Jimma University

Jimma Institute of Technology

Faculty of Computing and Informatics

Intrusion Detection System using Hybrid Machine Learning for MANET

A Thesis Submitted in partial fulfillment of the

requirements for The degree of Master of Science in

Computer Networking

by

Taye Endeshaw

December 27, 2021

Jimma, Ethiopia

Declaration

The work contained in this thesis has not been previously submitted to meet requirements for an award at this or any higher education institution.To the best of my knowledge and beliefs,the thesis contains no materials previously published or written by another person except where due reference is made.

Declared by: Taye Endeshaw

Signature ______________________ **Date** 12/27/2021

Confirmed by Principal Advisor: Kebebew Ababu(Ass.Prof)

Signature ______________________ **Date** ______________________

Confirmed by Co-Advisor:Getamesay Haile(MSc)

Signature ______________________ **Date** ______________________

Jimma University

Jimma Institute of Technology

Faculty of Computing and Informatics

Intrusion Detection System using Machine Learning for MANET

By: Taye Endeshaw

Advisor: Kebebew Ababu(Ass.Prof)

Co-advisor:Getamesay Haile(MSc)

Approved by:

Board of Examiners	**Signature**
Henock Mulugeta(PHD)	Henok
Berhanu M.(MSc)	

Abstract

A mobile ad-hoc network (MANET) is an infrastructure-less wireless network and self-organized. During communication MANETs don't use any proper infrastructure so MANET is prone to various sorts of attacks like distributed denial-of-service(DDoS), Bot,Secure Socket Shell(SSH-Bruteforce), and FTP-BruteForce.To provide adequate security against multi-level attacks detection-based schemes should be incorporated additionally to traditionally used prevention techniques because prevention-based techniques cannot prevent the attacks from compromised internal nodes. In this paper, a hybrid machine learning model with a new feature selection method is proposed for better performance of the Intrusion Detection System. In this proposed model, the Intrusion Detection System is built with a combination of supervised and unsupervised machine learning models.The obtained results show that the proposed intrusion detection is effective in detecting the DDoS, Bot, SSH-Bruteforce, and FTP-BruteForce type attacks with a high detection rate. The results show KNN (99.99% accuracy), K Means Clustering(99.99% accuracy), Decision Tree (99.99% accuracy and the hybrid also 99.99% accuracy . Finally, the paper concludes with a variety of future research directions within the design and implementation of intrusion detection systems for MANETs.

Keywords: *Intrusion Detection System, Classification, Machine Learning, Anomaly Detection, Support Vector Machine (SVM), Decision Tree, Naive Bayes, K Means Clustering, K Nearest Neighbors.*

Acknowledgments

Firstly, I would like to thank God for everything. I would also like to show my sincere gratitude to my advisors Kebebew Ababu (Ass.prof), and Mr. Getamesay Haile (MSc) for their initial guidance and continuous support during my thesis progress. Besides my advisor, I would like to thank my friend Mr. Endris mamo for his constant help. Last but not least, I would like to thank my family and my friends: many thanks to my mother Fantanesh, for her never-ending prayers and good wishes which keeps me focused towards my goal, and my younger sister saba, for always standing by me. Finally, thanks to my classmates and friends for their support and love.

Contents

List of Figures

List of Tables

Acronyms

AR	Attribute Ratio
CART	Classification and Regression Tree
CIA	Confidentiality, Integrity, and Availability
DDoS	Distributed Denial of Service
DIDS	Distributed intrusion Detection System
DOS	Denial of Service
DT	Decision Tree
FAR	False Alarm Rate
FTP-BruteForce	File Transfer Protocol-BruteForce
FN	False Negative
FP	False Positive
IDS	Intrusion Detection System
KNN	K-Nearest Neighbors
NIDS	Network-Based Intrusion Detection System
MANET	Mobile Adhoc Network
ML	Machine Learning
SVM	Support Vector Machine
PCA	Principal Component Analysis
PDAs	personal digital assistant
SSH-Bruteforce	SECURE SHELL Bruteforce
VANET	vehicular adhoc network

