DETECTION AND CLASSIFICATION OF FLOODING ATTACKS IN WIRELESS ADHOC NETWORKS USING MACHINE LEARNING

THESIS REPORT

Submitted in partial fulfillment of the requirements for the award of the Degree of Master of Technology in Department of Electronics & Communication Engineering with specialization in Communication Systems by the A P J Abdul Kalam Technological University

by

SHANIFA.E

TKM20ECCS11

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

TKM COLLEGE OF ENGINEERING

KOLLAM 691 005

July 2022

DETECTION AND CLASSIFICATION OF FLOODING ATTACKS IN WIRELESS ADHOC NETWORKS USING MACHINE LEARNING

THESIS REPORT

Submitted in partial fulfillment of the requirements for the award of the Degree of Master of Technology in Department of Electronics & Communication Engineering with specialization in Communication Systems by the A P J Abdul Kalam Technological University

by

SHANIFA.E

TKM20ECCS11

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

TKM COLLEGE OF ENGINEERING

KOLLAM 691 005

July 2022

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

TKM COLLEGE OF ENGINEERING

KOLLAM 691 005

CERTIFICATE

Certified that thesis report titled **" DETECTION AND CLASSIFICATION OF FLOODING ATTACKS IN WIRELESS ADHOC NETWORKS USING MACHINE LEARNING "** is a bonafide record of the work done by **SHANIFA E.** (Reg. No. TKM20ECCS11) under my supervision, in partial fulfillment of the requirements for the award of the Degree of Master of Technology in Electronics & Communication Engineering with specialization in Communication Systems by the A P J Abdul Kalam Technological University.

Guide & Coordinator:

Dr. Nishanth N.
Associate Professor ,
Dept. of ECE, TKMCE .

HoD:

Prof. Abid Hussain M.
Head of Department ,
Dept. of ECE, TKMCE .

Acknowledgement

At the outset, I find it my obligation to thank the Almighty God for giving me necessary wisdom to complete this project successfully.

I thank Prof. ABID HUSSAIN MUHAMMED, HoD, Department of Electronics and Communication Engineering for his encouragement and support.

I heartly express my profound gratitude to our project coordinator and my Guide, Dr. NISHANTH N. ,Associate Professor, Department of Electronics and Communication Engineering, for his advice, supervision , patience and support during the course of preparation and presentation of project.

I also express my heartfelt thanks to all my teachers, friends and my parents for providing the much needed support during the course of preparation and presentation of my project.

SHANIFA E.

TKM20ECCS11

ABSTRACT

In developing wireless adhoc networks, several sorts of attacks represent serious security problems.Various forms of attacks are currently being carried out against various services and resources, with the goal of compromising their availability, confidentiality, and integrity. Flooding based Denial of Service attacks has a serious impact on Wireless Local Area Networks. It may result in clients being denied service. This is a severe problem since it compromises one of the services offered by cyber security such as availability. In ad-hoc networks, it is crucial to detect and block such attacks in a timely manner. The objective of this thesis is to accurately detect and categorise flooding attacks such as TCP, UDP, or ICMP. This thesis additionally categorises additional assaults, such as U2R, R2L, and probe .The suggested method utilises various supervised machine learning algorithms, including SVM, KNN, Naive Baye's, Decision Tree, and Random Forest. Classification is performed using the NSL KDD data set. The outcome demonstrated that RF classifier provides the highest performance accuracy.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Wireless Ad-Hoc Networks

Ad-hoc networks are designed to function without the assistance of any fixed infrastructure.In ad hoc networks, the nodes can communicate with one another both inside and outside of their immediate radio range. In the second case, the nodes should set up an intermediary node that will serve as the router and direct the packet from its source to its destination. These networks lack infrastructure, and the connection is only temporary. The network's organisation and management are the responsibility of the terminals themselves. The individual terminals are free to move around and the entire network is mobile. Some terminal pairs in these networks might not be able to communicate with one another directly, forcing them to rely on other terminals to deliver messages to their intended recipients. Multi-hop or store-and-forward networks are common names for these networks. These networks' nodes serve as routers, establishing and maintaining connections to other nodes. No matter where the users are, these networks give mobile users access to information and ubiquitous computing capabilities. Diagram of wireless ad hoc networks is shown in Fig.1.1.

1.2 Advantages of Wireless Ad-Hoc Networks

- Ad hoc networks are easy to install and configure.
- Setting up ad-hoc networks is simple.

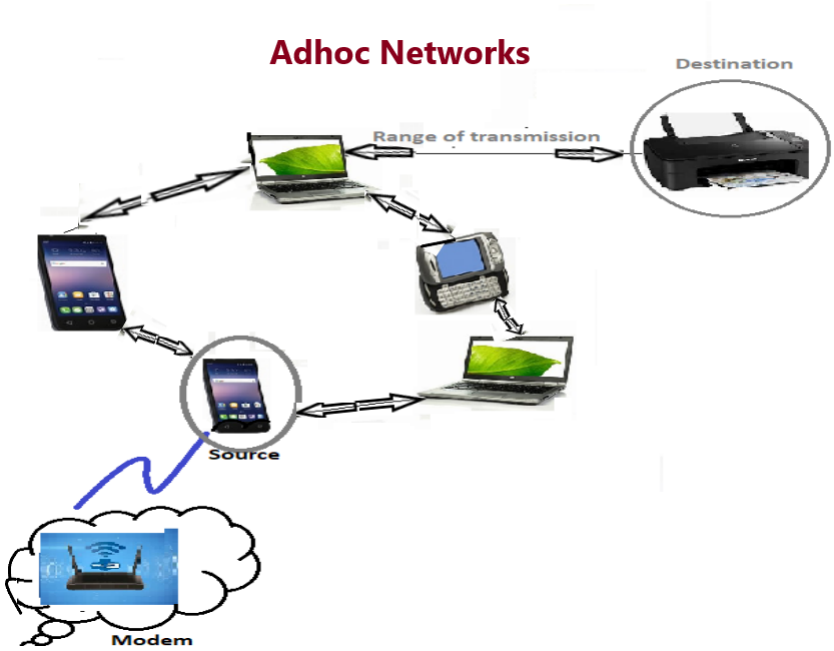

Figure 1.1: Wireless Ad-Hoc networks

- Since ad-hoc WLANs by definition do not need an access point, they are less expensive.
- These networks are appropriate for natural disasters like flooding earthquakes etc.

1.3 Security Issues in WANET

- If the node is within the transmission range, accessing the network is simple. Consequently, WANETs do not offer secure boundary.
- As the devices in network uses battery charge, there is constraints in power supply
- A wireless network's topology may change frequently and continuously. To establish a connection, complicated routing protocols are frequently required. However, they might create brand-new issues that need to be carefully considered.
- Additionally, ad hoc networks do not scale well. It gets more challenging to control an ad hoc network as the number of devices rises.

- Despite of any system malfunctions, the network's services must always be accessible. Denial-of-service (DoS) attacks are frequently used to describe attacks on the integrity and availability of services on networks. Attacks that take place in the transport layer include flooding attacks. If a packet loss goes unnoticed or the attacker is able to permanently stop the packet from being delivered, the attack is deemed successful.

From all the above issues, it is noted that attack detection is a very challenging problem due to the absence of centralised management system. Attacks may be Denial of service, Black hole ,Worm hole etc. Among these a most crucial attack is flooding based denial of service attacks and the developed method uses various supervised machine learning algorithms to detect and classify flooding based denial of service attacks.

1.4 Flooding based DoS Attacks

A Denial of Service (DoS) attack called flooding aims to overwhelm a network or service with a lot of traffic in order to bring it down. Attacks known as "floods" happen when a network or service is overloaded with packets that start incomplete connection requests and can no longer handle real connection requests. The goal of a flood attack is to overwhelm a server or host with connections that cannot be closed, eventually filling the host's memory buffer. A Denial of Service occurs when this buffer is full because no new connections can be made. By sending numerous connection establishment requests to the target, which then allocates resources to maintain state for those connections, flooding attacks deplete the memory resources of a node. RREQ requests are the first step in the route finding process. Even when no connection needs to be made, some nodes continue to send RREQ packets. Attacks involving RREQ flooding are the result, as seen in Fig.1.2. Flooding attack can be categorised in to TCP Flooding,UDP Flooding and ICMP Flooding according to the protocol used in the connection.

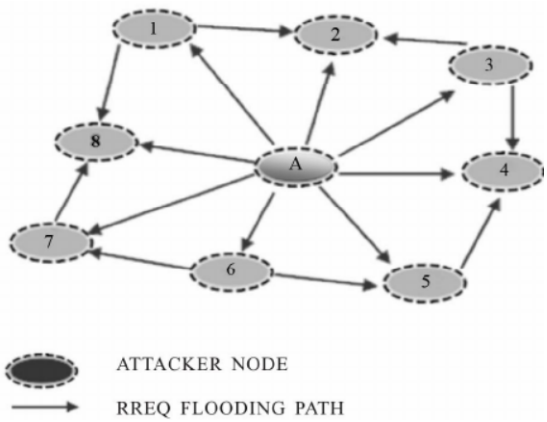

Figure 1.2: flooding attack

1.4.1 TCP Flooding

The Distributed Denial of Service (DDoS) attack known as TCP flood, also known as SYN flood, takes advantage of a portion of the standard TCP three-way handshake to saturate the targeted server with resources and make it unresponsive.In essence, SYN flood DDoS causes network saturation by sending TCP connection requests faster than the targeted machine can handle them. The exchange appears as follows when a client and server establish a typical TCP "three-way handshake":

- By sending a SYN (synchronise) message to the server, the client requests a connection.
- Server acknowledges by returning to the client a SYN-ACK (synchronize-acknowledge) message.
- Connection is made when the client replies with an ACK (acknowledge) message.

