American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

DISTRIBUTED DENIAL OF SERVICE (DDOS) DETECTION AND PREVENTION USING J48 ALGORITHM

Semester: Summer 22-23
Section: C
Network Security Project

Submitted to
Dr.MD MEHEDI HASAN

Submitted By

SN	Student Name	Student ID
1	*ANIK, MD. IMRAN AHMED*	*19-39927-1*
2	*GAZI MD.JUBAYAR HOSSAIN*	*19-40016-1*
3	*HARUN, MUHAMMAD BIN*	*19-41580-3*
4	*MD.NAFIZ GAFFAR*	*19-41731-3*
5	*SUKANNA, JOYSREE DEY*	*20-41946-1*
6	*BARI, SAMIHATUL*	*20-42018-1*

Abstract

The DDoS (distributed denial of service) assault is one of the most dangerous cyberattacks in the contemporary technological era. The use of the internet is becoming more and more prevalent. Too many cyberattacks are occurring. As a result, a lot of numerical information is lost. In this study, we offer an approach for both the quick identification and avoidance of attacks. For machine learning, we opt for the J48 (Decision Tree), also referred to as the C4.5 approach. The J48 approach has the greatest classification rate of all machine learning classifiers. There is support for both categorical and continuous data. Attacks will be identified and then filtered using the method we suggested in our proposed model, which employs training data in the detection phase of the j48 algorithm to enter the classification phase, where it classifies the data type (which data are under a DDoS assault and which are regular data), and then enters the testing data phase. After identifying the material, we will rapidly halt a DDoS onslaught. Other classifiers like KNN and Nave Bayes are outperformed by Random Forest J48 in terms of accuracy. We can assume that the methodology we've proposed is the most accurate for spotting and preventing DDoS attacks.

Acronym	Meaning
ANN	Artificial Neural Networks
C & C	Command and control
DDoS	Distributed Denial of Service
HTTP	Hyper Text Transfer Protocol
HD	Hellinger Distance
ICMP	Internet Control Message protocol
IDS	Intrusion detection system
IP	Internet Protocol
KNN	K-Nearest Neighbor
ML	Machine Learning
MLP	Multilayer Perceptron
SVM	Support Vector Machine
SDN	Software-Defined Network
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
TPR	True Positive Rate
FPR	False Positive Rate

Table 1A: List of abbreviations

Table of Contents

List of Tables

List of Figures

Part 1: Introduction

Due to how quickly modern technology has developed, people cannot exist without the internet or other forms of technology. Natural laws dictate that everything, no matter how lovely, must have some defects. An online assault is one of these. People's ignorance of the risks connected with utilizing the internet is one of the main factors. Recent changes have altered the criteria that attackers use to choose their victims. "Distributed denial of service" refers to the situation where users are unable to access the system because of network congestion. Natural laws dictate that everything, no matter how lovely, must have some defects. An online assault is one of these. People's ignorance of the risks connected with utilizing the internet is one of the main factors. Recent changes have altered the criteria that attackers use to choose their victims. A distributed denial of service occurs when users are unable to access the system because the network is clogged with unnecessary packets. Natural laws dictate that everything, no matter how lovely, must have some defects. An online assault is one of these. People's ignorance of the risks connected with utilizing the internet is one of the main factors. Recent changes have altered the criteria that attackers use to choose their victims. When a network is overwhelmed with unnecessary packets, attacks known as distributed denial of service (DDOS) occur, preventing users from accessing the system. Regular users are unable to access the system since hackers are frequently to blame for doing so by sending an unusually large volume of packets. Each system has a cap on the number of users who can sign in simultaneously. As long as the hackers are sending data, the system is being shut down. Security is weakened and the security level is breached when attacks happen for the first time; as a result, HTTPS is replaced with HTTP, where the s stands for the security level. The user of the server believed that the server may have crashed, but a widespread denial-of-service attack was actually to blame. The loading of websites takes a very lengthy time. On really important governmental and educational websites, it is a very dangerous practice. The likelihood of the data being lost is high. The number of hampers produced by the government has increased. Attacking bank websites raises questions about national security because the system is disrupted and a sizable sum of money is lost. An attacker can launch a distributed denial-of-service attack by seizing control of one or more of the millions of publicly accessible computer systems on the internet [1]. The first distributed denial-of-service assault, known as Panix, was launched in 1996 after a syn. Flood rendered the system unusable [2].

Accessibility is a vital element of security features (DdoS), and reducing accessibility is the main goal of a distributed denial-of-service attack. It accomplishes this by denying a victim or system the resources they need.[3] Every single distributed denial-of-service attack is carried out differently. [4] [5] After being delivered to the victim's or users' targeted systems, processing packets eats up a lot of system resources and bandwidth, which makes the system a target for DdoS attacks.[6] [7] The system then becomes illogical, stops accepting packets, and rejects all incoming requests. Assaults can occur in a variety of ways, as was already mentioned. The various DdoS assault types each have their own distinct set of properties and characteristics. IP spoofing and Transmission Control Protocol (TCP) flooding are two examples of this in use with the Internet Protocol. SYN Flood With Spoofing IP, User Datagram Protocol (UDP) Flooding, Internet Control Message Protocol (ICMP) Flooding, Hyper Text Transfer Protocol (HTTP), Get/Post, and Ping of Death are only a few examples of the countless DDoS attacks.[8–13].

The range of possible assaults is widening, as was previously mentioned. The first half of 2021 saw 25% more assaults than the same period in 2020, according to a report that was posted on Azure [14]. In the second half of 2021, Azure successfully fends off about 359,713 strikes from Skywards. DdoS attacks will continue to be brief, lasting 74% of the time under 30 minutes and 87% of the time under an hour, according to the statistics [14]. This indicates that attacks will probably still be intermittent. It takes over 25.3 billion searches to successfully prevent one assault, according to Imperva's data. This creates an updated standard for their risk management approach [15]. Imperva was able to effectively fend off a large attack that lasted four hours and peaked at 3.9 million RPS on June 27, 2022. (Applications each second). This rate is a lot higher than usual. It frequently has an RPS of 1.8 million on average. Multiple requests can be delivered over a single connection thanks to HTTP/2 multiplexing, which is used by attackers [15].

1.1 DdoS attacks

Although maintaining a clear separation between the control and data planes of the SDN network has many advantages, it also creates a new issue because it leaves the network more open to various threats. The ability of one of these attacks, known as Dos or Distribute Dos, to have catastrophic effects on an SDN network has been demonstrated. [16] defines a distributed denial-of-service attack as "an attack on a server in which a large number of packets are sent to cause an outage or degradation of service for legitimate users or to deprive an organization of necessary computer services, such as Internet access, email, on-premise, hosted, or cloud services". This definition states that a server attack occurs when a large number of packets are sent with the intention of interfering with or degrading service for authorized users. varieties and traits of DDOS By dispersing a lot of data packets over the network, a distributed denial of service assault aims to overwhelm it. To overwhelm the victim network, DdoS attacks use a variety of techniques, including TCP, UDP, ICMP, Random IP, and Botnets.

1.2 List of several kinds of DdoS attacks

1.2.1 TCP Flood

The most frequent DDoS assault is a TCP flood. TCP connection requests can be sent in high numbers while ignoring the victim server's SYN-ACK in TCP flood attacks. The destination server has a large number of connections open but only partially. These unfinished connections take up all or most of the system resources, so users that have access cannot access them.
[17]

1.2.2 Flooding of ICMP

An ICMP flood attack, also referred to as a smurf attack, is another type of distributed denial of service attack. This attack includes sending a large number of ICMP packets with fictitious source IP addresses to the target quickly. The ICMP answers provided by the compromised server will unintentionally be received by the owner of the fake IP address. This will result in decreased accessibility and performance for both the affected server and the real person who is the owner of the fictitious IP address [17].

1.2.3 Flooding of UDP

The third category of distributed denial-of-service attack is a UDP flood attack. They shoot a ton of UDP packets at the target. This kind of assault consists of the DNS amplification attack, which copies the victim's original IP address and sends a quick query to the DNS server. The victim's performance declines due to the DNS server's lengthy responses. Using a high quantity of UDP packets to attack the victim, the UDP flood attack can also be used to make the target unreachable to regular users [17]. By doing this, the victim's machine's accessibility to common users is restricted.

1.2.4 IP Flood of Randomness

Sending random IP packets can also be used to execute a DDoS attack, keeping the controller occupied while preventing it from responding to legitimate traffic [20]. The distributed denial of service attack creates a lot of malicious packets over time and can happen frequently at a set time of day [18]. For instance, it occurs each day at 7:00.

1.2.5 Botnets

DDOS attacks based on botnets are more harmful due of their sophistication. Army-sized clusters of compromised machines are known as botnets [16]. Many simple assault generation tools are available for little or no cost, if at all. Anyone can easily find the means or hire others to carry out any type of attack via the Internet because the manufacture of assaults is a lucrative industry. When malicious software is placed on a computer using unethical methods, a botnet is created. Users may be tricked into providing personal

information by SPAM, downloading files, clicking on links to websites, or receiving phishing emails. Through the infected workstation, the virus connects with the command and control server of the botnet's owner. The C&C server issues commands to the hundreds of infected computers, telling them to use peer-to-peer communication to attack the victim's servers and network. The C&C server disseminates these directives. The breadth and power of the botnet may be increased by combining infected devices' capacity to generate and transmit enormous volumes of attack data. The intrusion detection system, or IDS, is one of the most well-known remedies for the issue of DDoS attacks. It has the capacity to repel attacks [1]. The absence of readily accessible datasets is the main obstacle to preventing DDOS assaults. [1]

Cyberattacks have significantly increased in frequency and sophistication in recent years. Attacks like DDOS serve as a catalyst for cyberattacks. DDOS attacks interfere with server and network resources. Our technology instantly recognizes an attack, enabling us to lessen potential damage. The divide-and-conquer strategy is used in the J48 methodology, which works from the top level down. The J48 technique, which is used to classify a number of applications, may result in accurate classification outcomes when properly used.

One of the most potent machine learning algorithms is the J48 technique because it can evaluate data in both continuous and categorical modes. With J48, it is possible to make precise predictions and it could also help with comprehending the patterns in the data. Furthermore, precise forecasts might be made using the data. It addresses problems with numerical features, missing data, pruning, calculating error rates, the challenge of decision tree induction, and the generation of rules from trees. The j48 technique is faster and more accurate at detecting distributed denial-of-service assaults. Attacks involving distributed denial of service are discovered using data mining. Several scenarios, including TCP flooding, IP-faked SYN flooding, and UDP flooding, have been used to test the proposed method. These bugs gathered information about network traffic. The data was categorized using Xero, OneR, Naive Bayes, Bayes Net, Decision Stump, and J48. According to several of these techniques, J48 has the highest level of categorization accuracy. Our results show that the suggested strategy is essential for identifying DDOS assaults. In the real world, assault detection will be easier to do and more effective with the aid of contemporary technologies. DDOS attacks are detectable and countered by J48.

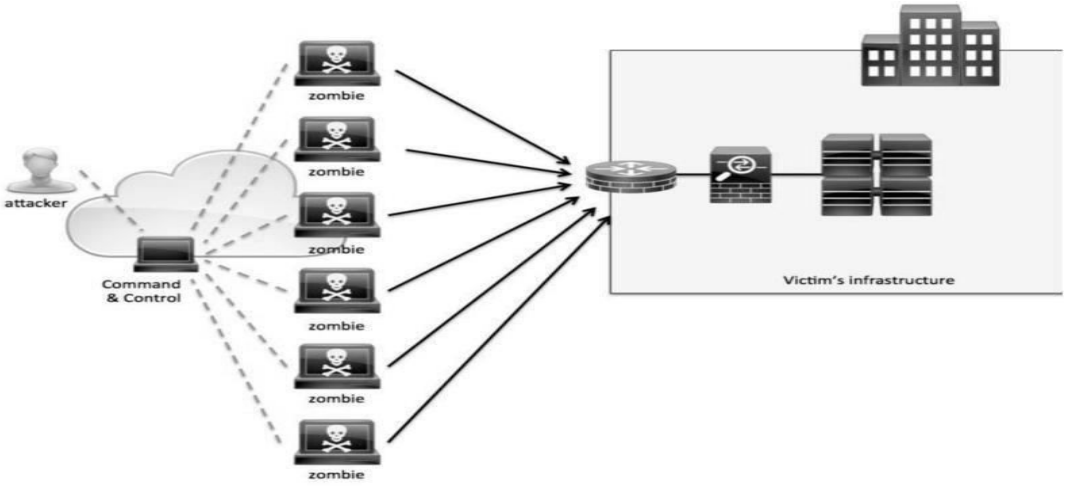
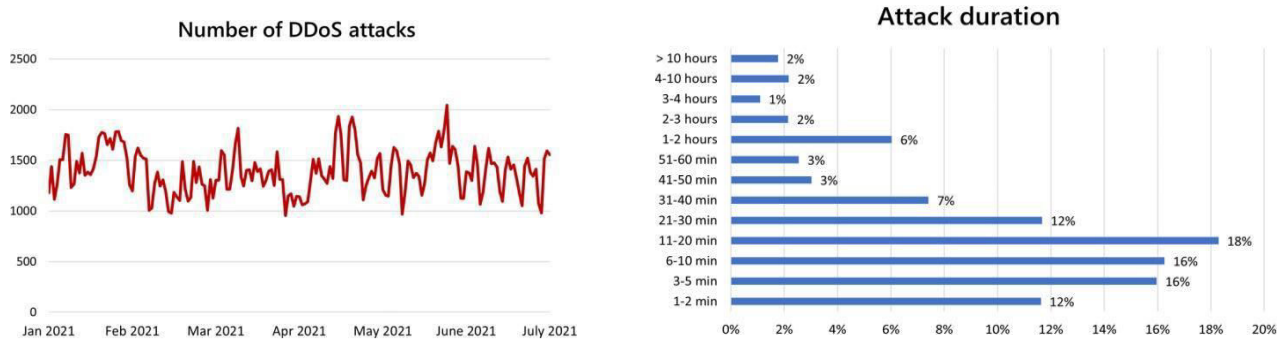


Figure 1: Distributed Denial of Service Attack [33].



