of-all-web-app-attacks/d/d-id/1334960.

[9] R. A. Manjula Suresh, "Evaluating Machine Learning Algorithms for Detecting DDoS Attacks".

[10] P. H. H. Nguyen Ngoc Tuan, "A DDoS Attack Mitigation Scheme in ISP Networks Using Machine Learning Based on SDN," 2020.

[11] T. X. Y. Mahjabin and G. J. Sun, "A survey of distributed denial-of-service attack, prevention, and mitigation techniques," vol. 13, no. 12, 2017.

[12] A.M. Christos Douligeris, "DDOS ATTACKS AND DEFENSE MECHANISMS: A CLASSIFICATION".

[13] J. Kuhns, "Global Information Assurance Certification Paper," 2001.

[14] 2001 CERT Advisories, CERT Division.

[15] D. Dittrich, "The DoS Project's 'trinoo' distributed denial of service attack tool".

[16] P. J. Criscuolo, "Distributed Denial of Service," 2000, p. 8.

[17] A.M. Christos Douligeris, "DDOS ATTACKS AND DEFENSE MECHANISMS: A CLASSIFICATION".

[18] S. S.Lakshminarasimman, "Detecting DDoS Attacks using Decision Tree AlgorithmAlgorithm," 2017.

[19] P. T. H. a. K. T. Khaing, "Detection Model for Daniel-of-Service Attacks using Random Forest and k-Nearest Neighbors," 2013.

[20] T. A. D. K.R.W.V.Bandara, "Preventing DDoS attack using Data mining Algorithms," 2016.

[21] A.K. D. G. a. D. N. H. T. Sagar Pande, "DDOS Detection Using Machine Learning Technique".

[22] A.M. Y. J. M. R. Kazi Abu Taher, "Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection," 2019.

[23] S. Mishra, "SQL Injection Detection Using Machine Learning," 2019.

[24] W. Aorso, "AMNESIA Analysis and Monitoring for Neutralizing SQL-Injection Attacks," in IEEE and ACM, 2005.

[25] D. a. M. V.Haldar, "Dynamic Taint Propagation for Java," 2005.

[26] S. SINGH. [Online]. Available: https://www.analyticsvidhya.com/blog/2019/07/how-get-started-nlp-6-unique-ways-perform-tokenization/.

[27] S. B. Anurekh Kumar, "Use of Query Tokenization to Detect and Prevent," IJSTE - International Journal of

Science Technology & Engineering, 2015.

[28] J. Xu. [Online]. Available: https://towardsdatascience.com/how-to-detect-mean-tweets-with-machine-learning-deaa9dc6a8a8.

[29] A.S. Lohit Barki, "Detection of Distributed Denial of Service Attacks in Software Defined Networks," 2016.

[30] G. Saporito. [Online]. Available: https://towardsdatascience.com/a-deeper-dive-into-the-nsl-kdd-data-set-15c753364657.

[31] P. Aggarwal, "Analysis of KDD Dataset Attributes - Class wise For Intrusion".