In a SYN flood attack, the attacker repeatedly sends SYN packets to all of the server's ports while frequently using a fictitious IP address. The server receives numerous requests to establish communication that seem to be legitimate, but it is unaware of the attack. Each open port sends a SYN-ACK packet in response to each attempt.

If the IP address is spoofed, the malicious client either never receives the SYN-ACK in the first place or fails to send the expected ACK. In either case, the server that is being attacked will wait a while for the acknowledgment of its SYN-ACK packet. Fig.1.3 deals with TCP Flooding.

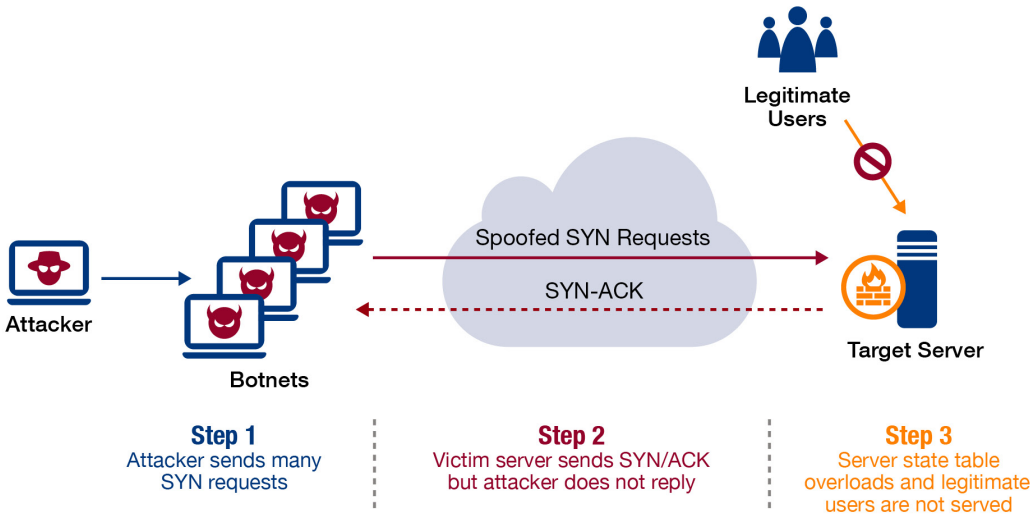

Figure 1.3: TCP flooding

1.4.2 UDP Flooding

A Denial of Service (DoS) attack known as a 'UDP flood' occurs when an attacker floods the targeted host's random ports with IP packets containing UDP datagrams.If there are no applications connected to these datagrams, the receiving host searches for them and sends a "Destination Unreachable" packet back. The system becomes overloaded and unresponsive to additional clients as more and more UDP packets are received and answered. The User Datagram Protocols (UDP) that are used in the UDP flood attack have specific characteristics. The operating system searches for applications that are listening on the specified port if a UDP packet is received on a server. The server must notify the sender if no app is found. The server notifies the sender that the packet could not be delivered using the Internet Control Message Protocol (ICMP) because UDP is a connectionless protocol. UDP flooding is illustrated in fig.1.4. The process that takes place in the event of a UDP flood attack is as follows:

- On the target system, an attacker sends UDP packets to arbitrary ports with a spoof IP sender address.
- For each incoming packet, the system must repeat the following process.Since the UDP packet's specified port was chosen at random, this is typically not the case. However, you should check it just in case.
- Since the IP address has been spoof, send an ICMP "destination unreachable"

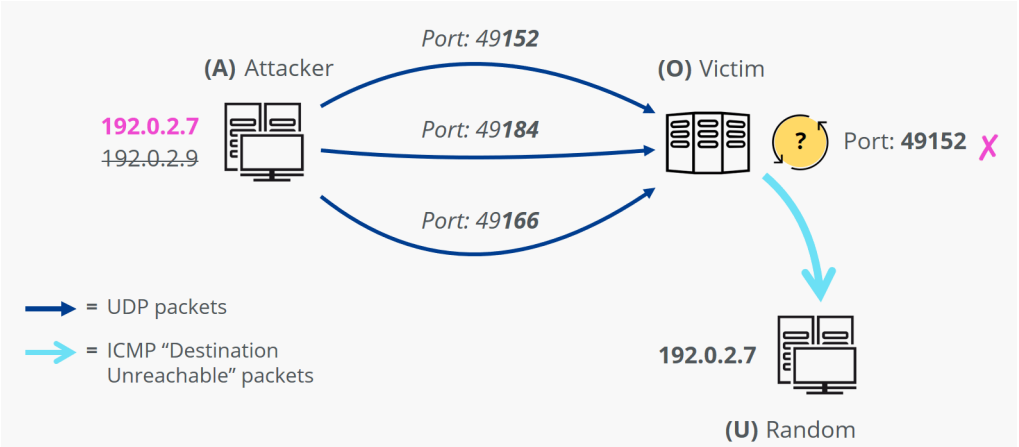

Figure 1.4: UDP flooding

packet to the alleged sender; this packet is typically received by a chance bystander.

1.4.3 ICMP Flooding

ICMP Flood is a type of denial-of-service attack in which the attacker bombards a target with ICMP echo-request packets in an effort to render the target unreachable to regular traffic. The attack changes into a DDoS, or distributed denial-of-service attack, when the attack traffic originates from numerous devices.It is also called ping flooding. An internet layer protocol used by network devices to communicate is the Internet Control Message Protocol (ICMP), which is used in a Ping Flood attack. ICMP is the protocol used by the network diagnostic tools traceroute and ping. Pinging a network device to check on its connectivity and overall health as well as the connection between the sender and the device is frequently done using ICMP echo-request and echo-reply messages.

Server resources are needed to process each ICMP request and send a response. Additionally, bandwidth is needed for the request's incoming message (echo-request) and outgoing response (echo-reply). The Ping Flood attack seeks to either overburden the targeted device's capacity to handle the large volume of requests or to saturate the network connection with erroneous traffic. The attack traffic is significantly increased by having many devices in a botnet target the same internet asset or infrastructure component with ICMP requests, potentially causing a disruption of regular network activity. In the past, attackers would frequently insert a fake IP address to conceal

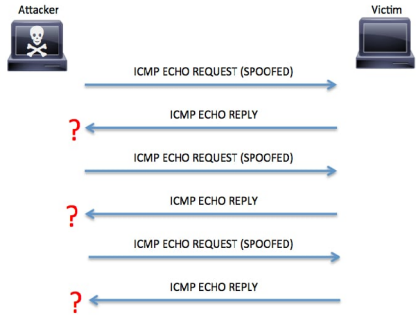

Figure 1.5: ICMP flooding

the sending device. Modern botnet attacks rely on a vast network of unspoofed bots to overwhelm a target's resources rather than the need to disguise the bot's IP.

There are two repeating steps that make up the DDoS form of an ICMP Ping Flood:

- The attacker uses a number of devices to send numerous ICMP echo request packets to the targeted server.
- The targeted server then responds by sending an ICMP echo reply packet to the IP address of each requesting device.

1.5 Research Problem

This thesis work aims to detect and classify flooding attacks in the wireless adhoc network using Machine learning and thereby

1. increase accuracy
2. reduce end to end delay
3. increase true positive rate(TPR)

1.6 Organization of Report

Organization of the thesis report is as follows. Chapter 2 presents Literature review which include various works proposed in the area of Detection of Flooding Attack. Chapter 3 presents the method which is devoloped to detect and classify flooding

attacks and also deals with implementation tool. Chapter 4 deals with brief overview of NSL KDD dataset and Chapter 5 deals with basics of Machine Learning. Chapter 6 deals with machine learning classifiers used in this project. Chapter 7 discuss about the results and chapter 8 concludes the thesis work.

Chapter 2

Literature Review

To identify flooding attacks, a variety of static models have been developed. To detect a high-rate attack, some research has been suggested. Some researchers focused on detecting low-rate flooding attacks. Persistent assaults are high-rate attacks, while pulsed rate attacks are low-rate attacks. The amount of RREQ queries transmitted by a node, TCP packets, received signal intensity, and other factors are used to detect attacks. Each model has its own set of benefits and drawbacks.

N.Nishanth *et al.*,[6] submitted a paper that included RREQ traffic modelling and a Bayesian Inference-based optimal technique for detecting persistent RREQ flooding attacks. Using D-S evidence theory, the method was further enhanced for the detection of high and low rate pulsed RREQ flooding attacks. By transmitting pulse sequences on a regular basis, low-rate denial of service (LDoS) attacks exploit flaws in the TCP protocol to limit TCP throughput and network quality of links. D.Tang *et al.*,[7] proposed a new model based on a weighted Euclidean distance and the advanced Mean Shift clustering technique (WEDMS). The WEDMS algorithm is a step forward from the MS algorithm. For cluster analysis, the MS algorithm often uses Euclidean distance, which evaluates each feature of the data equally and ignores the relevance of distinct qualities. As a result, erroneous data classification occurs to some extent. The weighted coefficients are determined by the degree of data dispersion. Discrete aspects of network traffic with LDoS assaults are more noticeable than those of normal network traffic, it is reported.