Chapter 1

Introduction

With the increasing use of handheld wireless devices (tablet computers, cell phones, mobile Internet devices, PDAs, and then on) and up to date advancements in wireless communication technology, Mobile Adhoc Network (MANET) is gaining more importance in commercial, military, public, and personal sectors.The flexibleness and openness of MANET make it attractive for various forms of applications, like military communication, emergency search and rescue operations, disaster recovery, firefighting, policing, communication between moving vehicles (VANET), sensor networks, battlefields, conferences, and so on [1]. A MANET [2] may be a collection of wireless mobile devices (called nodes) that dynamically form a network in environments, an independent mobile node which will communicate with one another via radio transmission like disaster rescue, urgent conference or operation, without the support of a network infrastructure. The topology of the network may change frequently because nodes can join or leave the network at any time. In a MANET, nodes coordinate among themselves to take care of the connections among them. Data transfer from a source node to a non-neighbor destination node is routed through mediate nodes. A node can act like a bunch and a router at an identical time. A network routing protocol during a MANET specifies how nodes within the network communicate with one another. It enables the nodes to get and maintain the routes between any two of them. AN Ad-hoc network may be a temporary, no center, and self-organized wireless network MANET may be a wireless local area network model with no centralized infrastructure like base stations or access points [3, 4].

To supply valid communication between two mobile nodes beyond their transmission mechanism aim MANETs, the intermediate nodes are wont to forward the packets during a multi-hop fashion [5]. The essential assumption of all ad hoc routing protocols is that every mobile node would be reliable, trustworthy, and cooperative within the basic operation of the network [6]. Because of the advantage of flexible, easy form network, and low cost, it's been widely employed in military, disaster relief. The opposite aspect, the Adhoc network has no fixed infrastructure and no fixed self-protection mechanism. However, MANETs are more prone to different type of security attacks because of their inherent characteristics, like dynamic topology, multi-hop environment, error-prone communication media, limited bandwidth, computing power constraints, limited physical security, and frequent routing updates [7]. So, providing secure communication over the MANETs may be a major concern. In MANETs, there's no clearly defined central place, where any security mechanism (to detect, prevent, or get over security attack) is often deployed. Therefore, the safety attacks from both external and internal nodes can compromise the safety and privacy of the network [8]. These attacks can freeze the complete operation of the network and violate the core security principles, i.e., confidentiality, integrity, and availability (CIA) [8, 9, 10, 11]. So, an Ad-hoc network must face more security problems than a traditional network. The flexibleness provided by open broadcasting introduces new security risks. In other words, the Ad-hoc network not only faces the protection problems of the traditional network but also faces new security threats like DoS, Bot, SSH-Bruteforce, and FTP-BruteForce attacks. Although there are many intrusion prevention methods in this area.

Intrusion prevention is dependable 100 percent impossible. Hence, anomaly detection may be used because the next line of defense to issue the first signal. To handle this problem, we require IDS.One of the foremost effective ways to guard the confidentiality, integrity, and availability of data and enterprise systems once an attacker has compromised its defenses is to deploy (IDS)for MANET. Intrusion Detection Systems are defined by the National Institute of Technology (NIST) as ”software or hardware systems that automate the method of monitoring the events occurring in an exceedingly ADP system or network, analyzing them for signs of security problems” [12]. Intrusion Detection is that the art and science of finding attackers that have bypassed preventive defense mechanisms like firewalls, access control, and other protection mechanisms further up or down the stack. More formally, Intrusion Detection is defined by NIST because the ”process of monitoring the events occurring in an exceedingly ADPS or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices” [13]. There are two main styles of Intrusion Detection Systems: Host-based and Network-based. Host-based intrusion detection systems monitor and control data coming from a private workstation using tools and techniques like host-based firewalls, anti-virus/anti-malware agents, data-loss prevention agents, and monitoring call trees. Network-based defenses monitor and control network traffic flows via firewalls, anti-virus, proxies, and network intrusion detection techniques. Network Intrusion Detection Systems (NIDS) are essential security tools that help increase the protection posture of a network. NIDS has become necessary, together with firewalls, anti-virus, access control, and other common defense-in-depth strategies towards helping cyber threat operations teams become tuned in to attacks, security incidents, and potential breaches occurring on their networks. The main target of this research is on advancing HIDSs in MANET, by leveraging recent advances in machine learning.