A: Number of Attacks B: Duration

Figure 2-3: DDOS statistics by Azure [14].

Part 2: Background and related work

Numerous machine learning techniques have been employed in research endeavors focused on detecting Distributed Denial of Service (DDoS) attacks. These algorithms encompass a range of options, including Naive Bayes, K-Nearest Neighbors (KNN), Random Forest, Support Vector Machine (SVM), Decision Tree (C4.5/J48), and Multi-Layer Perceptron (MLP). The primary objective is to enhance the precision of DDoS attack detection.

In terms of performance evaluation, Ismanto et al.'s study reveals that the C4.5 decision tree (J48) algorithm outperforms its counterparts, demonstrating superior accuracy percentages and quicker training times. Specifically, C4.5 achieved an impressive accuracy rate of approximately 99.05%, surpassing the Naive Bayes method.

The classification of DDoS attacks is divided into two main categories: those that target bandwidth and those that target system resources.

Researchers have leveraged various datasets, including the KDD and NSL KDD datasets, to fuel their DDoS detection efforts. Diverse machine learning methodologies have been explored to enhance accuracy, encompassing incremental clustering, Principal Component Analysis (PCA), and Artificial Neural Networks (ANN).

One innovative technique involves the application of the Hellinger Distance (HD) method. This approach compares traffic analysis with baseline and incoming requests to identify potential DDoS attacks. The method has demonstrated remarkable success, achieving a high rate of accurate classification between legitimate and DDoS-related packets.

Software-Defined Networking (SDN) presents an intriguing avenue for DDoS detection. Researchers, such as Banitalebi Dehkordi et al., have proposed a comprehensive approach involving data collection, entropy-based categorization, and analysis. Their experiments, conducted using datasets like UNB-ISCX, CTU-13, and ISOT, indicate that this approach surpasses competing methods.

O. Rahman et al. have conducted research in which they evaluated the effectiveness of J48, Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) in identifying and mitigating DDoS attacks within an SDN network. Their findings favored the J48 algorithm as the top performer in both training and testing scenarios.

In a distinct research effort aimed at safeguarding wireless network nodes, Lakshminarasimman et al. devised a unique method utilizing the Decision Tree approach. In this endeavor, J48 and Random Forest, well-known data categorization techniques, were combined to classify incidents into various types of assaults. Here, the J48 technique exhibited superior performance over the random forest decision tree algorithm.

Narasimha Mallikarjunan et al. introduced an innovative machine learning-based anomaly detection technique for network security. Their method stands out for its remarkable accuracy, outperforming Naive Bayes and random forest algorithms in terms of precision. Additionally, this technique maintains a low false-positive rate.

As Software-Defined Networking (SDN) networks continue to evolve, the separation of the data plane and control plane introduces new security challenges. DDoS attacks targeting SDN controllers are on the

rise, necessitating enhanced security measures. Vieira et al. conducted an extensive literature review to address these concerns, highlighting the vulnerability of SDN controllers to DDoS attacks. They emphasize the need for improved DDoS defenses in SDN networks.

Part 2A: Classifier Accuracy

In a study referenced as [37], the authors employed various methods to counteract a DDoS attack on a Software Defined Networking (SDN) network. SDN, known for its advantages such as scalability, flexibility, monitoring, and innovation simplicity, requires robust security measures. Among the most perilous threats to SDN is the DDoS attack. The study utilized machine learning techniques, specifically J48, Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbors (K-NN), to detect and mitigate DDoS attacks within the SDN network. The research identified J48 as the most effective machine learning model, excelling in both training and testing times.

In a separate study referenced as [38], the authors proposed a hybrid approach for identifying DDoS attacks, combining SDN network security with statistical analysis and machine learning. SDN, valued for its scalability, flexibility, monitoring capabilities, and ease of innovation, serves as the foundation for this strategy. Implementing machine learning techniques significantly improved the accuracy of detection, increasing it from 87/88% to an impressive 99.86%. The success of this approach was contingent on experimental data sets, outperforming existing methods.

In another study referred to as [39], the authors recognized Distributed Denial of Service (DDoS) attacks, specifically SYN flood attacks, as a severe security threat. Their machine learning algorithms aimed to detect SYN flood attacks, and their performance was assessed. A classification model was trained and tested using a telecom network packet capture dataset created with Hping3 and Wireshark tools, and the Weka data mining tool was employed for implementation. Among the algorithms tested, including J48, AdaBoost, Naive Bayes, and ANN, J48 exhibited the highest accuracy at 98.57%, making it a suggested method for SYN attack detection.

The authors of [40] developed machine learning techniques to safeguard cloud computing against DDoS attacks. Cloud computing is known for its cost-effectiveness, but it also needs protection from DDoS threats. The authors utilized Support Vector Machine, Naive Bayes, and Random Forest classification methods for DDoS prevention, with Support Vector Machine identified as the most effective defense against DDoS attacks.

In [41], a researcher enhanced the J48 algorithm to boost the efficacy and precision of an Intrusion Detection System (IDS). The primary aim of an IDS is to detect and trigger alerts. The updated J48 algorithm aimed to enhance detection accuracy and performance within the IDS system.

Furthermore, [42] addressed the challenge of managing large volumes of internet traffic during DDoS attacks in cloud computing. The author employed a feature selection approach to preprocess datasets for attack classification. According to their findings using the NSL-KDD dataset, the EMFFS technique combined with 13 features outperformed other feature selection methods from the literature and individual filter feature selection methods when using the J48 classifier.

Part 3: Propose Model

3.1: Detection and Prevention

As technology evolves, Distributed Denial of Service (DDoS) attacks have become more complex, posing significant challenges for organizations. Network traffic often contains a mix of legitimate and malicious content, necessitating constant monitoring and analysis to identify policy violations and safeguard against

attacks [44]. IP spoofing makes blocking the attacker's IP address ineffective, shifting the focus towards understanding the attack's characteristics. Distinguishing DDoS attacks from regular traffic is challenging due to their similarities, but common traits include malicious packets with the same destination and port addresses, as well as differences in packet sizes compared to regular traffic [43].

Organizations employ various preventive measures like firewalls, access control lists, antivirus software, and Intrusion Detection Systems/Intrusion Prevention Systems (IDS/IPS) to thwart unauthorized access [45]. Timely detection of malicious behavior is crucial, emphasizing the importance of IDS speed, accuracy, and reliability [44]. Recognizing the need for machine learning techniques, particularly in Software-Defined Networking (SDN), the research community is exploring effective IDS solutions [46].

Machine learning involves the development of computer programs capable of learning from data without explicit programming [18]. It relies on using data samples to establish connections between inputs and outcomes [10]. Machine learning algorithms can be employed to differentiate DDoS attacks from normal traffic, utilizing various indicators such as packet count, average packet size, bytes, packet and bit rates, among others [44]. In this context, our research primarily focuses on the J48 classifier for anomaly detection and prevention.

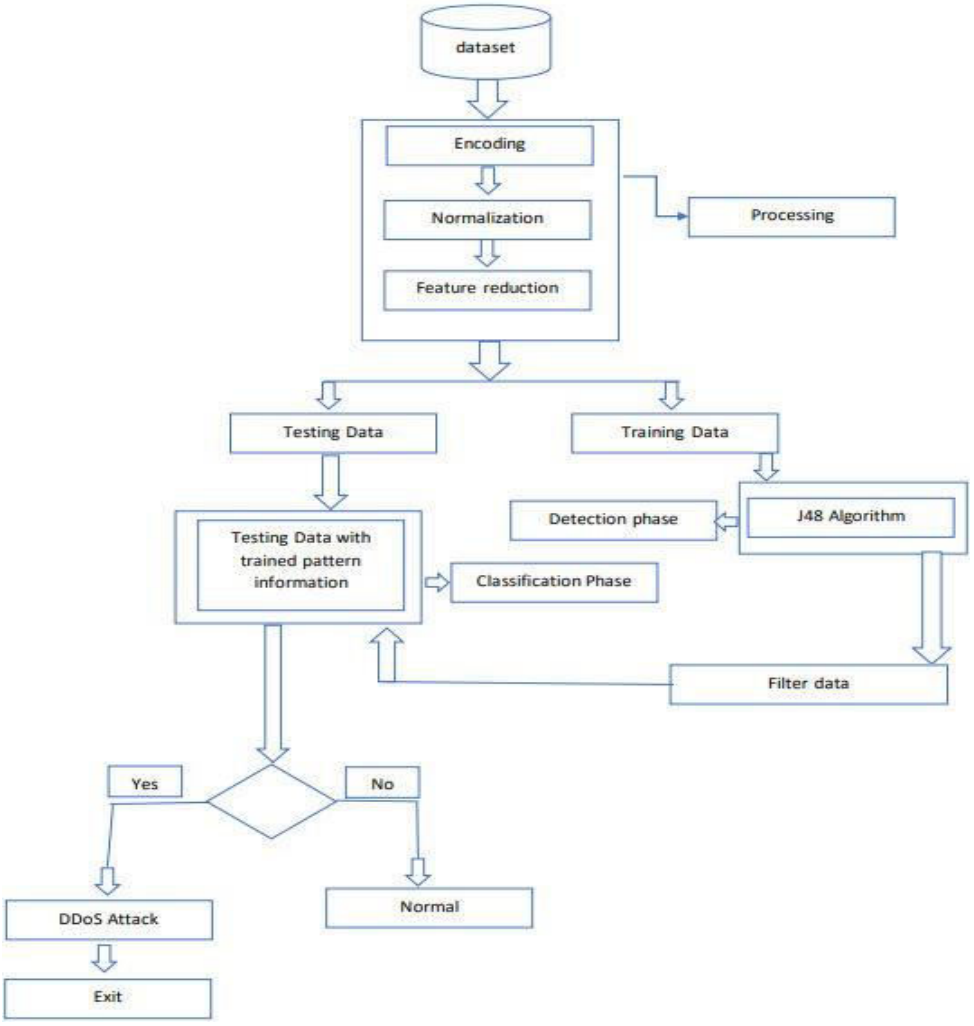


Figure 4: The proposed system model

To defend against DDoS attacks, the initial step is to distinguish normal datasets from those under attack. This involves data gathering, selection, and analysis. Subsequent data processing includes encoding, normalization, and feature reduction. Following these steps, training and testing are conducted, involving the detection and filtration of datasets using the J48 classifier during the training phase. J48 contributes to enhanced detection accuracy and dataset protection against DDoS attacks. The model employs WEKA and machine learning algorithms for detection and prevention, with the primary goal of strengthening prevention measures. This approach can effectively identify DDoS attacks, and J48 plays a key role in improving accuracy. The methodology is versatile and adaptable to various types of DDoS attacks, offering robust security enhancements for networks and servers.

DDoS attacks pose a severe threat, and numerous machine learning algorithms are deployed for their prevention and detection. To address the escalating number of server attacks, accuracy is of utmost importance. In this regard, our proposed model excels, surpassing other machine learning techniques in accuracy. The model primarily relies on J48 algorithms to enhance precision in DDoS attack detection and protection. In our experiment, we developed a recommended model, which performed well using Kali Linux. While we also explored the K-Nearest Neighbors (KNN) approach and the Naive Bayes algorithm, J48 proved to be themost effective in our testing. We successfully detected DDoS attacks using both continuous and categorical datasets, improving accuracy by transforming missing datasets into continuous types. The data sets underwentthorough processing in three phases to prepare them for training and testing.

Part 4: Methodology

To detect and prevent Distributed Denial of Service (DDoS) attacks, we propose a model illustrated in Figure 4. Our suggested model comprises several stages, beginning with data acquisition. Subsequently, we move to data pre-processing, which includes data encoding and normalization. After normalization, we proceed to the Feature Reduction stage, utilizing only the relevant dataset.