A novel robust model for detecting and preventing HELLO flooding attacks was developed by T. Aditya *et al.*,[8] using an optimised deep learning approach. The

steps used in this model include cluster head selection, k-path generation, HELLO flooding attack detection and prevention, and optimal shortest path selection. Following the random cluster head selection and k-paths generation, a few Route Discovery Frequency Vectors, such as Route Discovery Time and Inter Route Discovery Time of each node, are calculated in order to detect the HELLO flooding attack. For the TCP-SYN flood attack, Ram Kumar *et al.*,[9] proposed an adaptive thresholding-based detection and prevention mechanism. The adaptive threshold algorithm is used to calculate dynamic threshold (ATA). This algorithm notifies users when the threshold determined by the adaptive thresholding algorithm is exceeded, thereby assisting in overcoming the drawbacks of static thresholding, such as a high false positive rate. The suggested mechanism is very effective at identifying and thwarting TCP SYN flood attacks by using an adaptive thresholding algorithm.

P.Mohammadi *et al.*,[10] proposed a model based on the DSR routing protocol that detects flooding attacks quickly. The method was created using the suppressing neighbours technique, which detects malicious nodes during the route build up stage. If a malicious node is detected, it is isolated for a while while it investigates its behaviour in the network to avoid a flooding attack at the network layer. If a node is detected as malicious, it will be placed on the detention list for a specified period of time. It will be considered as normal node after this time period has expired.Each node examines the detention field list before transmitting a data packet. Data packets will not be sent to a node that is on the detention list. In any other case, the node will be handled like a typical node and will receive data packets.

G.Truong *et al.*,[11] proposed an efficient source-side DoS detection method which includes an adaptive threshold that takes traffic seasonality into account. In order to identify subtle attack traffic, the threshold for detecting DoS attacks is adaptively adjusted to fluctuating legitimate traffic. Additionally, by comprehending the seasonality of legitimate traffic, it is possible to update the threshold more carefully, even when a subtle attack occurs, and this contributes to a low false positive rate. To distinguish between legitimate traffic and attack traffic and to update the threshold for detecting attacks, the traffic seasonality is obtained by watching a source-side network. To identify and lessen the effects of RREQ flooding attacks on the network, S. Kumar *et al.*,[12] propose a direct trust-based security scheme in which each node evaluates

the trust degree value of its neighbours over a brief period of time by examining the frequency of RREQ packets originated by them. Using the node's trust degree value as an input, the proposed scheme is smoothly expanded to suppress excess RREQ and bogus RREQ flooding packets at one hop neighbours during the route discovery process. This scheme differs from existing methods in that it does not immediately prevent a normal node from using its services because of an increase in the number of RREQ packets in certain unusual situations. Z. Wei *et al.*,[13] propose a trust management scheme in which the trust model has two components: direct observation trust and indirect observation trust. The trust value is derived using Bayesian inference, which is a type of uncertain reasoning when the full probability model can be defined, with direct observation from an observer node. When indirect observation, also known as secondhand information, is obtained from the observer node's neighbours, the trust value is derived using the Dempster–Shafer theory (DST), which is another type of uncertain reasoning when the proposition of interest can be derived indirectly. In MANETs, more accurate trust values of observed nodes were obtained by combining these two components.

Increased network throughput, low end-to-end delay, and other benefits are common among the above models. However, all of them have a high rate of false positives. By incorporating supervised machine learning techniques, the developed model attempts to compensate for this disadvantage.

Chapter 3

Developed Method

The aim of this project is to detect and classify flooding attack with high accuracy.The suggested approach make use of supervised machine learning techniques and can be seen in fig. 3.1. It consists of Data collection, Data Preprocessing, Splitting data , Training Model and Model Evaluation. The first step is gathering the data. NSL KDD training dataset makes up the information gathered. It contains normal data and 21 attack types, which are divided into 4 groups by DoS, r2l, probe, and u2r. Since our goal is to accurately classify flooding attacks, we are unable to use this dataset directly. The collected dataset will need to be updated in order to achieve that. Preprocessing is the next step. It is the phase of machine learning that is most important. The model will forecast incorrectly if we are unable to train with accurate data. Details of the preprocessing are shown in Fig. 3.

The dataset obtained in.arff (attribute relation file format) format is first converted to .csv (comma separated value) format during the preprocessing stage. The attacks in the dataset are then classified as DoS, Probe, U2R, and R2L in accordance with Table 1. The classification of DoS into TCP flooding, ICMP flooding, and UDP flooding is the following step. DoS attacks are transformed into these flooding attacks in accordance with Fig. 4. A new column has been added to the collected dataset, and the output class labels now include information about TCP, UDP, and ICMP flooding attacks. The process involves examining class labels and prototypes. The class is renamed to TCP flooding if the attack is a DoS and the prototype is TCP. Otherwise, it is categorised as flooding using either UDP or ICMP depending on the given prototype. The dataset can now be used to train supervised machine learning

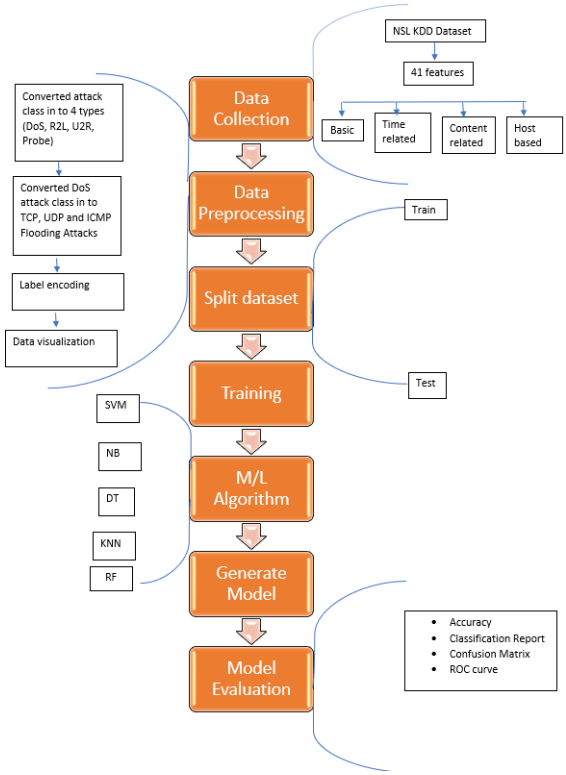

Figure 3.1: Developed Model

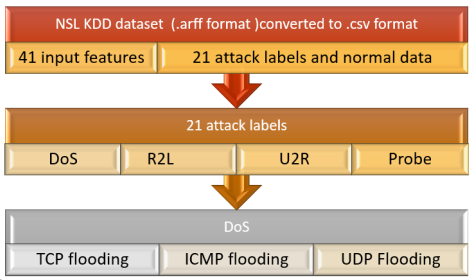

Figure 3.2: Preprocessing

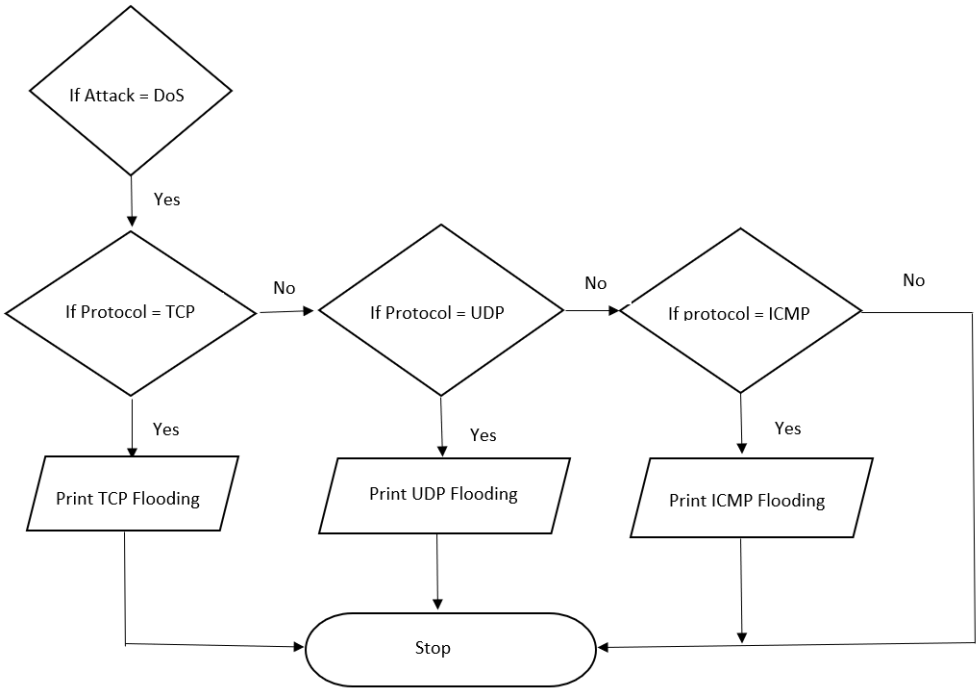

Figure 3.3: Updation in NSLKDD Dataset

models and has the ability to classify flooding attacks. Then To make categorical variables machine-readable, label encoding is used to transform them into a numerical format. Four categorical features—protocol, service, flag, and output class—are included in the dataset.