An intrusion may be a way of behavior that tries to regulate within the integrity, confidentiality, or availability of a resource. There are three models: Anomaly-based IDS, Misuse-based IDS, and specification-based IDS.The primary model is Misuse based IDS which is additionally referred to as signature-based IDS.It's generally preferred by commercial IDS. The system is simply as strong as signatures are previously stored within the database of the IDs and it matches the signature if an attack is found. But if a signature isn't in IDS then can't be detected. The Second technique is Anomaly-based IDS, during which firstly the IDS makes a standard profile of the network and puts this normal profile as a base profile, compare with the monitored network profile. The good thing about this IDS technique is that it may be able to detect attacks without prior knowledge of the attack. The last model is specification-based IDS, it combines the strength of anomaly-based and misuse-based detection techniques providing detection of a known and unknown attack. It can detect a brand-new attack that doesn't follow system specifications. When an intrusion is detected, an appropriate response is triggered per response policy [14]. Hinton et al. [15] proposed the concept of deep learning in 2006. Deep neural networks contain more hidden layers than shallow neural networks. Together with the rise of layers, compared to the standard machine learning method, deep neural network brainpower can do higher accuracy. One among the intense attacks to be considered within the Adhoc network the DDOS, Bot, SSH-Bruteforce, and FTP-BruteForce attack. The DDOS works on by huge amount and collaborated way of attack that affect the asset of the target node. The DDOS attack is launched by sending an enormous number of packets to the target node through coordination of an oversized amount of packet, this huge traffic consumes the bandwidth and doesn't allow the other important packet reached to the victim. In recent years, Mobile Adhoc NETworks (MANETs) have generated great interest among researchers within the development of theoretical and practical concepts, and their implementation under several computing environments. However, MANETs are highly prone to various security attacks because of their inherent characteristics. To provide adequate security against multi-level attacks, the researchers think that detection-based schemes should be incorporated additionally to traditionally used prevention techniques because prevention-based techniques cannot prevent attacks from compromised internal nodes.

An Intrusion detection system is an efficient defense that detects and prevents security attacks at various levels. This paper tries to provide efficient intrusion detection techniques for MANETs by technology layout and detection algorithms.We used local prepared local dataset for MANET for our IDS.First, we preprocess the produced dataset using with dataset preprocessor tool. Train the model and after learning these patterns of malicious and benign by training a Decision Tree, the system can reliably and effectively detect and classify modern attack traffic with a high degree of accuracy, high rate of recall, and an occasional rate of false-positive rate. This is often considered to be a kind of pattern-based detection because the system is trained on known well and known bad patterns and taught to detect these patterns in future, unseen network flows. Second, an alternate deep learning approach, called KNN, K Means Clustering, is employed to detect and classify attack traffic within the case where there is not any labeled malicious training data. This approach is very important because in practice it's going to be difficult to get labeled training data to coach a supervised deep-learning algorithm on malicious and benign traffic. Besides, the character of the adversary is that they're constantly evolving and attempting new attacks, that a pattern-based system might not be effective since new attacks may have patterns that are vastly different than what has been seen historically. This second approach is taken into account by an unsupervised anomaly-based approach because the learning algorithm will put the traffic into clusters, whereby anomalous activity (e.g. outliers) will stand out from the traditional traffic. These detection techniques are evaluated during a privately created dataset with different deep learning algorithms by customizing and improve the detection accuracy by applying new techniques for improving the accuracy of the detection rate and minimize the warning rate. Further, an endeavor has been made to match different intrusion detection techniques. Finally, the paper concludes with a variety of future research directions within the design and implementation of intrusion detection systems for MANETs.