Within our system, we work with two types of datasets:
1) Training Dataset
2) Test Dataset

4.1Training Dataset:

The training dataset is the data used to train our machine learning algorithm and model. It necessitates human intervention for exploration, analysis, and processing, crucial for the J48 machine learning method. Training data can come in various forms, such as text, images, videos, or audio. The quality and quantity of training data significantly influence the accuracy and success of a machine learning model. Our J48 algorithm relies on training data to identify patterns in the dataset, enabling accurate predictions.

4.2Test Dataset:

Once we create the model using the training dataset, we need to evaluate its performance with unseen data, known as the test dataset. Test data is utilized to assess the model's effectiveness and improvements. It aids in enhancing results and accuracy. The test dataset serves as an evaluation of the model's performance, comparing it to the training dataset. The accuracy of the model is also influenced by the quality of the training dataset. Typically, the test dataset is smaller than the training dataset, but it is essential for meaningful predictive testing. Splitting the dataset into training and test datasets is a crucial step in data pre-processing, allowing for better performance, increased contribution, and predictability. It is vital to prevent overfitting and underfitting, common issues in machine learning, by using techniques like cross-validation, early termination, and regularization.

4.3How J48 Works:

The J48 machine learning approach proved effective for both continuous and categorical datasets. J48 is widely used in various fields for data classification. We employed the J48 algorithm to detect, block, and improve the accuracy of DDoS attacks. The J48 machine learning method consists of two fundamental stages: classification and learning.

4.4Advantages and Disadvantages of the J48 Algorithm:

J48 offers advantages such as improved data accuracy, pattern recognition, and the ability to handle numerical features, missing values, pruning, and projected error rates. It excels in identifying and stopping DDoS attacks and can work with both categorical and continuous data. However, a significant drawback is its inability to handle missing values effectively. To address this issue, the Naive Bayes algorithm is preferred for datasets with missing values, as it can handle such cases efficiently. The dataset used for classification contains diverse data types, and the J48 algorithm aids in evaluating the model's performance.

4.5 Machine Configuration for Prevention:

4.1.1 Attack Part:

We initiate the attack using the Xerxes DoS tool, which automates Denial of Service (DoS) attacks. The Xerxes program, developed by hacker The Jester (th3j35t3r), allows us to execute multiple independent attacks against various target sites without requiring a botnet.

The attack is executed using the Kali Linux operating system and the Xerxes DoS program, targeting a website powered by the Ubuntu Linux operating system.

4.1.2 Prevention Part:

To stop the attack, we input the prevention command in the terminal. We start by identifying the attacker's IP address responsible for the attack, and subsequently, we block it. This action effectively halts the DDoS attack initiated by the hacker.

In summary, our proposed technique is designed to swiftly identify and mitigate DDoS attacks with a focus on minimizing data loss. We believe that our model, centered around the J48 algorithm, achieves the highest accuracy rate in detecting and mitigating DDoS assaults.

Detection & Prevention System:

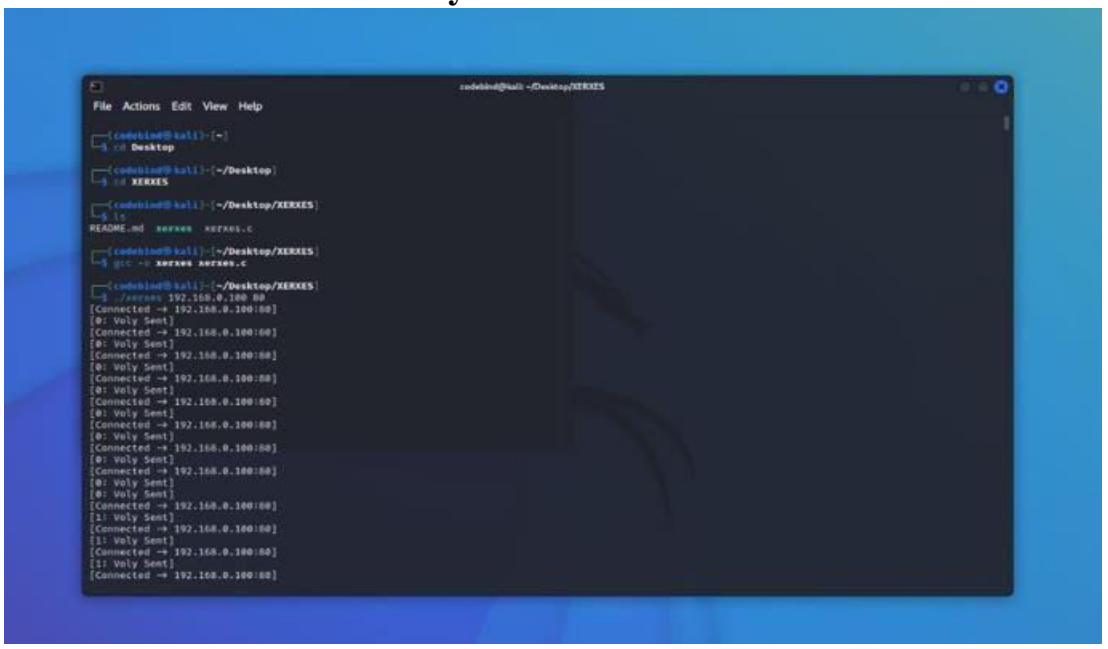


Figure 5: DDOS attack using kali Linux

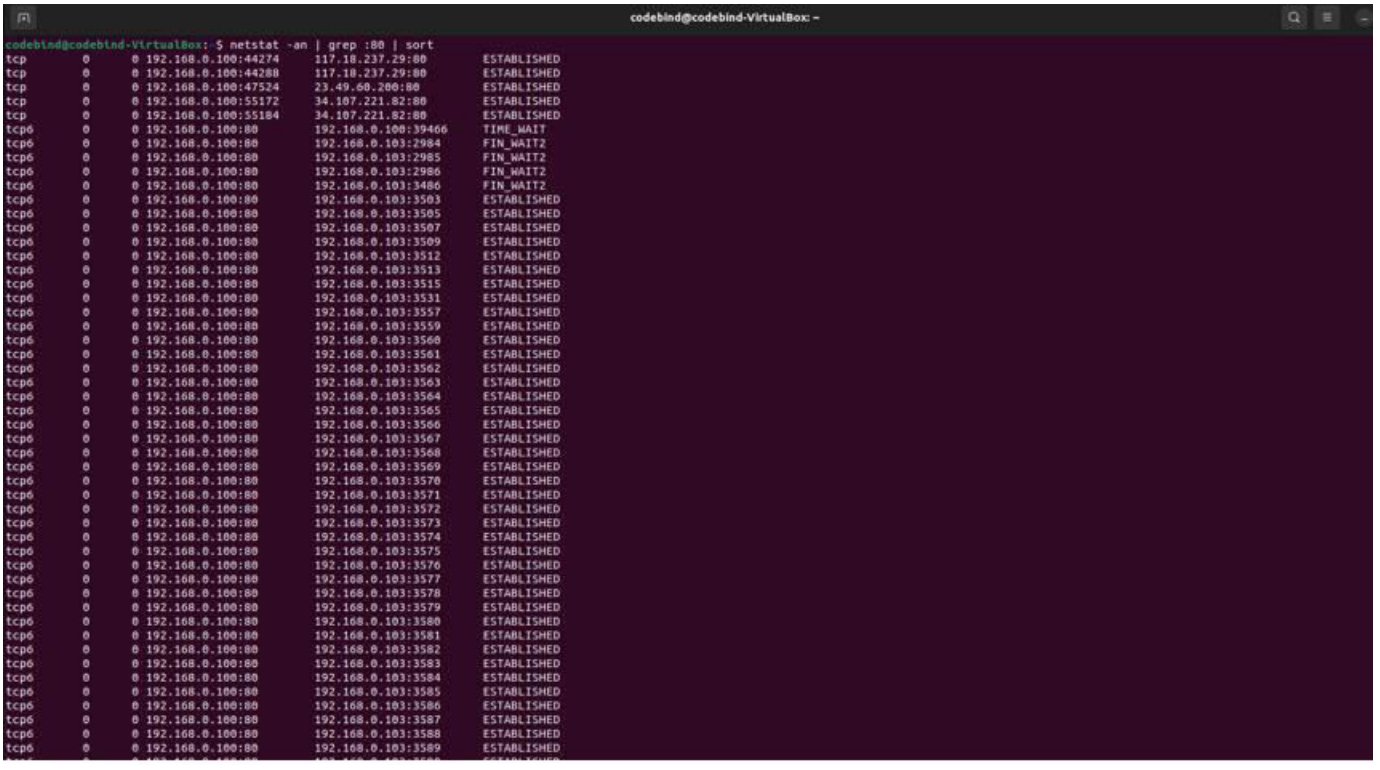


Figure 6: Packets send into the selected site

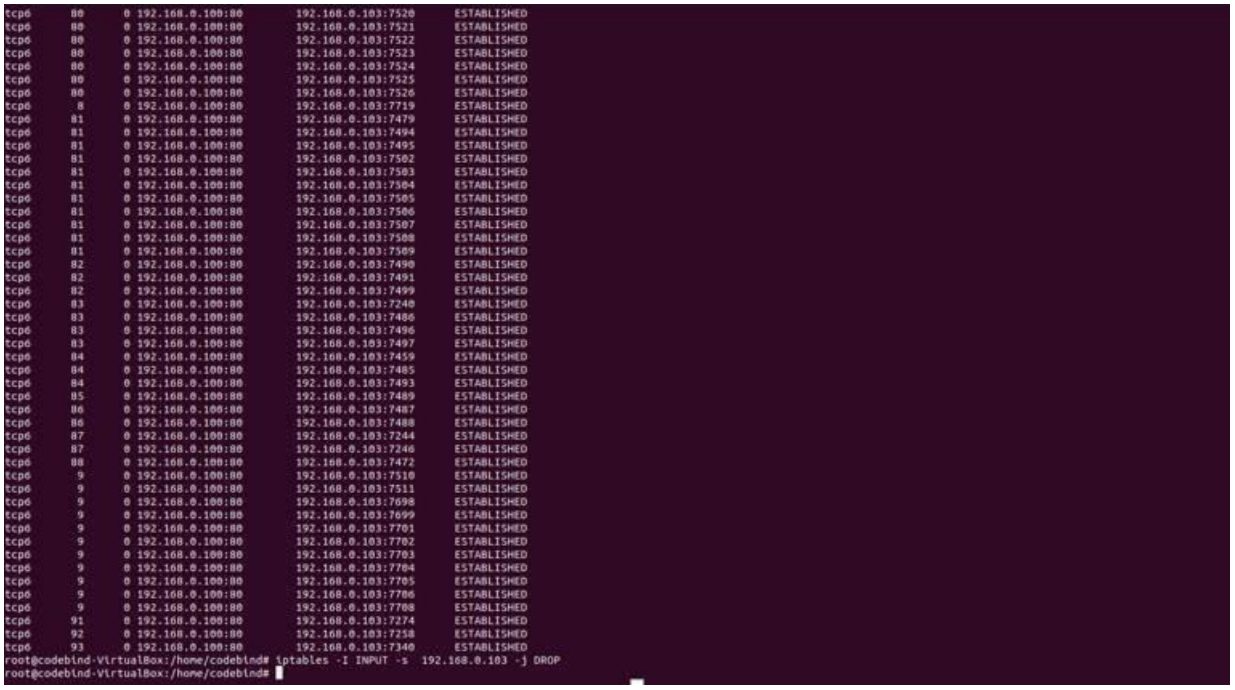


Figure 7: Packets start to drop

192.168.0.100

Figure 8: The site can’t acc

Part 5: Result Analysis

Dataset Collection:

Data set was collected from the Kaggle

Network assaults that have been recorded include UDP-Flood, Smurf, SIDDOS, HTTP-FLOOD, and regular traffic.

How Wake works?

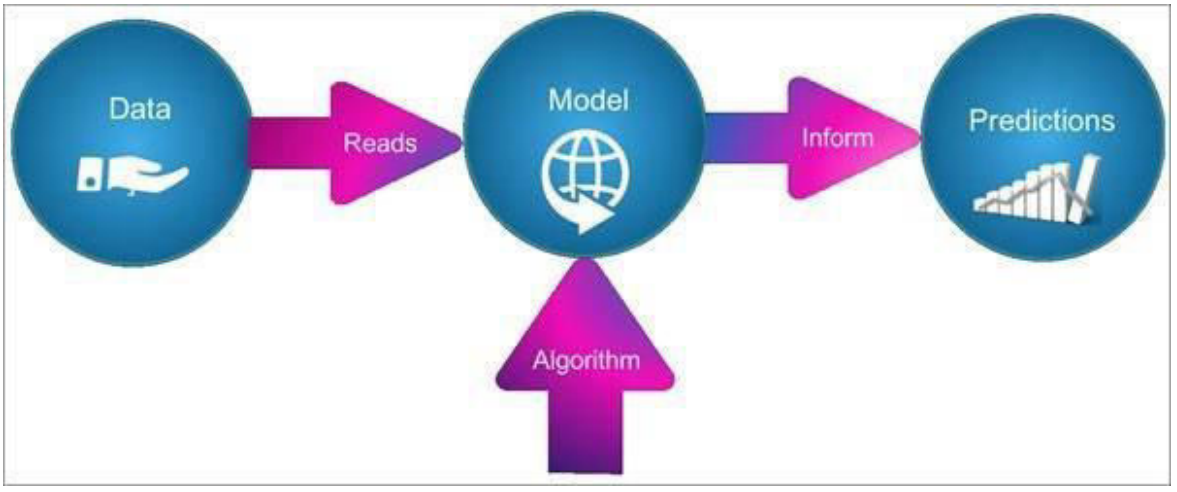


Figure 9: wake working principles

Dataset Description:
In main Dataset there are 1048576 Instance with 28 Instance now we make it short with 310244

Instance & 27 attributes as our training data set.