The dataset is divided into Train and Test in the following phase. Eighty pecentage of the data is used for training the models and remaining twenty percentage is kept aside for testing. Following that, machine learning algorithms (SVM, KNN, NB, DT, and RF) are applied to the training dataset for comparison and classification. From variety of available kernels in SVM, rbf kernel is used for classification because it has seven output labels and 41 input features and it is not linearly seperable. Hyperplanes divide the output labels, such as normal, TCP flooding, UDP flooding, ICMP flooding, R2L, Probe, and U2R in seperate sections. Hyperplane is fixed by maximising the distance between margins and points that appear close to these margins are called support vectors. All these 41 features and output labels are provided to the KNN as training input. When an unknown input is received in testing, it determines which k neighbours are the most close by calculating distance between each samples and then predicts the label with the most points. Here 'k' is fixed as 5 and find out most

Category	Attacks present in NSLKDD
DOS	Back, Land, Neptune, Pod, Smurf, Teardrop, Mailbomb, Processtable, Udpstorm, Apache2, Worm
Probe	Satan,IPsweep, Nmap, Portsweep, Mscan, Saint
R2L	Guess_password, Ftp_write, Imap, Phf, Multi hop, Warezmaster, Xlock, Xsnoop, Snmpguess, Snmpgetattack, Httptunnel, Sendmail, Named
U2R	Buffer_overflow, Loadmodule, Rootkit, Perl, Sqlattack, Xterm, Ps

Table 3.1: Attack types in NSL KDD Dataset

matched 5 datapoints nearest to the input sample . Then find out mostly voted labels and predicts the output class.

Prior probabilities of occurrence of each feature given that each attack label are first learned in Naive Baye's. The Bayes theorem can then be used to predict unknown data to classify the attacks. Using information gain, first locate the root node of the Decision Tree from the 41 features. The likelihood of each feature is then calculated, leading to the creation of an output classification. Random Forest is a group of Decision Trees that employs the ensembling technique. The given datasets are splitted first and then apply to each of the decision tree. Here there are a group of 100 decision Trees and each produce their own predictions. Then Random Forest finally predicts the class which corresponds to majority voting. Then evaluation matrices like accuracy, confusion matrix, classification report, and ROC are used to assess each classifier's performance. The results are discussed in chapter 7.

Chapter 4

NSL KDD dataset

Only selected records in the KDD dataset are included in the NSL KDD dataset. This chosen dataset offers a thorough examination of various machine learning techniques for intrusion detection and can be used by researchers as a useful benchmark data set [14] through [17]. The training dataset contains 21 different attacks, compared to 37 in the test dataset. The known attack types are those found in the training dataset, whereas the novel attacks are those found in the test dataset but not in the training datasets. In Dataset, the attack types are divided into four categories: DoS, Probe, U2R, and R2LAttacks. Table 1 can be used to group attack types.

A denial-of-service (DoS) attack happens when an attacker successfully overloads memory and computing resources or prevents authorised users from accessing a machine. In an R2L attack, the attacker bypasses logging in and attempts to gain access locally as a specific machine's user. In order to accomplish this, the attacker uses a vulnerability to send a packet to a remote machine over a network. By accessing a regular user account, an attacker can use User to Root (U2R) attacks to take control of a system and exploit a vulnerability. In Probe, an attacker can search a network of computers for known vulnerabilities or collect all the necessary information about the target system.

There are 41 attributes and one attack class in the dataset that are classified as normal or specific attack types. Basic attributes, content attributes, time-based traffic attributes, and host-based traffic attributes are among the 41 features divided into four categories. The value of these characteristics is primarily determined by their continuous, discrete, and symbolic value.

Table 4.1: Basic attributes in NSL KDD

Attribute No	Attributes
1	Duration
2	Type of Protocol
3	Destn service
4	Flag
5	Source bytes
6	Dstn bytes
7	Land
8	Wrong fragment
9	Urgent

4.0.1 Basic Attributes

Basic attributes are attributes in the NSL-KDD dataset that are derived from TCP/IP connections. These characteristics cause an implicit detection delay (for example, time slot, protocol type, service, and so on). Table 2 deals with 9 basic features present in NSL KDD. 'Duration' is the length of time duration of the connection

'Type of protocol' is the protocol used in the connection.It may be TCP,UDP or ICMP .Transmission Control Protocol is referred to as TCP. The packet transmission from source to destination is made easier by this transport layer protocol. Since it is a connection-oriented protocol, communication between computing devices in a network begins with the connection being established. This protocol is used with an IP protocol, so together, they are referred to as a TCP/IP. The main functionality of the TCP is to take the data from the application layer. The data is then divided into several packets, each of which is given a number, before being transmitted to the destination. The packets are reassembled by the TCP and sent to the application layer from the other side. As we know that TCP is a connection-oriented protocol, so the connection will remain established until the communication is not completed between the sender and the receiver. The User Datagram Protocol, or UDP, is a communication protocol that is employed throughout the network for particularly time-sensitive transmissions like video playback or DNS lookups. It speeds up communications by not formally establishing a connection before data is transferred. This enables the transfer of data very quickly, but it can also result in packets being lost in transit, opening up opportunities for DDoS attacks and other forms of exploitation. Devices

in a network can communicate issues with data transmission using the Internet Control Message Protocol (ICMP). In accordance with this definition of ICMP, one of the main purposes of ICMP is to check whether data is reaching its target at the appropriate time. Because of this, ICMP is crucial to the error reporting process and to testing a network's data transmission efficiency. Distributed denial-of-service (DDoS) attacks can also be carried out using it, though.

'Destn service' is defined as service used in Destination network.There are seventy services in NSL KDD dataset.'Flag' represents Status of the connection (Normal or Error). There are 10 flags . Normal connection establishment and termination are referred by 'SF'. 'REJ' refers to connection attempts that were rejected. 'SH' stands for a partially opened connection. 'S0' indicates a connection attempt, but no response is received. If the flag is 'S1', a connection has been made but not ended. 'S2' denotes that the originator established the connection and terminated it because the destination did not respond. 'S3' also shows the establishment of connections and their termination by sources and destinations that do not appear to be listening for connections. 'RST0' stands for source connection reset. 'RSTR' denotes a destination that has reset the connection. 'OTH' denotes a partially established connection that was left open.

'Source bytes' is another feature that shows the quantity of data bytes sent over a single connection from source to destination. 'Dstn bytes' is a representation of the total number of data bytes sent over a single connection from destination to source. 'Land' value ranges from 0 to 1. This variable has a value of 1 if the source and destination IP addresses and port numbers match. The term "wrong fragment" refers to all incorrect fragments in this context. Number of urgent packets in the connection is what 'urgent' deals with. The urgent bit is activated in packets that are considered urgent.

4.0.2 Content Attributes

Content attributes (for example, whether logged in, root shell, whether hot login etc.) access the original TCP packets' payload using domain-specific knowledge. Table 3 lists the 13 content features that are available. The 'hot' feature represents the quantity of "hot" indicators in the content, such as opening a system directory, making

Table 4.2: Content Attributesin NSL KDD

Attribute No	Attributes
10	Hot
11	Num of failed logins
12	whether Logged in
13	Num of compromised
14	Root shell
15	Su root attempted
16	Num of root
17	Num of file creations
18	Num of shells
19	Num of access files
20	Num of outbound cmds
21	whether hot login
22	whether guest login

programmes, and running programmes. Number of unsuccessful login attempts is indicated by 'Num of failed logins'. 'Whether Logged in' displays the login status. If you successfully log in, the value is 1, otherwise it's 0. When rootshell is obtained, status of 'root shell' is 1, otherwise it is 0. The quality 'Su root attempted' returns 1 if the su root command was used or attempted, and 0 otherwise. Number of root accesses or the number of operations carried out as a root in the connection are defined by 'Num of root'. 'Num of file creations' stands for the number of file creation operations performed within the connection, and 'Num of shells' stands for the number of shell prompts. 'Num of access files' describe the number of operations on access control files. The 'Num of outbound cmds' variable is used to represent the number of outbound commands in an ftp session. Another feature called "whether hot login" returns a value of 1 for root or admin logins and a value of 0 otherwise. The 'whether guest login' feature shows 1 if the login is a guest login and 0 otherwise.

4.0.3 Time based Attributes

Time-based traffic attributes, such as 'srvc serror rate' and 'srvc rerror rate', are created specifically to capture the dataset's one distinctive property—that is, features that develop over a two-second temporal window. This category includes 9 features.'Connection Count' is the sum of all connections made in the previous two

Table 4.3: Time related attributes in NSL KDD

Attribute No	Attributes
23	Connection count
24	Srvccount
25	Serror rate
26	Srvc serror rate
27	Rerror rate
28	Srvc rerror rate
29	Same srvc rate
30	Diff srvc rate
31	Srvc diff host rate

seconds to the same host as the current connection. 'Srvccount' is the number of connections made in the previous two seconds to the same service (port number) as the current connection. 'Serror rate' is the proportion of connections among those gathered in the feature called 'Connection count' that have activated the flags (S0, S1, S, or S3). 'Srvc serror rate ' is the proportion of connections among those included in 'Srvccount' that have activated the four flags (S0, S1, S, or S3) . 'Rerror rate' is the proportion of connections among all connections combined in count that have activated the flag (REJ). 'Srvc rerror rate' provides the proportion of connections out of all connections combined in 'srvccount' that have the flag (REJ) active. The proportion of connections among all connections combined in 'connection count' that were to the same service is known as the 'same srvcrate'. 'Diff srvc rate is the proportion of connections among all connections combined in 'Connection count' that were to different services. The percentage of connections among those included in the srvccount that were made to various destination machines is known as the srvcdiff host rate.