Historically, people have used intrusion detection systems (IDS)Historically, people have used intrusion detection systems (IDS) to guard their networks against adversaries. [13] This involves monitoring the network and detecting network attacks through the utilization of attack signatures; when the traffic matches a known predefined attack pattern, it raises that there has been an attack. Traditional IDS are useful in detecting known attacks but fail within the event of an unseen attack, leaving them hopeless, and also the network vulnerable against zero-day exploits. Additionally, the emergence of recent attack patterns necessitates the update of the signature database containing the definition of attacks. An alternate and also complementary solution is that the application of machine learning approaches because of the basis of attack detection. Machine learning use has rapidly grown over the past decade, with applications in many alternative areas like healthcare, product recommendation, and email spam filtering. [14]A category of machine learning algorithms supervised techniques, allows a system to 'learn' from past events to be ready to predict future outcomes, making it perfect to be used in detecting new network attacks that may go undetected by a conventional IDS. Another method, unsupervised learning, is additionally considered. This approach models normal network behavior and raises alarms when anomalous instances are detected. [15] This project follows a hybrid approach and makes use of labeled and unlabeled network data during training and testing.

1.1 Motivation

With the increasing use of handheld wireless devices (tablet Mobile ad-hoc networks (MANETs), wireless sensor networks (WSNs), and Internet of Things (IoT) are a category of networks that deploy low resource nodes and also the nodes that need rapid deployment. The goal is to develop an intrusion detection system (IDS) capable of handling such constraints. These IoT devices not only help in transmitting and receiving data but also connect various devices to the web. These devices are mobile or stationary looking on the appliance they're speculated to be used for. MANETS and mobile WSNs are the kinds of IoT networks, we are trying to secure during this work. Machine learning and artificial intelligence-based IDSs were studied extensively during the last decade. Various machine algorithms were explored like Neural networks and their newer version, deep learning, support vector machines (SVM), decision trees, k-NN clustering, and Naïve Bayes. However, the accuracy and performance are high. This work is motivated to support the issues: high detection accuracy and performance of the present IDS. . Besides, feature selection and a few best parameter selection methods are provided to reinforce the detection accuracy and performance of the previous works.

1.2 Statement of the problem

An (IDS) which is an important cybersecurity technique, monitors the state of software and hardware running in the network. Despite decades of development, existing IDSs still face challenges in improving detection accuracy, reducing the false alarm rate, and detecting unknown attacks. Every day there are new types of cyber-attacks that are faced by systems and networks of official and nonofficial organizations, e-commerce, and even people around the world. These attempts aim to obtain certain information or destroy the information itself to arrive at stopping the operation of these systems which completely rely on this information. Intrusion detection systems (IDS) are one solution to these problems [16]. To solve the above problems, there are two main types of machine learning: supervised and unsupervised learning. Supervised learning relies on useful information in labeled data. Classification is the most common task in supervised learning (and is also used most frequently in IDS); however, labeling data manually is expensive and time-consuming. Consequently, the lack of sufficient labeled data forms the main bottleneck to supervised learning. In contrast, unsupervised learning extracts valuable feature information from unlabeled data, making it much easier to obtain training data. However, the detection performance of unsupervised learning methods is usually inferior to those of supervised learning methods. Concerning the methods of intrusion detection in a network, NIDS is divided into two classes as Signature-based NIDS (SNIDS) and Anomaly-based NIDS (ANIDS). A Signature-based NIDS has pre-installed rules for the attacker. A pattern matching is performed while detecting the attack. The Anomaly-based NIDS detects the attack by observing the traffic shape. Its objective is to measure the deviation from normal traffic. A Signature-based NIDS is very much familiar and promising in the detection of any known attack and it has high accuracy concerning less false alarm rate. However, the performance of Signature-based NIDS starts to decrease and becomes challenging whenever any unknown attack or when there is anomaly traffic in the network. The signatures are pre-installed into the IDS system which has to be matched to detect any attack. Although Anomaly Detection NIDS (ADNIDS) produces extremely more false-positive alarm, it's hypothetically possible to identify any unknown attack where this idea is widely accepted among the research community [17].The misuse attack detection technique achieves maximum accuracy and minimum false alarm rate, but it cannot detect unknown attacks.