Attributes are:

1. SRC_ADDR_A
2. DDR
3. PKT_ID
4. FROM_NODE
5. TO_NODE
6. PKT_TYPE
7. PKT_SIZE
8. FLAGS
9. FID

10. SEQ_NUMBER
11. NUMBER_OF_BYTE
12. NODE_NAME_FROM
13. NODE_NAME_TO
14. PKT_IN
15. PKT_OUT
16. PKT_RATE
17. BYTE_RATE
18. PKT_AVG_SIZE
19. UTILIZATION
20. PKT_DELAY
21. PKT_SEND_TIME
22. PKT_RESEVED_TIME
23. FIRST_PKT_SENT
24. LAST_PKT_RESEVED
25. PKT_CLASS
26.
27. PKT_R
28. PKT_DELAY_NODE

After make training set we are applying J48 algorithm on it &

Analysis of the dataset:

Relation Name: final-dataset.training set

Name: PKT_ID

Instance:310244

Attributes: 27

Missing:10688 (3%)

Type: Numeric

Unique: 204308(66%)

Distinct: 2437

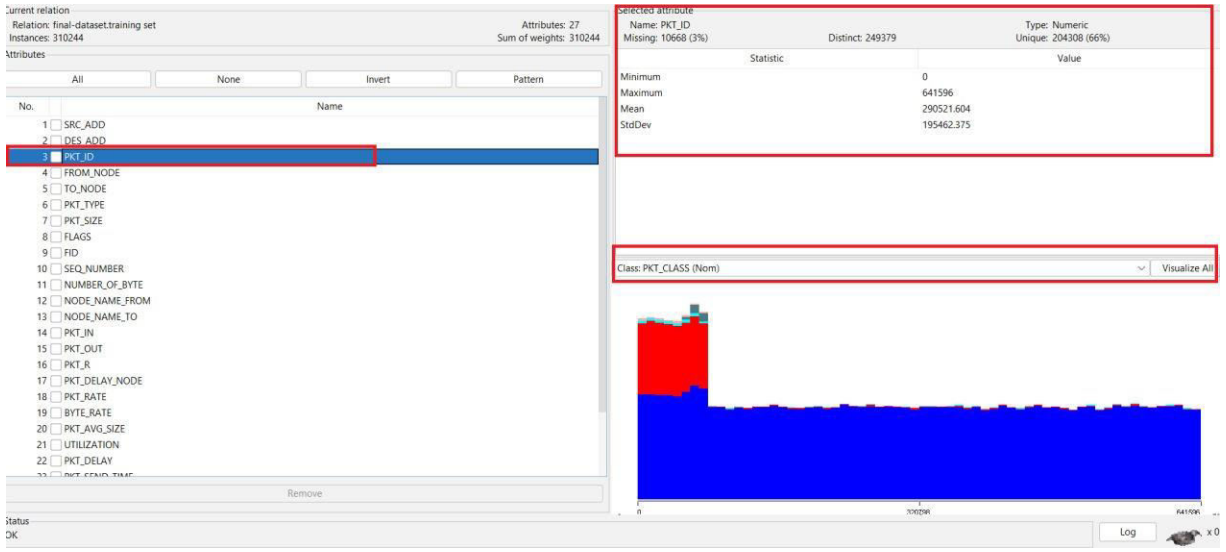


Figure 10: Dataset in details

Analysis of the dataset:
Relation Name: final-dataset.training set
NAME: PKT_CLASS
Instance:310244
Attributes: 27
Missing:10688 (3%)
Type: Nominal (PKT_CLASS type can't me numerical)
Unique: 0 (0%)
Distinct: 5

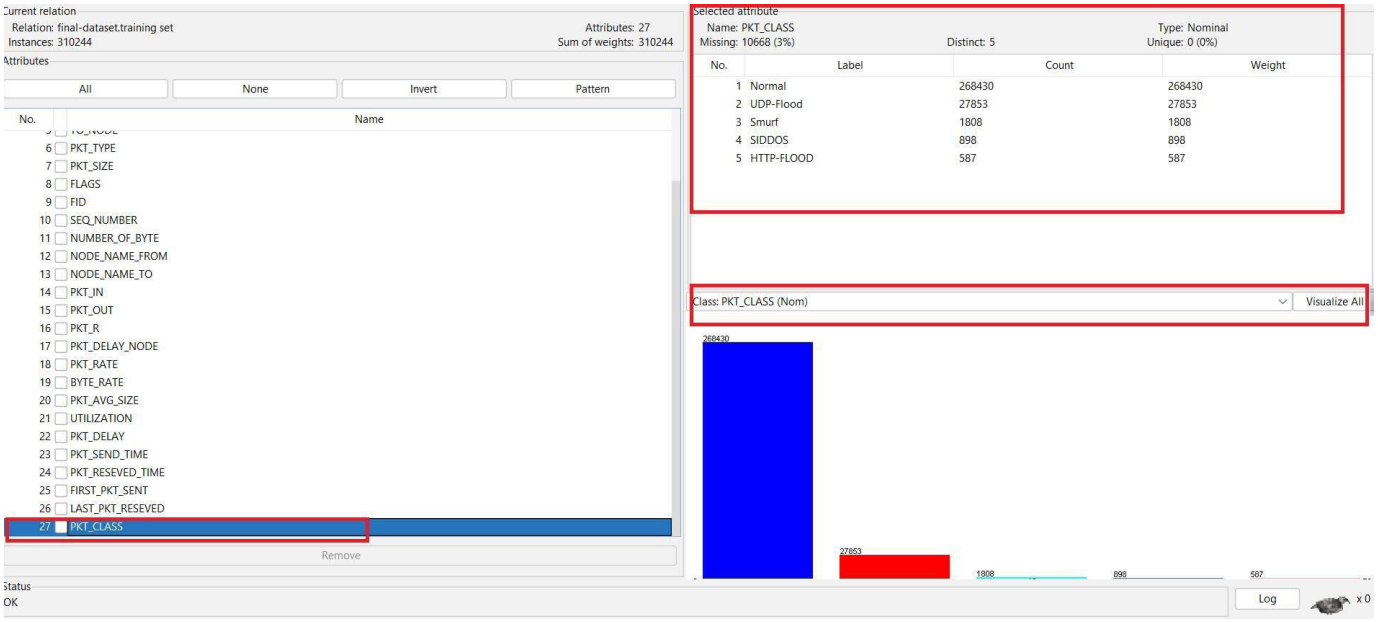


Figure 11: Analysis of dataset (PKT_CLASS)

Analysis of the dataset:
Relation Name: final-dataset.training set
NAME: PKT_CLASS
Instance:310244
Attributes: 27
Missing:10688 (3%)
Type: Numeric
Unique: 560 (0%)
Distinct: 576

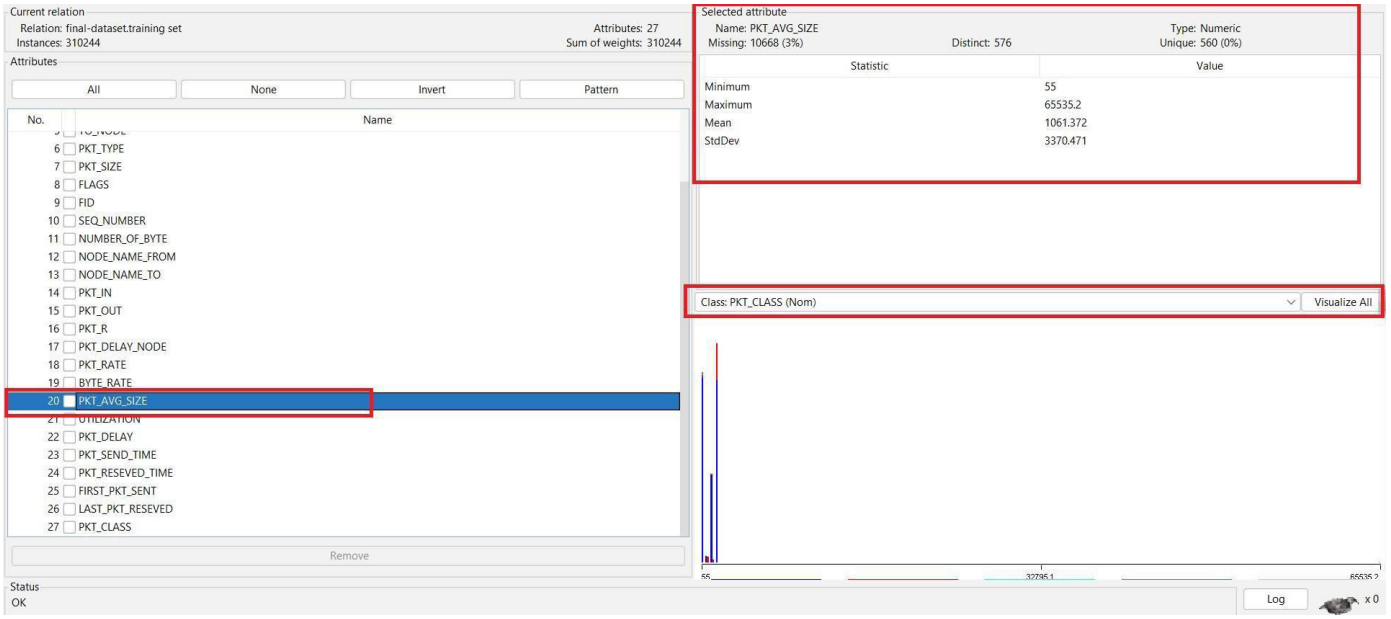


Figure 12

Histogram:

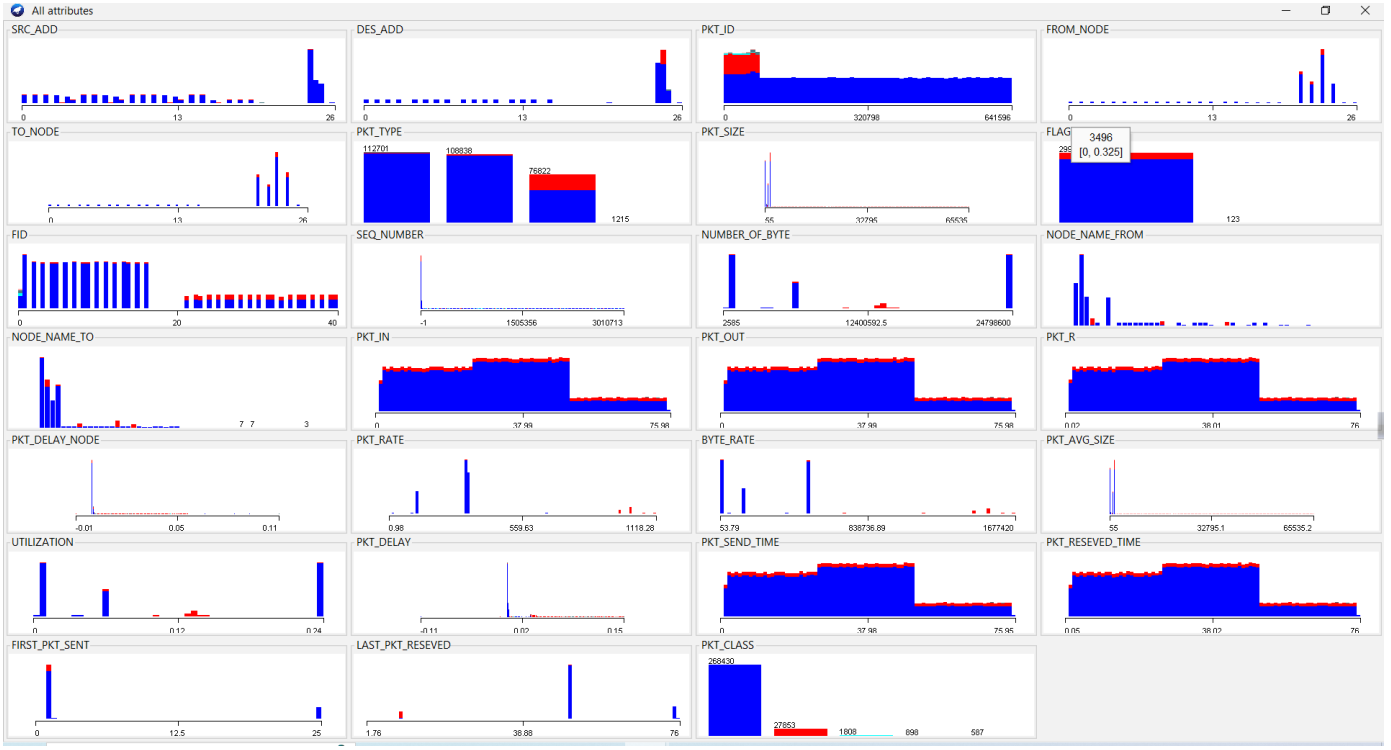


Figure 13: Attributes in details

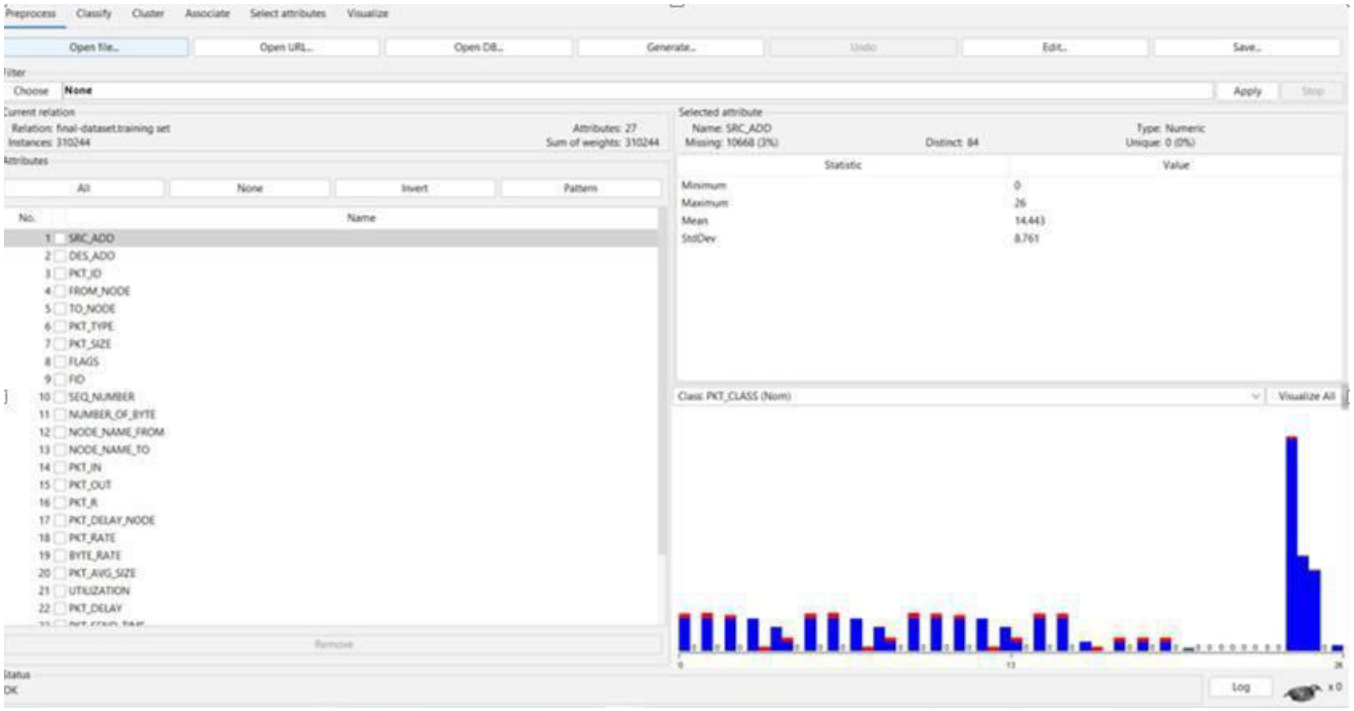


Figure 14: Selected Dataset

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     final-dataset.training set
Instances:    310244
Attributes:   27
              SRC_ADD
              DES_ADD
              PKT_ID
              FROM_NODE
              TO_NODE
              PKT_TYPE
              PKT_SIZE
              FLAGS
              FID
              SEQ_NUMBER
              NUMBER_OF_BYTE
              NODE_NAME_FROM
              NODE_NAME_TO
              PKT_IN
              PKT_OUT
              PKT_R
              PKT_DELAY_NODE
              PKT_RATE
              BYTE_RATE
              PKT_AVG_SIZE
              UTILIZATION
              PKT_DELAY
              PKT_SEND_TIME
              PKT_RESEVED_TIME
              FIRST_PKT_SENT
              LAST_PKT_RESEVED
              PKT_CLASS
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===
```

```
J48 pruned tree
------------------

PKT_RATE <= 658.090443
|   PKT_SIZE <= 1540
|   |   PKT_RATE <= 94.7212
|   |   |   BYTE_RATE <= 4354.82: Normal (1816.0/31.0)
|   |   |   BYTE_RATE > 4354.82: SIDDOS (986.0/135.0)
|   |   PKT_RATE > 94.7212: Normal (270650.0/4046.0)
|   PKT_SIZE > 1540
|   |   PKT_TYPE = tcp: HTTP-FLOOD (558.0/15.0)
|   |   PKT_TYPE = ack: Smurf (0.0)
|   |   PKT_TYPE = cbr: Smurf (0.0)
|   |   PKT_TYPE = ping: Smurf (580.0)
PKT_RATE > 658.090443: UDP-Flood (24986.0)

Number of Leaves  :     8

Size of the tree :      13


Time taken to build model: 28.95 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      295348               98.5887 %
Incorrectly Classified Instances      4228                1.4113 %
Kappa statistic                          0.9204
Mean absolute error                      0.0111
Root mean squared error                  0.0746
Relative absolute error                 14.7603 %
Root relative squared error             38.425  %
Total Number of Instances           299576
Ignored Class Unknown Instances             10668
```

Tree:

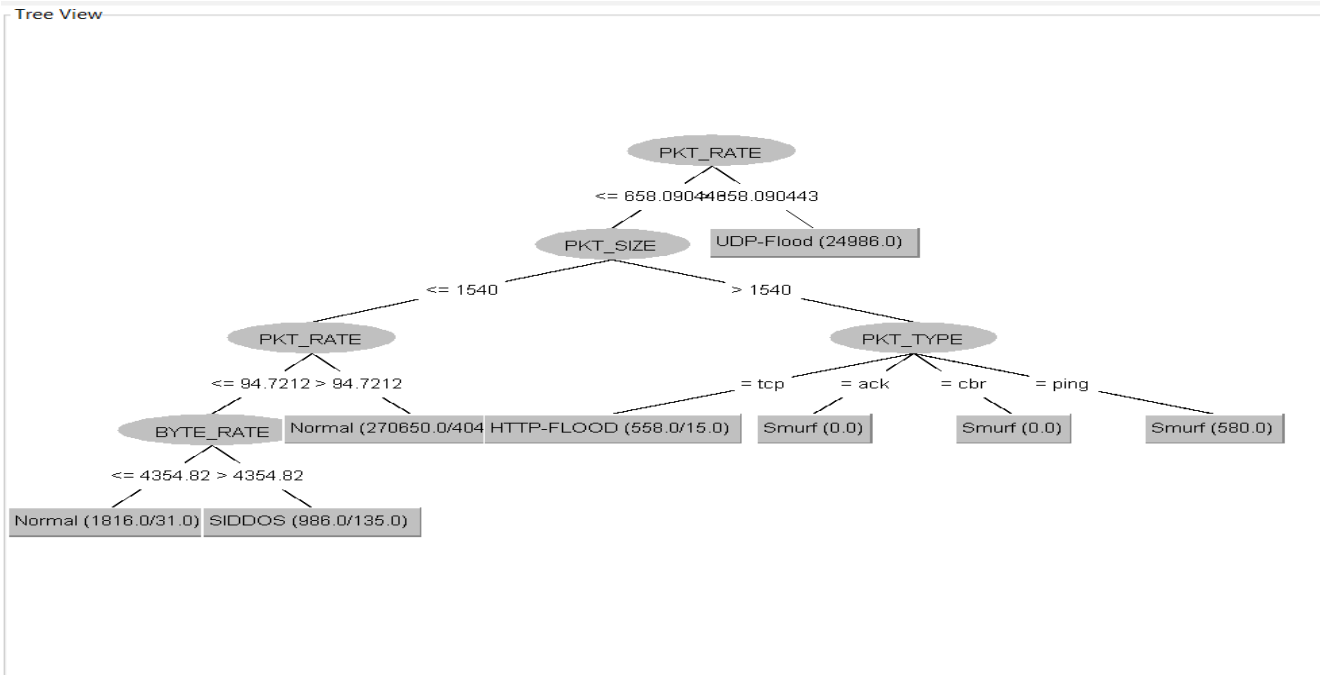
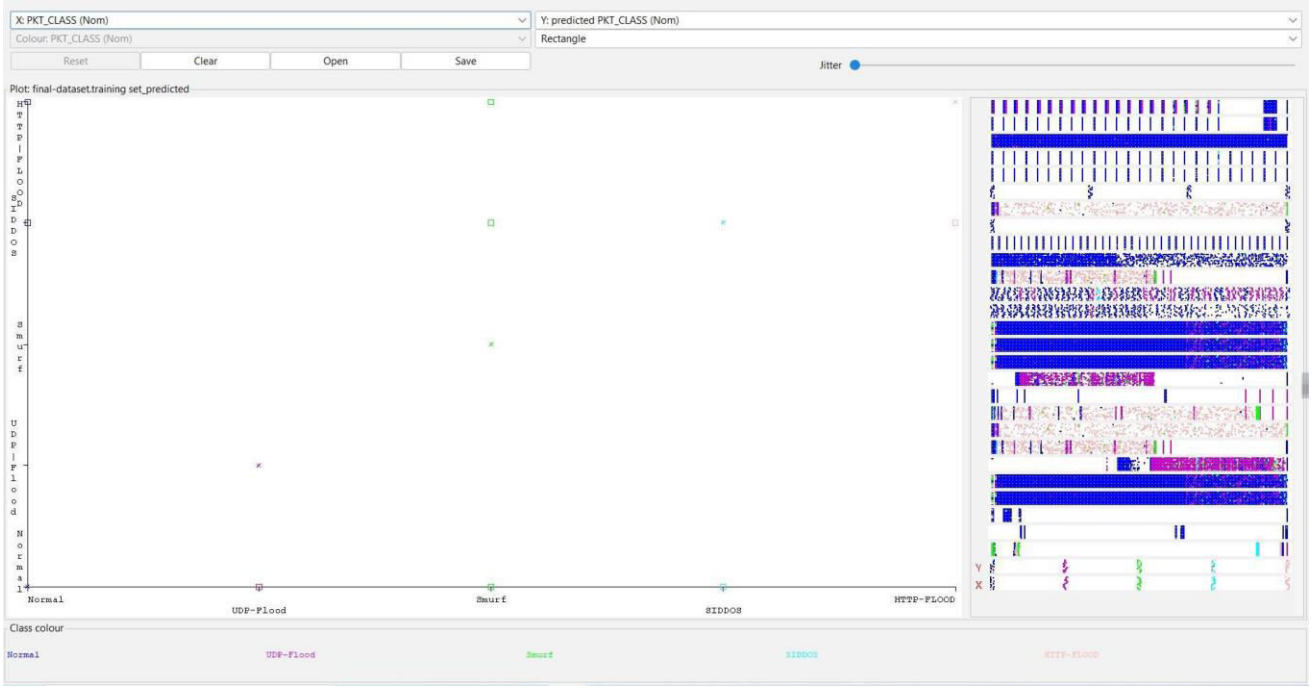


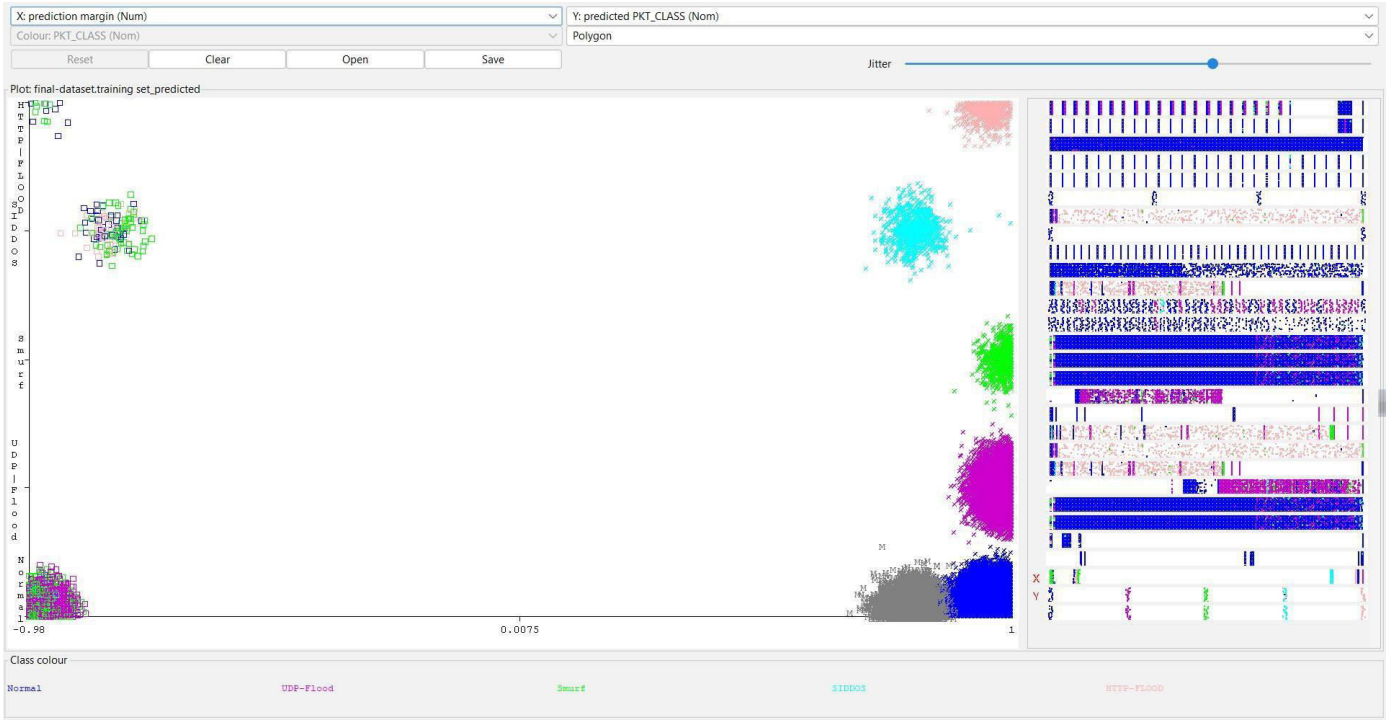
Fig: Decision Tree

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      295348               98.5887 %
Incorrectly Classified Instances      4228                1.4113 %
Kappa statistic                          0.9204
Mean absolute error                      0.0111
Root mean squared error                  0.0746
Relative absolute error                 14.7603 %
Root relative squared error             38.425  %
Total Number of Instances           299576
Ignored Class Unknown Instances             10668

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.131    0.985      1.000    0.992      0.924    0.950     0.985     Normal
                 0.897    0.000    1.000      0.897    0.946      0.942    0.945     0.915     UDP-Flood
                 0.321    0.000    1.000      0.321    0.486      0.565    0.690     0.347     Smurf
                 0.947    0.000    0.863      0.947    0.903      0.903    0.968     0.803     SIDDOS
                 0.925    0.000    0.973      0.925    0.948      0.949    1.000     0.925     HTTP-FLOOD
Weighted Avg.    0.986    0.117    0.986      0.986    0.985      0.924    0.948     0.974

=== Confusion Matrix ===

      a      b      c      d      e   <-- classified as
 268389      0      0     33      8 |      a = Normal
   2867  24986      0      0      0 |      b = UDP-Flood
   1163      0    580     58      7 |      c = Smurf
     48      0      0    850      0 |      d = SIDDOS
      0      0      0     44    543 |      e = HTTP-FLOOD
```