4.0.4 Host based features

Few attacks in the NSL-KDD dataset's span intervals longer than two seconds. Host-based traffic features are designed to access all attacks which span longer than 2 second intervals that have the same destination host as the current connection. (example:Dstn host count, Dstn host srvc count). This category includes 10 features. "Dstn host count" indicates the number of connections with the same destination host

Table 4.4: Host based attributes in NSL KDD

Attribute No	**Attributes**
32	Dstn host count
33	Dstn host srvc count
34	Dstn host same srvc rate
35	Dstn host diff srvc rate
36	Dstn host same src port rate
37	Dstn host srvc diff host rate
38	Dstn host serror rate
39	Dstn host srvc serror rate
40	Dstn host rerror rate
41	Dstn host srv rerror rate

IP address, and "Dstn host srvc count" indicates the number of connections with the same port number. 'Dstn host same srvc rate' is the percentage of connections that were to the same service, among the connections aggregated in 'dstn host count'. The percentage of connections among the connections totalled in dstn host count that were to various services is known as 'dstn host diff srvc rate'. 'Dstn host same src port rate' is the percentage of connections that were to the same source port, among the connections aggregated in 'dstn host srvc count'. 'Dstn host srvc diff host rate' is the percentage of connections that were to different destination machines, among the connections aggregated in 'dstn host srvc count'. The 'Dstn host serror rate' is the percentage of connections that have activated the flag (S0, S1, S, or S3) among the connections aggregated in 'dstn host count'. The ' Dstn host srverror rate' is the proportion of connections out of all connections combined in 'dstn host srvc count, that have the flag (S0, S1, S, or S3) activated. The proportion of connections among those aggregated in dst host count that have the flag (REJ) activated is represented by 'dstn host srv rerror rate'.

Chapter 5

Machine Learning

The scientific discipline of machine learning enables computers to learn without explicit programming. One of the most intriguing technologies that has ever been developed is machine learning. The ability to learn is what, as the name suggests, gives the computer a more human-like quality. Today, machine learning is being actively used, possibly in a lot more places than one might think.Machine learning issues can be categorised in a variety of ways. They are based on characteristics of learning signal and output expected

Based on the characteristics of the learning signal

This classification depends on how the machine is trained to build a model .They are

- **Supervised learning**: A "teacher" presents sample inputs and the desired outputs to the computer with the intention of teaching it a general rule that maps inputs to outputs. The training procedure is carried out repeatedly until the model's accuracy on the training set reaches the desired level. In supervised learning, the data is labelled.
- **Unsupervised learning**:The learning algorithm is not given any labels; instead, it is left to its own devices to identify structure in the input. It is employed to divide the population into various groups. Unsupervised learning may serve as a primary objective (discovering hidden patterns in data). Unsupervised learning uses unlabeled data.

- **Semi-supervised learning**:Semi-supervised learning problems are those in which only a portion of the input data is labelled despite the large amount. These issues fall somewhere between supervised learning and unsupervised learning. For instance, consider a photo archive where only a small percentage of the images are labelled (such as dog, cat, or person) and the vast majority are not.

- **Reinforcement learning**:A computer programme must accomplish a specific task while interacting with a dynamic environment (such as driving a vehicle or playing a game against an opponent). As the programme moves through its problem space, feedback in the form of rewards and penalties is given.

Based on the output that a machine learning system is expected to produce

This classification is based on the nature of expected output and they are

- **Classification**: The learner is required to create a model that categorises unseen inputs into one or more of the inputs' two or more classes (multi-label classification). Usually, this is handled under supervision. When email (or other) messages are the inputs and the classes are "spam" and "not spam," that is an example of classification.

- **Regression**: Although the outputs are not discrete but rather continuous, it is still a supervised learning problem.

- **Clustering**: A grouping of inputs is required in this situation. The groups are not known in advance, unlike in classification, so this is typically an unsupervised task.

- **Dimensionality reduction**: By translating them into a lower-dimensional space, it simplifies inputs. A related issue is topic modelling, in which a computer programme is tasked with identifying the documents that cover similar topics from a list of human language documents.

5.1 Terms used in machine learning

1. **Model**: A model is a particular representation that has been discovered from data using a machine learning algorithm. A hypothesis is another name for a model.

2. **Feature**: A feature is a specific, quantifiable characteristic of data. A feature vector can be used to conveniently describe a collection of numerical features. The model receives feature vectors as input data. For instance, characteristics such as colour, smell, taste, etc. may be used to predict a fruit. Note: For efficient algorithms, selecting informative, discriminating, and independent features is a key step. Use a feature extractor to typically take the pertinent features out of the raw data.

3. **Target (Label)**: The value that the model must predict is known as a target variable or label. The label for each set of input in the fruit example from the features section would be the fruit's name, such as apple, orange, banana, etc.

4. **Training**: To create a model (hypothesis) that will later map new data to one of the categories trained on, a set of inputs (features) and expected outputs (labels) are provided.

5. **Prediction**: When the model is complete, it can be fed a set of inputs from which it will predict an outcome (label).

Chapter 6

Machine Learning Classifiers

An algorithm automatically arranges or categorises data into one or more of a set of "classes." One of the most prevalent examples is an email classifier, which scans, emails and filters them according to whether they are spam or not. They can help automate processes and save a tonne of time and money. In this project, SVM, KNN classifiers, Naive Baye's Classifier, Decision Tree and Random Forest are used.

6.1 Support Vector Machines

SVMs[18] are machine learning algorithms used for regression and classification tasks. SVMs are a potent machine learning algorithm used for outlier detection, regression, and classification. A model is created by an SVM classifier that classes new data points into one of the predetermined categories. As a result, it can be thought of as a binary linear non-probabilistic classifier. SVMs are applicable to linear classification tasks. Using the kernel trick, SVMs can effectively perform non-linear classification in addition to linear classification. It allows us to automatically map the inputs into large feature spaces. Several SVM jargons include,

- **Hyperplane**: A decision boundary known as a hyperplane divides a set of data points with various class labels. The maximum margin hyperplane is used by the SVM classifier to divide the data points into groups. This hyperplane is referred to as the maximum margin hyperplane, and the maximum margin classifier is the linear classifier it defines.

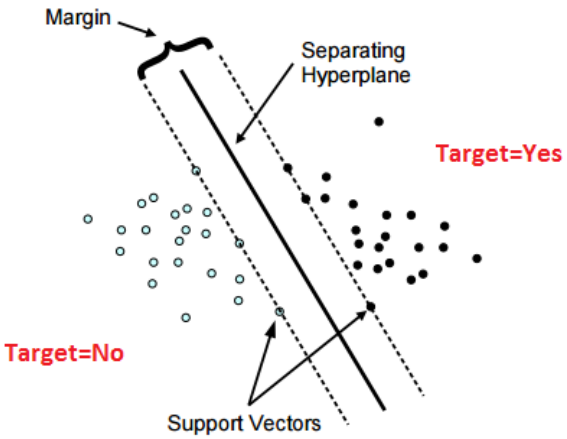

Figure 6.1: SVM classifier

- **Support Vectors**: The sample data points that are closest to the hyperplane are called support vectors. By calculating margins, these data points will help define the separating line or hyperplane.

- **Margin**: The distance between the two lines on the closest data points is known as a margin. It is calculated as the perpendicular distance between the line and the nearest data points or support vectors. We aim to maximise this separation gap in SVMs to obtain the highest possible margin.

The terms used in the SVM classifier are explained in Fig. 4.1, and Fig. 4.2 clearly demonstrates the idea of maximum margin and maximum margin hyperplane. In the following two steps, SVM looks for the hyperplane with the greatest margin.

1. Create hyperplanes that separate the classes as best you can. Numerous hyperplanes could categorise the data. The best hyperplane that best depicts the greatest margin of separation between the two classes should be sought out.

2. Select the hyperplane so that the distance to the support vectors on either side is as far away from it as possible. A maximum margin classifier is a linear classifier that is defined by such a maximum margin hyperplane, if one exists..

In actuality, a kernel is used to implement the SVM algorithm. It employs a method known as the kernel trick. A kernel is essentially just a function that maps data to a higher dimension with separable data, to put it simply. A kernel increases the dimension of a low-dimensional input data space. Therefore, by including more

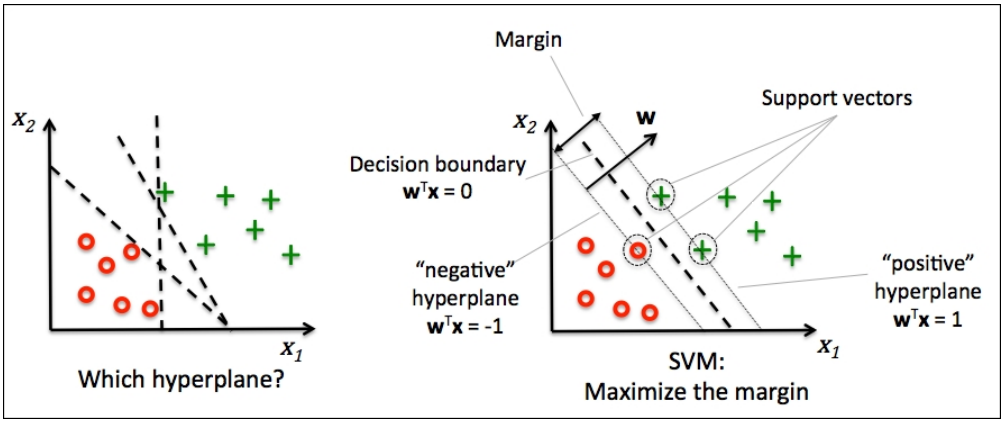

Figure 6.2: maximum margin and maximum margin hyperplane

dimensions, it transforms non-linear separable problems into linear separable problems. We can thus create a classifier that is more accurate thanks to the kernel trick. As a result, it helps with non-linear separation issues. There are four widely used kernels: the linear kernel, the polynomial kernel, the RBF kernel (also known as the Gaussian kernel), and the sigmoid kernel.

- When the data can be separated linearly, a linear kernel is used. It implies that lines can be used to divide up data. It is one of the most widely used kernels. It is typically used when a dataset contains a large number of features. For text classification purposes, linear kernel is frequently employed. It is typically quicker to train using a linear kernel.

- The similarity of vectors (training samples) in a feature space over the polynomials of the original variables is represented by a polynomial kernel. The polynomial kernel examines combinations of the input samples in addition to the given features of input samples to determine how similar they are.

- A general-purpose kernel is the radial basis function kernel. When we are unfamiliar with the data, we use it.

6.2 KNN classifier

The most straightforward machine learning algorithm is called k Nearest Neighbors, or kNN [19]. It is a non-parametric algorithm used for regression and classification problems. Non-parametric means that no data distribution assumption is

necessary. Therefore, kNN does not require the use of any underlying assumptions. The k closest training examples in the feature space are the input for classification and regression tasks, respectively. Whether kNN is applied for classification or regression determines the results.

- In kNN classification, the output is a class membership. The given data point is classified based on the majority of type of its neighbours. The data point is assigned to the most frequent class among its k nearest neighbours. Usually k is a small positive integer. If k=1, then the data point is simply assigned to the class of that single nearest neighbour.
- In kNN regression, the output is simply some property value for the object. This value is the average of the values of k nearest neighbours

6.3 KNN algorithm

It simply calculates the distance between a sample data point and all the other training data points. The distance can be Euclidean distance or Manhattan distance. Then, it selects the k nearest data points where k can be any integer. Finally, it assigns the sample data point to the class to which the majority of the k data points belong.

k is the number of nearest neighbours in the kNN algorithm. K is typically an odd number since it aids in determining the majority of the class. The algorithm is called the nearest neighbour algorithm when k=1.

At the time of building the model, we must choose a parameter called the number of neighbours (k) in the kNN. The most difficult problem in kNN is choosing the best value for k. A low value of k indicates that noise will have a greater impact on the outcome. The likelihood of overfitting is therefore very high. The construction of the kNN model is computationally time-consuming for large values of k. Additionally, a high value of k will result in a smoother decision boundary, which will result in a lower variance but a higher bias.

If there are more even classes than odd classes, the data scientists choose an odd value for k. Choose the value of k using the elbow method. Utilize the Cross Validation

technique to enhance the results. Check the performance of the kNN algorithm with various k values using the cross-validation technique. The model that provides good accuracy can be regarded as the best option. Depending on the circumstances of each case, testing the outcome after applying each possible value of k is occasionally the best course of action.

6.4 Naive Baye's Classifier

A supervised learning algorithm called Naive Bayes[20] is employed for classification tasks. It is also known as Naive Bayes Classifier for this reason.Similar to other supervised learning algorithms, naive bayes predicts a target variable using features. The main distinction is that naive bayes makes the assumption that features are unrelated to one another and do not correlate. In actuality, this is not the case.

Naive bayes classifier calculates the probability of a class given a set of feature values (i.e. p(b_i / a_1, a_2 , ... , a_n)).

$$p(b_i/a_1, a_2, ...a_n) = p(a_1, a_2, ...a_n/b_i).p(b_i)/p(a_1, a_2, ...a_n) \tag{6.1}$$

p(a_1, a_2 , ... , a_n) means the likelihood that a certain set of features, given a class label, will occur. The naive Bayes algorithm makes the assumption that each feature is independent of the others..So,The probability of a class (p(b_i)) is very simple to calculate.That is

$$p(b_i) = (no\_of\_observations\_with\_class\_b_i)/(no\_of\_all\_observations) \tag{6.2}$$

Under the assumption of features being independent, p(a_1, a_2 , ... , a_n / b_i) can be written as:

$$p(a_1, a_2, ...a_n/b_i) = p(a_1/b_i).p(a_2/b_i).p(a_3/b_i).....p(a_n/b_i) \tag{6.3}$$

Considering the class label, the conditional probability for a single feature (i.e. p(a_1/b_i))can be estimated more easily from the data. Probability distributions of features for

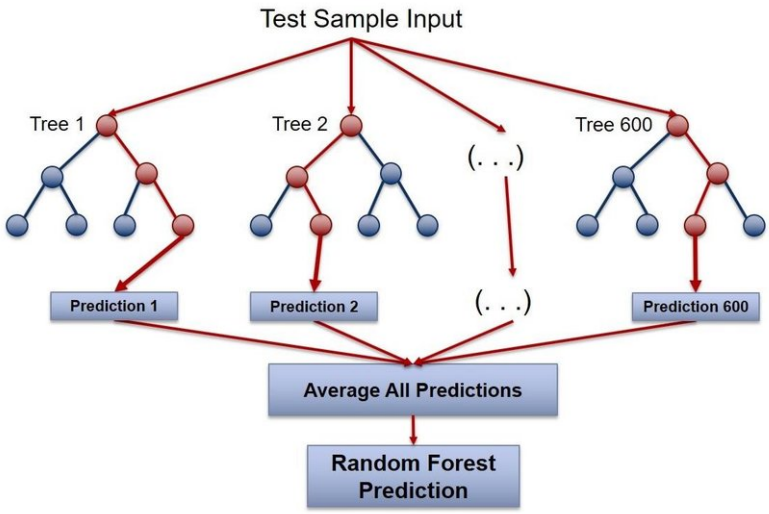

Figure 6.3: Random forest

each class must be stored separately by the algorithm.

6.5 Random Forest

A popular machine learning algorithm called random forest[22] combines the output of various decision trees to produce a single outcome. Its widespread use is fueled by its adaptability and usability because it can solve classification and regression issues.

There are three main hyperparameters for random forest algorithms that must be set prior to training. Node size, tree count, and sampled feature count are a few of these. From there, classification or regression issues can be resolved using the random forest classifier.

Each decision tree in the ensemble that makes up the random forest algorithm is made of a data sample taken from a training set with replacement known as the bootstrap sample. One-third of that training sample, or the out-of-bag (oob) sample, is set aside as test data. The explanation of random forest is provided in Fig. 4.3.

6.6 Decision Tree

Classification and regression issues can be resolved using the supervised learning technique known as a decision tree [21], but it is frequently chosen to do so. It is a

tree-structured classifier, where each leaf node represents the classification outcome and internal nodes represent the features of a dataset. The two nodes in a decision tree are the Decision Node and Leaf Node. Decision nodes are used to make decisions and have multiple branches, whereas Leaf nodes are the outcomes of decisions and do not have any additional branches. The test or decision-making process is carried out using the features of the provided dataset. It is a graphical method of obtaining all solutions that are possible under the given circumstances to a decision or a problem. Because it starts with the root node and expands on succeeding branches to form a structure resembling a tree, it is known as a decision tree. A tree is built using the CART algorithm, which stands for Classification and Regression Tree Algorithm. Simply asking a question and then segmenting the tree into subtrees based on the answer (Yes/No) is all that a decision tree does.

Chapter 7

Results and Discussion

This project is being implemented using Python. The Integrated Development Environment used is Pycharm. Code completion and inspection, advanced debugging, and support for web programming and frameworks like Django and Flask are some special features that Pycharm offers.The classifiers are trained using 80% of the NSLKDD data that has been previously processed. The evaluation metrics are accuracy, classification report, confusion matrix, and ROC is made from remaining 20% and is labelled as test dataset. The accuracy of a dataset is equal to the total number of accurate predictions divided by the total number of predictions made. Table 7.1 gives comparison of accuracy of different machine learning models and it is noted that both DT and RF comes with highest accuracy.