Classifier Error:



Prediction Margin (Num)



Cost/Benefit Analysis-Tree.j48 (Class=SIDDOS)
Classification Accuracy: 99.7106%
Confusion Matrix:

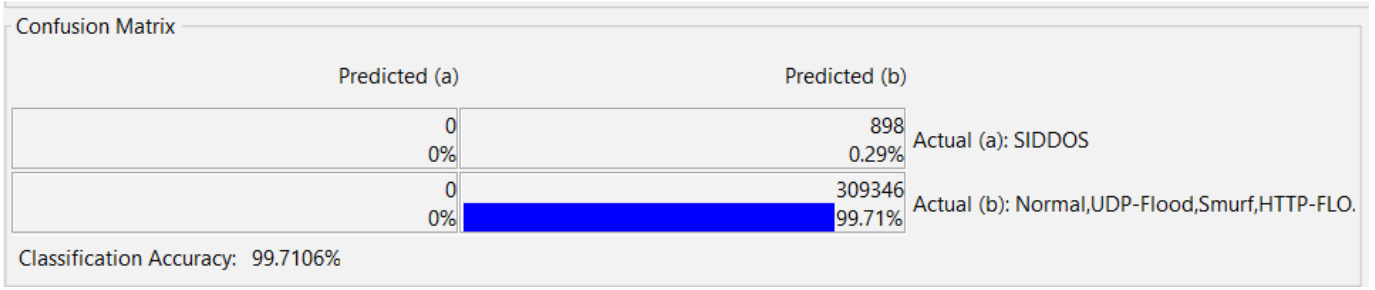


Fig: Confusion Matrix

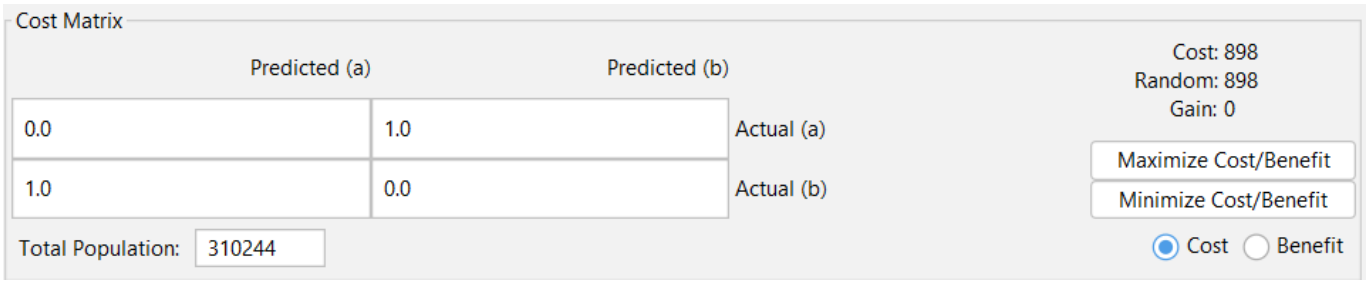


Figure: Cost Matrix

Applying Naïve Bayes Classifier:

```
=== Classifier model (full training set) ===

Naive Bayes Classifier

                          Class
Attribute                Normal      UDP-Flood         Smurf        SIDDOS     HTTP-FLOOD
                          (0.9)         (0.09)          (0.01)           (0)           (0)
=============================================================================================
SRC_ADD
  mean                  14.8968        10.1469        13.6364        19.7287       10.4489
  std. dev.              8.8757         6.3023         8.3371         2.2526        5.8451
  weight sum             268430          27853           1808            898           587
  precision              0.3133         0.3133         0.3133         0.3133        0.3133

DES_ADD
  mean                  17.6243        23.8334        19.7732        24.7049       24.3367
  std. dev.              8.7065         3.4716         7.8924         2.3582        0.3302
  weight sum             268430          27853           1808            898           587
  precision              0.3333         0.3333         0.3333         0.3333        0.3333

PKT_ID
  mean                315340.0919    68254.5892    215492.6626    82347.6022    37317.1301
  std. dev.           187386.7518   105835.6569   202651.2538     76281.95     22061.931
  weight sum             268430          27853           1808            898           587
  precision              2.5728         2.5728         2.5728         2.5728        2.5728

FROM_NODE
  mean                   19.737         18.1033        19.3573        21.1837       18.5605
  std. dev.              6.2057         6.9117         6.3385         1.9789        6.4484
  weight sum             268430          27853           1808            898           587
  precision                   1              1              1              1             1

TO_NODE
  mean                  20.6359        22.5977        21.3092        22.8742       22.8722
  std. dev.              5.4829         2.1301         4.8001         2.0102        1.0931
  weight sum             268430          27853           1808            898           587
  precision                   1              1              1              1             1
```

Summary:

```
Time taken to build model: 1.64 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      290400               96.937  %
Incorrectly Classified Instances      9176                3.063  %
Kappa statistic                          0.8393
Mean absolute error                      0.0123
Root mean squared error                  0.1102
Relative absolute error                 16.2822 %
Root relative squared error             56.7778 %
Total Number of Instances           299576
Ignored Class Unknown Instances             10668

=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 0.983     0.129     0.985      0.983     0.984      0.849     0.942     0.984     Normal
                 0.897     0.000     1.000      0.897     0.946      0.942     0.947     0.918     UDP-Flood
                 0.018     0.012     0.009      0.018     0.012      0.005     0.391     0.005     Smurf
                 0.943     0.003     0.459      0.943     0.618      0.657     0.969     0.816     SIDDOS
                 0.935     0.002     0.466      0.935     0.622      0.659     0.997     0.917     HTTP-FLOOD
Weighted Avg.    0.969     0.116     0.978      0.969     0.973      0.852     0.940     0.971
```

Matrix of Confusion for Naive Bayes:

```
=== Confusion Matrix ===

      a      b      c      d      e    <-- classified as
 263985      0   3515    893     37 |      a = Normal
   2844  24986     19      3      1 |      b = UDP-Flood
   1124      0     33     64    587 |      c = Smurf
     46      0      1    847      4 |      d = SIDDOS
      0      0      0     38    549 |      e = HTTP-FLOOD
```

Figure 15-17: Result after Naïve Bayes Algorithm apply data set

Applying KNN Algorithm:

```
=== Classifier model (full training set) ===

IB1 instance-based classifier
using 1 nearest neighbour(s) for classification


Time taken to build model: 0.08 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      291317               97.2431 %
Incorrectly Classified Instances      8259                2.7569 %
Kappa statistic                          0.8535
Mean absolute error                      0.011
Root mean squared error                  0.105
Relative absolute error                 14.6375 %
Root relative squared error             54.0935 %
Total Number of Instances           299576
Ignored Class Unknown Instances             10668
```

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall   F-Measure   MCC      ROC Area   PRC Area   Class
               0.985     0.130     0.985       0.985    0.985       0.856    0.927      0.978      Normal
               0.898     0.010     0.901       0.898    0.900       0.889    0.945      0.825      UDP-Flood
               0.321     0.004     0.319       0.321    0.320       0.316    0.658      0.106      Smurf
               0.805     0.001     0.815       0.805    0.810       0.810    0.903      0.658      SIDDOS
               0.903     0.000     0.906       0.903    0.904       0.904    0.937      0.050      HTTP-FLOOD
Weighted Avg.  0.972     0.118     0.972       0.972    0.972       0.856    0.927      0.956
```

Confusion Matrix:

```
=== Confusion Matrix ===

      a      b      c      d      e    <-- classified as
 264471   2733   1146     72      8 |      a = Normal
   2823  25012     17      1      0 |      b = UDP-Flood
   1148     14    581     57      8 |      c = Smurf
     72      1     63    723     39 |      d = SIDDOS
     11      0     12     34    530 |      e = HTTP-FLOOD
```

Figure 18-19: Result after KNN Algorithm apply data set

Algorithm	Accuracy	Correctly Classified Instances		Incorrectly Classified Instances	
J48	98.5887 %	295348	98.5887 %	4228	1.4113 %
Naïve Bayes	96.937 %	290400	96.937 %	9176	3.063 %
KNN	97.2431 %	291317	97.2431 %	8259	2.7569 %

Table 5A: Classifier accuracy comparison

Test Set:
We make a test set from our training set.In test set now have 47488 instance & 27 Attributes.

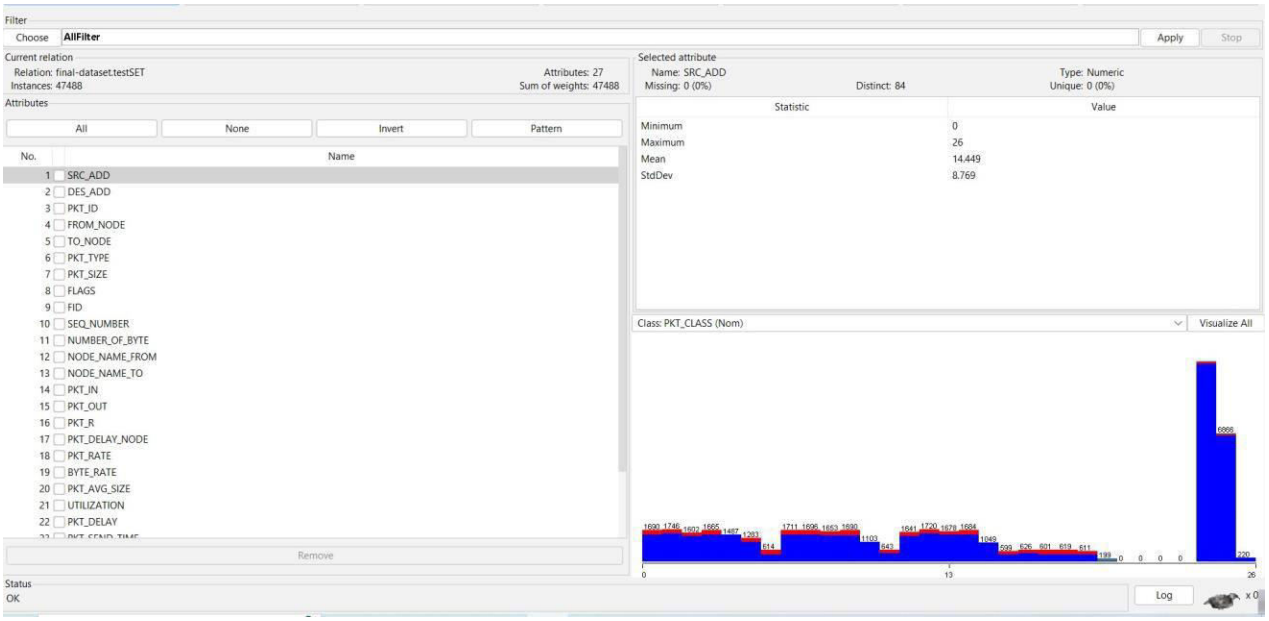


Figure 20: Selected test Dataset

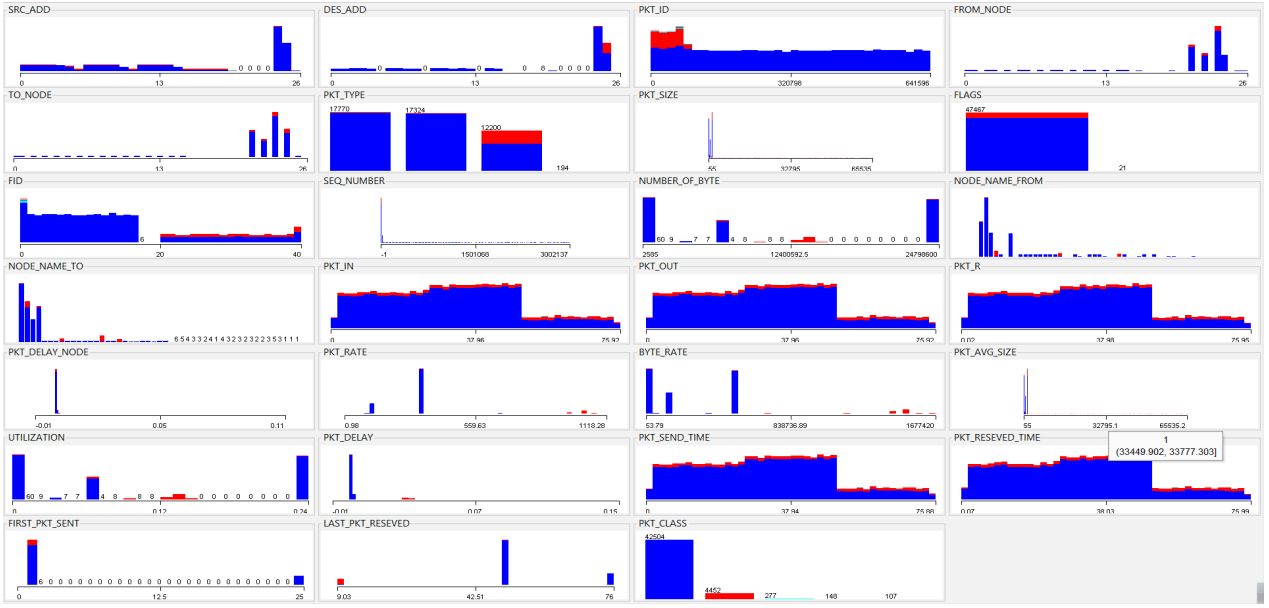


Figure 21: Details of All attributes

Applying J48 Algorithm:

```
=== Run information ===

Scheme:       weka.classifiers.misc.InputMappedClassifier -I -trim -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Relation:     final-dataset.testSET
Instances:    47488
Attributes:   27
              SRC_ADD
              DES_ADD
              PKT_ID
              FROM_NODE
              TO_NODE
              PKT_TYPE
              PKT_SIZE
              FLAGS
              FID
              SEQ_NUMBER
              NUMBER_OF_BYTE
              NODE_NAME_FROM
              NODE_NAME_TO
              PKT_IN
              PKT_OUT
              PKT_R
              PKT_DELAY_NODE
              PKT_RATE
              BYTE_RATE
              PKT_AVG_SIZE
              UTILIZATION
              PKT_DELAY
              PKT_SEND_TIME
              PKT_RESEVED_TIME
              FIRST_PKT_SENT
              LAST_PKT_RESEVED
              PKT_CLASS
Test mode:    user supplied test set:  size unknown (reading incrementally)
```

```
=== Classifier model (full training set) ===

J48 pruned tree
------------------

PKT_RATE <= 658.090443
|   PKT_SIZE <= 1540
|   |   PKT_RATE <= 94.7212
|   |   |   BYTE_RATE <= 4354.82: Normal (293.0/4.0)
|   |   |   BYTE_RATE > 4354.82
|   |   |   |   SRC_ADD <= 15.2: HTTP-FLOOD (4.0)
|   |   |   |   SRC_ADD > 15.2: SIDDOS (158.0/19.0)
|   |   PKT_RATE > 94.7212: Normal (42821.0/614.0)
|   PKT_SIZE > 1540
|   |   PKT_TYPE = tcp: HTTP-FLOOD (99.0/3.0)
|   |   PKT_TYPE = ack: HTTP-FLOOD (0.0)
|   |   PKT_TYPE = cbr: HTTP-FLOOD (0.0)
|   |   PKT_TYPE = ping: Smurf (94.0)
PKT_RATE > 658.090443: UDP-Flood (4019.0)

Number of Leaves  :     9

Size of the tree :      15


Time taken to build model: 2.33 seconds

=== Stratified cross-validation ===
```

Summary:

```
=== Summary ===

Correctly Classified Instances       46844               98.6439 %
Incorrectly Classified Instances       644                1.3561 %
Kappa statistic                          0.9244
Mean absolute error                      0.0107
Root mean squared error                  0.0732
Relative absolute error                 14.0355 %
Root relative squared error             37.5232 %
Total Number of Instances            47488

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               1.000    0.124    0.986      1.000    0.993      0.928    0.935     0.985     Normal
               0.903    0.000    1.000      0.903    0.949      0.945    0.950     0.921     UDP-Flood
               0.339    0.000    0.979      0.339    0.504      0.575    0.688     0.355     Smurf
               0.939    0.000    0.869      0.939    0.903      0.903    0.955     0.780     SIDDOS
               0.916    0.000    0.970      0.916    0.942      0.943    1.000     0.900     HTTP-FLOOD
Weighted Avg.  0.986    0.111    0.987      0.986    0.985      0.928    0.935     0.974
```

Confusion Matrix:

```
=== Confusion Matrix ===

     a     b     c     d     e   <-- classified as
 42494     0     2     6     2 |     a = Normal
   433  4019     0     0     0 |     b = UDP-Flood
   176     0    94     6     1 |     c = Smurf
     9     0     0   139     0 |     d = SIDDOS
     0     0     0     9    98 |     e = HTTP-FLOOD
```

Figure 22-25: Applying J48 algorithm on test dataset

Result:
Correctly classified: 98.6439%
Incorrectly Classified: 1.3561%

Applying Naïve Bayes:

Summary:

```
=== Summary ===

Correctly Classified Instances       46105               97.0877 %
Incorrectly Classified Instances      1383                2.9123 %
Kappa statistic                          0.8481
Mean absolute error                      0.0118
Root mean squared error                  0.1078
Relative absolute error                 15.4584 %
Root relative squared error             55.2676 %
Total Number of Instances            47488

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.984    0.123    0.986      0.984    0.985      0.858    0.930     0.984     Normal
                 0.903    0.000    1.000      0.903    0.949      0.945    0.951     0.925     UDP-Flood
                 0.014    0.013    0.007      0.014    0.009      0.001    0.405     0.005     Smurf
                 0.932    0.002    0.639      0.932    0.758      0.771    0.963     0.823     SIDDOS
                 0.935    0.002    0.500      0.935    0.651      0.683    1.000     0.910     HTTP-FLOOD
Weighted Avg.    0.971    0.110    0.979      0.971    0.974      0.861    0.929     0.972
```

Confusion Matrix:

```
=== Confusion Matrix ===

     a     b     c     d     e   <-- classified as
 41844     1   591    64     4 |     a = Normal
   430  4019     2     1     0 |     b = UDP-Flood
   172     0     4     6    95 |     c = Smurf
     9     0     0   138     1 |     d = SIDDOS
     0     0     0     7   100 |     e = HTTP-FLOOD
```

Figure 26-27: Applying Naïve Bayes algorithm on test dataset

Result:
Correctly classified: 97.0877%
Incorrectly Classified: 2.9123%

Applying KNN algorithm:

```
=== Summary ===

Correctly Classified Instances       46232               97.3551 %
Incorrectly Classified Instances      1256                2.6449 %
Kappa statistic                          0.8606
Mean absolute error                      0.0106
Root mean squared error                  0.1029
Relative absolute error                 13.9589 %
Root relative squared error             52.754  %
Total Number of Instances            47488

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.986    0.124    0.985      0.986    0.986      0.863    0.930     0.984     Normal
                 0.904    0.009    0.909      0.904    0.906      0.897    0.947     0.832     UDP-Flood
                 0.347    0.004    0.339      0.347    0.343      0.339    0.673     0.124     Smurf
                 0.784    0.001    0.789      0.784    0.786      0.786    0.896     0.640     SIDDOS
                 0.897    0.000    0.873      0.897    0.885      0.885    0.954     0.791     HTTP-FLOOD
Weighted Avg.    0.974    0.112    0.974      0.974    0.974      0.863    0.930     0.963
```

Conf
usion Matrix:

```
=== Confusion Matrix ===

     a     b     c     d     e   <-- classified as
 41900   404   178    18     4 |     a = Normal
   428  4024     0     0     0 |     b = UDP-Flood
   174     0    96     6     1 |     c = Smurf
    15     0     8   116     9 |     d = SIDDOS
     3     0     1     7    96 |     e = HTTP-FLOOD
```

Figure 28-29: Applying J48 algorithm on test dataset

Result:
Correctly classified: 97.3551%
Incorrectly Classified:
2.6449%Table: classifier
accuracy

Classifier	Accuracy
J48	98.6439%
Naïve Bayes	97.0877%
KNN	97.3551

Table 5B: classifier accuracy

Part 6: Discussion

Our research demonstrates that employing J48 machine learning technology allows for a more precise identification and prevention of DDoS attacks. We have established a strong correlation between an intrusion detection system and a machine learning algorithm, with the intrusion detection system proving to be the most reliable method for mitigating DDoS attacks. In our efforts to safeguard against DDoS attacks, we primarily rely on the J48 machine learning approach.
While various machine learning techniques like KNN, Random Forest, SVM, Decision Tree, and MLP have been explored for DDoS attack prevention, our study produces specific results. However, in line with similar research findings, our use of the J48 machine learning approach for detecting and thwarting DDoS attacks offers a more rational explanation.
As we employ machine learning techniques like J48 to identify and counter DDoS attacks, our findings align with those of previous studies. Moreover, our research contributes a higher level of accuracy compared to earlier publications. Within this study, we also provide insights into how the J48 machine learning algorithm can be utilized to detect and block DDoS attacks, shedding new light on the synergy between intrusion detection systems and machine learning algorithms, resulting in improved detection and prevention rates.

Part 7: Conclusion and Future Work

Based on the outcomes of our research, we have devised a strategy for safeguarding against DDoS attacks, which are among the most destructive cyber threats in today's technology-driven era. Our investigation delved into various machine learning techniques for DDoS detection, ultimately leading us to conclude that the J48 machine learning approach is better suited for preventing DDoS attacks. By employing our proposed methodology, we have the potential to enhance the accuracy of DDoS attack detection.

In our illustrative example, we demonstrated how J48 decision tree algorithms can swiftly identify and subsequently halt DDoS attacks. Our model relies on two datasets: one for training and another for testing. The detection process commences with the training dataset, which identifies datasets that are or could potentially be under attack. Following this, the J48 classifier filters the dataset before moving to the testing phase, where it is trained using pattern data. Our system segregates the dataset into two categories: those affected by DDoS attacks and those unaffected. Utilizing the J48 decision tree approach, we can promptly identify DDoS attacks, and the process is relatively straightforward. We believe that our technology exhibits the highest level of accuracy in detecting and responding to DDoS attacks.

We are committed to further developing our proposed model, refining the process of segregating DDoS attack data from regular data, and continuing our work on DDoS attack detection using J48. In the future, we aim to address the challenges encountered during our research for this paper and complete the entire system independently.

References

[1] P. S. Saini, S. Behal, and S. Bhatia, “Detection of DDoS Attacks using Machine Learning Algorithms,” 2020 7th International Conference on Computing for Sustainable Global Development (INDIACom), Mar. 2020, doi: 10.23919/indiacom49435.2020.9083716.

[2] S. Wani, M. Imthiyas, H. Almohamedh, K. M. Alhamed, S. Almotairi, and Y. Gulzar, “Distributed Denial of Service (DDoS) Mitigation Using Blockchain—A Comprehensive Insight,” Symmetry, vol. 13, no. 2, p. 227, Jan. 2021, doi: 10.3390/sym13020227.

[3] E. Söğüt, S. Oyucu, and O. A. Erdem, “Detecting Different Types of Distributed Denial of Service Attacks,” Gazi University Journal of Science Part C: Design and Technology, vol. 9, no. 1, pp. 12–25, Mar. 2021, doi: 10.29109/gujsc.840126/(accessed Sept. 27,2022).

[4] Devi S. R., Yogesh P. (2012). Detection of Application Layer DDoS Attacks Using Information Theory Based Metrics. Computer Science & Information Technology, Vol. 10, 217–223.

[5] Baykara, M., Daş, R. (2017). A Novel Hybrid Approach for Detection of Web-Based Attacks in Intrusion Detection Systems. International Journal of Computer Networks andApplications, 4(2), 62- 76.

[6] L. Stein and J. Stewart, “WWW Security FAQ: Securing Against Denial of Service Attacks,” www.w3.org, Feb. 23, 2003. http://www.w3.org/Security/Faq/wwwsf6.html (accessed Oct. 30, 2022).

[7] GezgınD. M. and E. Buluş, “Kablosuz Ağlar için bir DoS Saldırısı Tasarımı,” Bilişim Teknolojileri Dergisi, vol. 6, no. 3, pp. 17–23, Jan. 2014, Accessed: Nov. 01, 2022. [Online]. Available: https://dergipark.org.tr/en/pub/gazibtd/issue/6629/88018#article_cite

[8] A. Raza, “Anomaly Detection Systems for Distributed Denial of Service Attacks,” dspace.library.uvic.ca, 2016, Accessed: Nov. 01, 2022. [Online]. Available: https://dspace.library.uvic.ca/handle/1828/7817

[9] Wueest C. The continued rise of DDoS attacks. White Paper: Security Response, Symantec Corporation. 2014.

[10] Sonar, Krushang, Upadhyay, and Hardik, “A survey: DDOS attack on Internet of Things,” International Journal of Engineering Research and Development, vol. 10, no. 11, pp. 58–63, 2014.

[11] Çelikbilek, İ. (2016). TCP SYN Seli Saldırısının Etkilerini Azaltmak için Yeni SYN Çerezleri Gerçeklemesi. İstanbul Şehir Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul.

[12] McGregory and Steve, “Preparing for the next DDoS attack,” Network Security, vol. 2013, no. 5, pp. 5–6, 2013.

[13] Ingle, A. and Awade, and M, “Intrusion Detection for TCP--SYNC Flood Attack,” International Journal of Advanced Research in Computer Science, vol. 4, no. 5, pp. 9–11, 2013.

[14] A. Toh, "Azure DDoS Protection—2021 Q1 and Q2 DDoS attack trends," Azure, Aug. 12, 2021. https://azure.microsoft.com/en-us/blog/azure-ddos-protection-2021-q1-and-q2-ddos-attack-trends/(accessed Sept. 26, 2022).

[15] Stapel and . Klepfish, "Record 25.3 Billion Request Multiplexing DDoS Attack Mitigated by Imperva," Imperva, Sep. 19, 2022. https://www.imperva.com/blog/record 25-3-billion-request-multiplexing-attack-mitigated-by-imperva/(accessed Sept. 27,2022).

[16] J. Smith-perrone and J. Sims, "Securing cloud, SDN and large data network environments from emerging DDoS attacks," 2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence, 2017, pp. 466-469, doi:10.1109/CONFLUENCE.2017.7943196.

[17] B. Zhang, T. Zhang and Z. Yu, "DDoS detection and prevention based on artificial intelligence techniques," 2017 3rd IEEE International Conference on Computer and Communications(ICCC), 2017, pp. 1276-1280.

[18] N. I. G. Dharma, M. F. Muthohar, J. D. A. Prayuda, K. Priagung and D. Choi, "Timebased DDoS detection and mitigation for SDN controller," 2015 17th Asia Pacific Network Operations and Management Symposium (APNOMS), 2015, pp. 550-553, doi: 10.1109/APNOMS.2015.7275389.

[19] Ismanto, H. and Wardoyo, and Retantyo, “RETRACTED: Comparison of running time between C4. 5 and k-nearest neighbor (k-NN) algorithm on deciding mainstay area clustering,” International Journal of Advances in Intelligent Informatics, vol. 2, no. 1, pp. 1–1, 2016.

[20] A. Nagaraja, U. Boregowda, and R. Vangipuram, “Study of Detection of DDoS attacks in cloud environment Using Regression Analysis,” International Conference on Data Science, E- learning and Information Systems 2021, p. 7, Apr. 2021, doi: 10.1145/3460620.3460750.

[21] Stephen M. Specht and Ruby B. Lee. 2004. Distributed Denial of Service: Taxonomies of Attacks, Tools, and Countermeasures. In Proceedings of the ISCA 17th International Conference on Parallel and Distributed Computing Systems, September 15-17, 2004, The Canterbury Hotel, San Francisco, California, USA, David A. Bader and Ashfaq A. Khokhar (Eds.). ISCA, 543–550.

[22] S. Kumar Pydipalli, S. Kasthuri and J. S, "DDOS DETECTION SYSTEM USING C4.5 DECISION TREE ALGORITHM," International Research Journal of Engineering and Technology (IRJET) , vol. 05, no. 12, Dec. 2018.

[23] C. Hu, H. Li, Y. Jiang, Y. Cheng, and P. Heegaard. 2016. Deep semantics inspection over big network data at wire speed. IEEE Network 30, 1 (2016), 18–23. https: //doi.org/10.1109/MNET.2016.7389826

[24] A. Nagaraja, S. Aljawarneh and P. H. S. , "PAREEKSHA: A Machine Learning Approach for Intrusion and Anomaly Detection. In Proceed ings of the First International Conference on Data Science," E-Learning and Information Systems (Madrid, Spain) (DATA '18). Association for Computing Machinery, New York, NY, USA, vol. Article 36, pp. 6, Oct. 2018, doi: 10.1145/3279996.3280032.

[25] Dusan Stevanovic, Natalija Vlajic, and Aijun An. 2013. Detection of malicious and non-malicious website visitors using unsupervised neural network learning. Applied Soft Computing 13,1 (2013), 698–708. https://doi.org/10.1016/j.asoc. 2012.08.028

[26] P. Chourasiya, "Implementation of Hybrid Approach for Intrusion Detection in Cloud Computing Environment," International Journal of Scientific Research in Science and Technology, pp. 233–237, Dec. 2018, doi: 10. 32628/CSEIT183878.

[27] N. Bharot, P. Verma, S. Sharma, and V. Suraparaju, "Distributed Denialof-Service Attack Detection and Mitigation Using Feature Selection and Intensive Care Request Processing Unit," Arabian Journal for Science and Engineering, vol. 43, no. 2, pp. 959-967, 2018.

[28] A. Banitalebi Dehkordi, M. Soltanaghaei, and F. Z. Boroujeni, "The DDoS attacks detection through machine learning and statistical methods in SDN," The JTheynal of Supercomputing, vol. 77, no. 3, pp. 2383-2415, 2021.

[29] O. Rahman, M. A. G. Quraishi, and C.-H. Lung, "DDoS attacks detection and mitigation in SDN using machine learning," in 2019 IEEE world congress on services (SERVICES), 2019, vol. 2642: IEEE, pp. 184-189.

[30] S. Lakshminarasimman, S. Ruswin, and K. Sundarakantham, "Detecting DDoS attacks using decision tree algorithm," in 2017 FTheyth International Conference on Signal Processing,

Communication and Networking (ICSCN), 2017: IEEE, pp. 1-6.

[31] K. Narasimha Mallikarjunan, A. Bhuvaneshwaran, K. Sundarakantham, and S. Mercy Shalinie, "DDAM: detecting DDoS attacks using machine learning approach," in Computational Intelligence: Theories, Applications and Future Directions-Volume I: Springer, 2019, pp. 261-273.

[32] S. Kumar Pydipalli, S. Kasthuri and J. S, "DDOS DETECTION SYSTEM USING C4.5 DECISION TREE ALGORITHM," International Research Journal of Engineering and Technology (IRJET) , vol. 05, no. 12, Dec. 2018.

[33] C. Hu, H. Li, Y. Jiang, Y. Cheng, and P. Heegaard. 2016. Deep semantics inspection over big network data at wire speed. IEEE Network 30, 1 (2016), 18–23. https: //doi.org/10.1109/MNET.2016.7389826

[34] A. Nagaraja, S. Aljawarneh and P. H. S. , "PAREEKSHA: A Machine Learning Approach for Intrusion and Anomaly Detection. In Proceed ings of the First International Conference on Data Science," E-Learning and Information Systems (Madrid, Spain) (DATA '18). Association for Computing Machinery, New York, NY, USA, vol. Article 36, pp. 6, Oct. 2018, doi: 10.1145/3279996.3280032.

[35] Dusan Stevanovic, Natalija Vlajic, and Aijun An. 2013. Detection of malicious and non-malicious website visitors using unsupervised neural network learning. Applied Soft Computing 13,1 (2013), 698–708. https://doi.org/10.1016/j.asoc. 2012.08.028

[36] P. Chourasiya, "Implementation of Hybrid Approach for Intrusion Detection in Cloud Computing Environment," International Journal of Scientific Research in Science and Technology, pp. 233–237, Dec. 2018, doi: 10. 32628/CSEIT183878.

[37] N. Bharot, P. Verma, S. Sharma, and V. Suraparaju, "Distributed Denialof-Service Attack Detection and Mitigation Using Feature Selection and Intensive Care Request Processing Unit," Arabian Journal for Science and Engineering, vol. 43, no. 2, pp. 959-967, 2018.

[38] A. Banitalebi Dehkordi, M. Soltanaghaei, and F. Z. Boroujeni, "The DDoS attacks detection through machine learning and statistical methods in SDN," The JTheynal of Supercomputing, vol. 77, no. 3, pp. 2383-2415, 2021.

[39] O. Rahman, M. A. G. Quraishi, and C.-H. Lung, "DDoS attacks detection and mitigation in SDN using machine learning," in 2019 IEEE world congress on services (SERVICES), 2019, vol. 2642: IEEE, pp. 184-189.

[40] S. Lakshminarasimman, S. Ruswin, and K. Sundarakantham, "Detecting DDoS attacks using decision tree algorithm," in 2017 FTheyth International Conference on Signal Processing, Communication and Networking (ICSCN), 2017: IEEE, pp. 1-6.

[41] K. Narasimha Mallikarjunan, A. Bhuvaneshwaran, K. Sundarakantham, and S. Mercy Shalinie, "DDAM: detecting DDoS attacks using machine learning approach," in Computational Intelligence: Theories, Applications and Future Directions-Volume I: Springer, 2019, pp. 261-273.

[42] A. M. Vieira, R. d. S. M. Junior, and A. d. R. L. Ribeiro, "Systematic Mapping on Prevention of DDoS Attacks on Software Defined Networks," in 2021 IEEE International Systems Conference (SysCon), 2021: IEEE, pp. 18.

[43] M. J. Mirchev and S. T. Mirtchev, "System for DDoS attack mitigation by discovering the attack vectors through statistical traffic analysis," International Journal of Information and Computer Security, vol. 13, no. 3/4, p. 309, May 2020, doi: 10.1504/ijics.2020.109479.

[44] Ashraf, Javed, and S. Latif, "Handling intrusion and DDoS attacks in Software Defined Networks using machine learning techniques," presented at the National Software Engineering Conference. IEEE, 2014.

[45] M. Suresh and R. Anitha, "Evaluating Machine Learning Algorithms for Detecting DDoS Attacks," Advances in Network Security and Applications, pp. 441–452, 2011, doi: 10.1007/978-3- 642-22540-6_42.

[46] S. Das, A. M. Mahfouz, D. Venugopal, and S. Shiva, "DDoS Intrusion Detection Through Machine Learning Ensemble," IEEE Xplore, Jul. 01, 2019.

[47] https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8859416 (accessed Oct. 20, 2022).

[48] Rahman, O., Quraishi, M.A.G. and Lung, C.H., 2019, July. DDoS attacks detection and mitigation in SDN using machine learning. In 2019 IEEE world congress on services (SERVICES) (Vol. 2642, pp. 184-189). IEEE.

[49] Aryal, B. and Abbas, R. and Collings, and Iain B, "SDN Enabled DDoS Attack Detection and Mitigation for," Journal of Communications, vol. 16, no. 7, 2021.

[50] W. B. W Mariam and Y. Negash, "Performance Evaluation of Machine Learning Algorithms for Detection of SYN Flood Attack," 2021 IEEE AFRICON, vol. 1–6, Sep. 2021, doi: 10.1109/africon51333.2021.9570968.

[51] Wani, A.R., Rana, Q.P., Saxena, U. and Pandey, N., 2019, February. Analysis and detection of DDoS attacks on cloud computing environment using machine learning techniques. In 2019 Amity International conference on artificial intelligence (AICAI) (pp. 870-875). IEEE.