The counts of test records that the classification model correctly and incorrectly predicted are used to assess the performance of the model. The confusion matrix provides a more insightful picture which is not only the performance of a predictive model, but also which classes are being predicted correctly and incorrectly, and what type of errors are being made. To illustrate, we can see how the 4 classifica-

Table 7.1: Comparison of different ML models

Sl.No	**Models**	**Accuracy**
1	SVM	0.94
2	NB	0.86
3	KNN	0.99
4	DT	0.9977
5	RF	0.9987

Table 7.2: Representation of labels in Confusion Matrix

Labels in Confusion Matrix	Class
0	Normal
1	TCP Flooding
2	R2L
3	Probe
4	ICMP FLooding
5	UDP Flooding
6	U2R

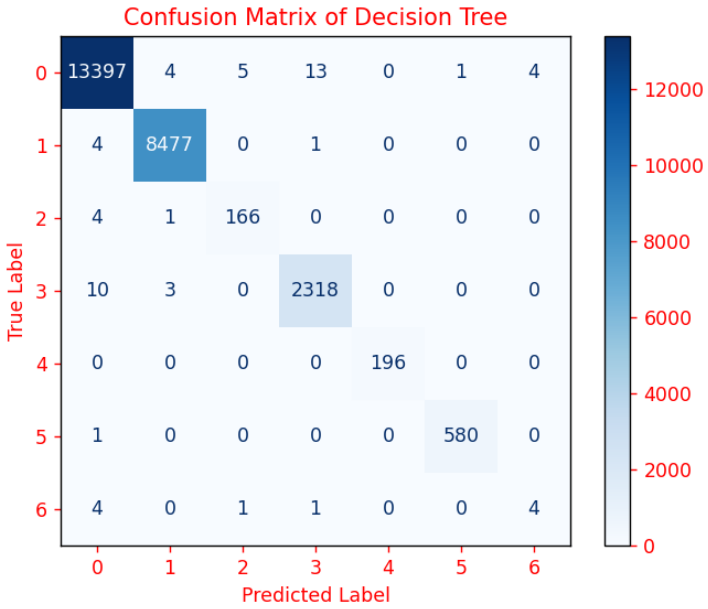

Figure 7.1: Confusion Matrix of Decision Tree

tion metrics are calculated (TP, FP, FN, TN), and our predicted value compared to the actual value in a confusion matrix is clearly presented in the confusion matrix table.Confusion matrix for different models obtained are shown in figures 7.1 to 7.5. From the confusion matrix of RF, it can be noted that among 13,424 normal samples 3 samples are incorrectly predicted. From 8482 samples of TCP Flooding , 8479 are correctly predicted. All 196 samples of UDP Flooding are correctly predicted and in case of ICMP Flooding, 1 sample is predicted incorrectly.In confusion matrix representation of each attack is given in the Table 7.2.

Recall (also known as sensitivity), precision, specificity, accuracy, and most importantly the AUC-ROC Curve can all be measured using the classification report. One important method for displaying a classification model's performance visually is the

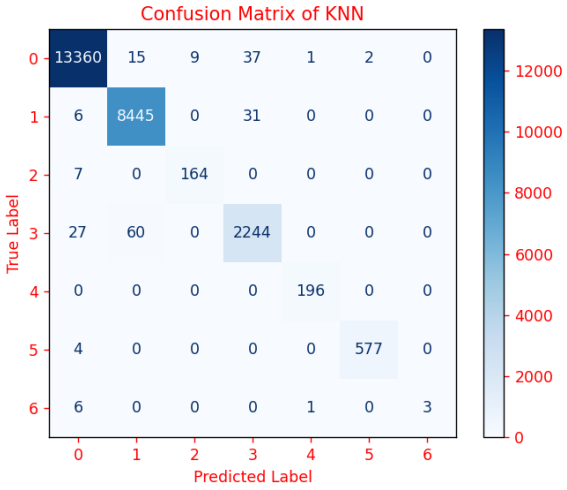

Figure 7.2: Confusion Matrix of KNN

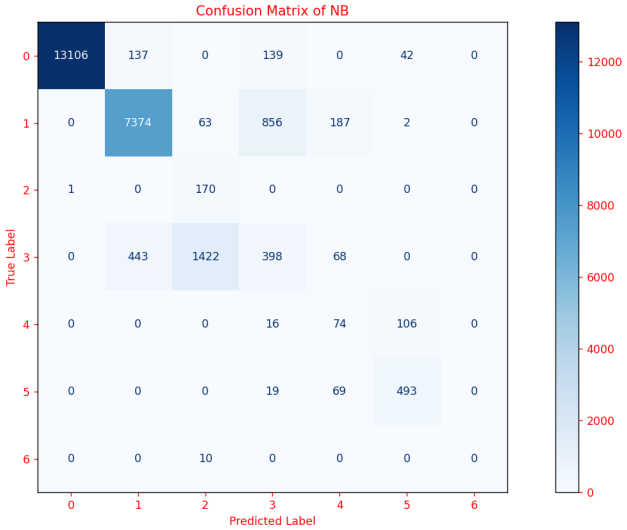

Figure 7.3: Confusion Matrix of Naive Baye's

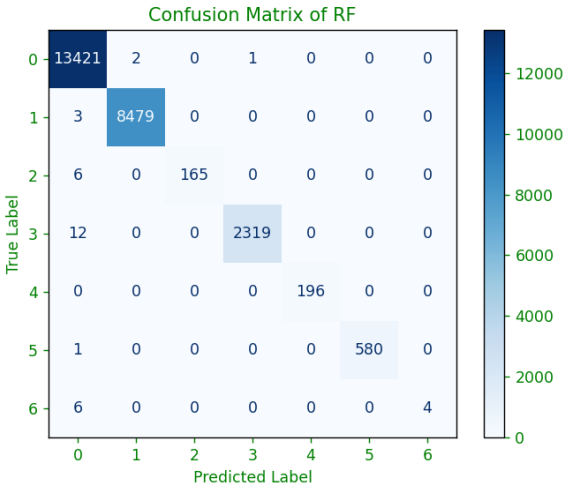

Figure 7.4: Confusion Matrix of Random Forest

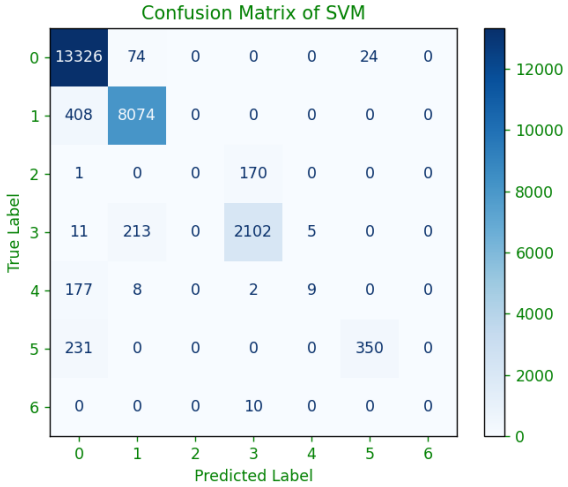

Figure 7.5: Confusion Matrix of SVM

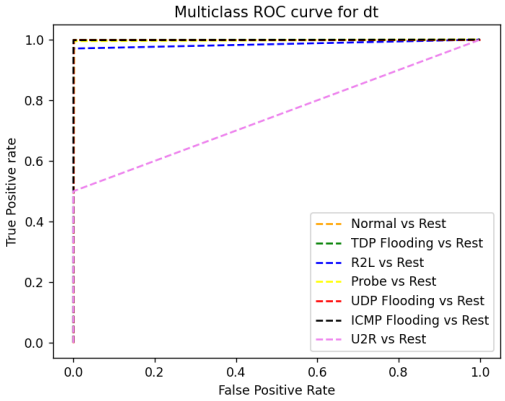

Figure 7.6: ROC of Decision Tree

ROC curve. Using various probability thresholds, it summarises the trade-off between the true positive rate (tpr) and false positive rate (fpr) for a predictive model. The ROC of various models is displayed in figures 7.5 to 7.8. As a function of the model's threshold for classifying a positive case, a ROC curve compares the true positive rate (tpr) and false positive rate (fpr).TPR is the ratio of TP to (TP+FN) and FPR is defined as FP to (FP+TN).The ROC plots contain seven different plots for each classifiers as it plots one positive class vs rest(negative). When comparing all classifiers, it can be seen that RF gives accurate predictions from starting itself and Naive Baye's gives worst performance.

Table 7.2 compares the classification reports for each class and classification reports of each model is shown in figures Here too, we can see that DT & RF consistently deliver the best results in terms of precision, recall, and f1 score. Overall, we can state

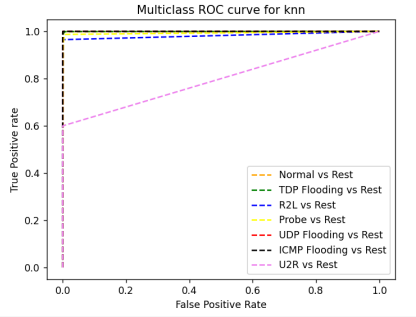

Figure 7.7: ROC of KNN

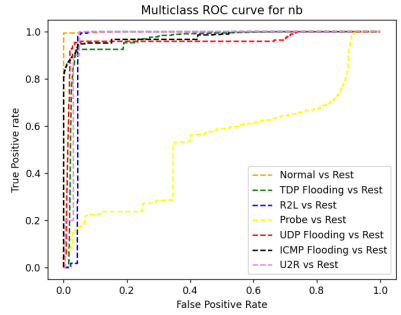

Figure 7.8: ROC of Naive Baye's Classifier

that the RF classifier achieves the highest performance.

	precision	recall	f1-score	support
0	0.94	0.99	0.97	13424
1	0.96	0.95	0.96	8482
2	0.00	0.00	0.00	171
3	0.92	0.90	0.91	2331
4	0.64	0.05	0.09	196
5	0.94	0.60	0.73	581
6	0.00	0.00	0.00	10
accuracy			0.95	25195
macro avg	0.63	0.50	0.52	25195
weighted avg	0.94	0.95	0.94	25195

Figure 7.9: Classification report of SVM

Table 7.3: Comparison of classification report

Attack class	ML model	Precision	Recall	F1 score
No attack	SVM	0.94	0.99	0.97
	NB	1	0.98	0.99
	KNN	1	1	1
	DT	1	1	1
	RF	1	1	1
TCP Flooding	SVM	0.96	0.95	0.96
	NB	0.93	0.87	0.90
	KNN	0.99	1	0.99
	DT	1	1	1
	RF	1	1	1
UDP Flooding	SVM	0.94	0.6	0.73
	NB	0.19	0.38	0.25
	KNN	0.99	1	0.99
	DT	1	1	1
	RF	1	1	1
ICMP Flooding	SVM	0.64	0.05	0.09
	NB	0.77	0.85	0.81
	KNN	1	0.99	0.99
	DT	1	1	1
	RF	1	1	1

classification report of NB Classifier is :

	precision	recall	f1-score	support
0	1.00	0.98	0.99	13424
1	0.93	0.87	0.90	8482
2	0.10	0.99	0.19	171
3	0.28	0.17	0.21	2331
4	0.19	0.38	0.25	196
5	0.77	0.85	0.81	581
6	0.00	0.00	0.00	10
accuracy			0.86	25195
macro avg	0.47	0.61	0.48	25195
weighted avg	0.89	0.86	0.87	25195

Figure 7.10: Classification report of Naive Baye's Classifier

classification report of KNN Classifier is :

	precision	recall	f1-score	support
0	1.00	1.00	1.00	13424
1	0.99	1.00	0.99	8482
2	0.95	0.96	0.95	171
3	0.97	0.96	0.97	2331
4	0.99	1.00	0.99	196
5	1.00	0.99	0.99	581
6	1.00	0.30	0.46	10
accuracy			0.99	25195
macro avg	0.98	0.89	0.91	25195
weighted avg	0.99	0.99	0.99	25195

Figure 7.11: Classification report of KNN

classification report of DT Classifier is :

	precision	recall	f1-score	support
0	1.00	1.00	1.00	13424
1	1.00	1.00	1.00	8482
2	0.97	0.97	0.97	171
3	0.99	0.99	0.99	2331
4	1.00	1.00	1.00	196
5	1.00	1.00	1.00	581
6	0.50	0.40	0.44	10
accuracy			1.00	25195
macro avg	0.92	0.91	0.91	25195
weighted avg	1.00	1.00	1.00	25195

Figure 7.12: Classification report of Decision Tree

classification report of RF Classifier is :

	precision	recall	f1-score	support
0	1.00	1.00	1.00	13424
1	1.00	1.00	1.00	8482
2	1.00	0.96	0.98	171
3	1.00	0.99	1.00	2331
4	1.00	1.00	1.00	196
5	1.00	1.00	1.00	581
6	1.00	0.40	0.57	10
accuracy			1.00	25195
macro avg	1.00	0.91	0.94	25195
weighted avg	1.00	1.00	1.00	25195

Figure 7.13: Classification report of Random Forest

Chapter 8

Conclusion

The increased demand of Wireless ad hoc networks may lead to increase securiy vulnerabilities such as flooding based denial of service attacks. The primary goal of the thesis work is detection and classification of flooding attacks in wireless adhoc networks with high true positive rate as flooding based denial of service attacks is a serious concern in wireless adhoc networks . To do so , supervised machine learning model using NSL KDD dataset is proposed. Updation in NSL KDD dataset is done by including a column which classifies dos attack in to TCP flooding,,UDP Flooding and ICMP Flooding. The classification is done by checking both type of protocol and attack type in NSL KDD dataset. Then machine learning algorithms such as SVM , Naive Baye's, KNN , Decision Tree and Random Forest is trained using 80 % of the dataset and all the above models are evaluated with remaining 20% test dataset.It is found that highest accuracy is achieved with both Decision Tree and Random Forest classifiers.

References

[1] 1.A. Sahi, D. Lai, Y. Li and M. Diykh, "An Efficient DDoS TCP Flood Attack Detection and Prevention System in a Cloud Environment," in IEEE Access, vol. 5, pp. 6036-6048, 2017, doi: 10.1109/ACCESS.2017.2688460.

[2] Ramadhan, G., Kurniawan, Y., & Chang-Soo Kim. (2016). Design of TCP SYN Flood DDoS attack detection using artificial immune systems. 2016 6th International Conference on System Engineering and Technology (ICSET). doi:10.1109/icsengt.2016.7849626.

[3] Lidong Zhou and Z. J. Haas, "Securing ad hoc networks," in IEEE Network, vol. 13, no. 6, pp. 24-30, Nov.-Dec. 1999, doi: 10.1109/65.806983.

[4] Boro D., Basumatary H., Goswami T., & Bhattacharyya D. K. (2016). UDP Flooding Attack Detection Using Information Metric Measure. Proceedings of International Conference on ICT for Sustainable Development, 143–153. doi:10.1007/978-981-10-0129-1-16.

[5] Harshita, Detection and Prevention of ICMP Flood DDOS Attack, International Journal of New Technology and Research (IJNTR) ISSN:2454-4116, Volume-3, Issue-3, March 2017.

[6] [6] N Nishanth, A Mujeeb,"Modeling and Detection of Flooding based Denial of Service Attacks in Wireless Ad Hoc Networks using Uncertain Reasoning,"IEEE Transactions on Cognitive Communications and Networking ,2021.

[7] Ramkumar B and Subbulakshmi , "TCP Syn Flood Attack Detection and Prevention System using Adaptive Thresholding Method,"ITM Web of Conferences,2021.

[8] T. Aditya Sai Srinivas, S.S. Manivannan,"Prevention of hello flood attack in IoT using combination of deep learning with improved rider optimization algorithm,"Computer Communications,Elsevier,2020.

[9] Dan Tang, Jianping Man,Liu Tang, Ye Feng, Qiuwei Yang "WEDMS: An advanced mean shift clustering algorithm for LDoS attacks detection," Elsievier,2020.

[10] Payam Mohammadi,Ali Ghaffari,"Defending Against Flooding Attacks in Mobile Ad-Hoc Networks Based on Statistical Analysis,"Springer,2019.

[11] Giang-Truong Nguyen, Van-Quyet Nguyen, Sinh-Ngoc Nguyen, and Kyungbaek Kim,"Traffic Seasonality aware Threshold Adjustment for Effective Source-side DoS Attack Detection,"KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS,2019.

[12] Sunil Kumar , Kamlesh Dutta,"Direct Trust Based Security Scheme for RREQ Flooding Attack in Mobile Ad Hoc Networks,"International Journal of Electronics, 2017.

[13] S. Revathi, Dr. A. Malathi, A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection , International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 12, December – 2013.

[14] Monowar Hussain Bhuyan, D K Bhattacharyya1 and J K Kalita , Incremental Approaches for Network Anomaly Detection: Existing Solutions and Challenges, International Journal of Communication Networks and Information Security, Vol. 0, No. 0, August 2011.

[15] M.S. Irfan Ahmed, Riyad A.M, Mohamed Jamshad K, Information gain based feature selection for intrusion detection systems, International Journal of Scientific & Engineering Research Vol. 8, Issue 7, July-2017 ISSN 2229-5518.

[16] Luis Alfredo A´ lvarez Almeida, Juan Carlos Martinez Santos, Evaluating Features Selection on NSL-KDD Data-Set to Train a Support Vector Machine-Based Intrusion Detection System, 978-1-7281-1614-3/19/$31.00 ©2019 IEEE.

[17] Hany M. Harb, Afaf A. Zaghrot, Mohamed A. Gomaa and Abeer S. Desuky, Selecting Optimal Subset of Features for Intrusion Detection Systems, Advances in Computational Sciences and Technology ISSN 0973-6107 Volume 4 Number 2 (2011) pp. 179-192 © Research India Publications.

[18] Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf. B, Support vector machines. IEEE Intelligent Systems and Their Applications, 13(4), 18–28. doi:10.1109/5254.708428,1998

[19] Liao, Y., & Vemuri, V. R. , Use of K-Nearest Neighbor classifier for intrusion detection. Computers & Security, 21(5), 439–448. doi:10.1016/s0167-4048(02)00514-x ,2002.

[20] S.L. Ting, W.H. Ip, Albert H.C. Tsang,Is Naive Bayes a Good Classifier for Document Classification?,International Journal of Software Engineering and Its Applications , Vol. 5, No. 3, July, 2011.

[21] Rokach L., & Maimon O. , Decision Trees, Data Mining and Knowledge Discovery Handbook, 165–192. doi:10.1007/0-387-25465-x_9.

[22] Speiser, J. L., Miller, M. E., Tooze, J., & Ip, E. A Comparison of Random Forest Variable Selection Methods for Classification Prediction Modeling, Expert Systems with Applications , doi:10.1016/j.eswa.2019.05.028,2019.

List of Publications

1. Shanifa. E, Dr. Nishanth N, "Detection And Classification of Flooding Attacks in Wireless Adhoc Networks Using Machine Learning", Arabian Journal of Science and Engineering , Manuscript ID : AJSE-D-22-05118, Submission Date : 12-July-2022,Status : Under Review .

Appendix