In today's world, Network and System Security are of paramount importance in the digital communication environment. On par with the developments in technology, many threats have emerged for information security which has worse effects when it comes to sensitive transactions. Nowadays, intruders can easily break the walls of the network and can cause many kinds of breaches such as the crash of the networks, Denial Of Service, injecting Malware and so on. In order to avoid those breaches, it is badly needed for a security administrator to detect the intruder and prevent him from entering into the network. In daily life, new threats and associated solutions are emerging together.

1.3 Objectives

1.3.1 Objective

The general objective of this study is develop an intrusion detection system for MANET using hybrid machine learning algorithm.

1.3.2 Objective Specific Objectives

To accomplish the above-stated general objective, the subsequent specific objectives are required.

- To design algorithms and techniques that are used for ID using machine learning,
- Explore the most recent dataset which will be used for model training and testing.
- Preprocess the dataset and identify the simplest features for model training and testing.
- Design an algorithm for detecting and a classify attack traffic
- Explore different machine learning classification algorithms and train them on the prepared training data.
- Select the most effective classification model supported the evaluation results.
- Write and implement an algorithm for the proposed solution
- Test and analyze the results obtained.
- Finally, conclude the results and specifies future work.

1.4 Scope (Limitations and Delimitation)

The scope of the study will only be considering DDos, Bot, SSH-Bruteforce, and FTP-BruteForce attacks the proposed system will detect a set of attacks, such as Denial of Service (DoS), Brute force, Bot, and DDOS or the network flow is of a benign class.
Limitation One of the limitations of this research work is that only one dataset (which contains network traffic data and is prepared in local with a small number of devices) is used to build models that can classify the input data into normal or attack. Also, one feature selection method is applied to the dataset to select relevant features for model building and testing.

- The Model don't prevent any attack because the IDS is for Detecting the attack.
- Not consider other type of attack except (DoS), Brute force, Bot, and DDOS
- The dataset set is prepared locally within small numbers of device

1.5 Methods

The main focus of the thesis is to improve detection accuracy using a machine learning algorithm. The methodology will be following the following list of concepts.

- **Literature Review:** A detailed review will be done on journal articles, conference proceedings, books, dissertations, and the Web to have a deep understanding of the previous works in the study area. Different methods and techniques in network intrusion detection will be examined. From the review of these works, appropriate techniques and methods will be selected.
- **Data Collection:** For implementing the proposed model a dataset prepared that has the latest attack and normal data with less redundant records from Wireshark will be collected through analyzing different network flow.
- **Tools and Development Environments:** Different free and open-source tools will be used during model development. Python programming language, jupyter notebook, TensorFlow, Keras, and another deep learning library will be used to implement the system. The entire implementation will be done in the Anaconda environment using jupyter and collab.
- **Testing and Evaluation:** Every trained model will be tested on new data that is unseen during the training to determine how well it classifies the samples correctly. The performance of the models will be evaluated using classification model evaluation metrics.
- Produce private dataset is used during training and testing
- A lightweight algorithm is applied for classifying and training (on the dataset)

1.6 Application of Results

There are several technologies and methods to safeguard businesses and organizations' networks from malicious activities. A network IDS in MANET is one among these mechanisms which protect a system from network-based threats by reading all packets and searches for any suspicious patterns and when threats are discovered it notifies the administrator or excluding the source from accessing the network. The results of this study can help organizations to guard their network against external and internal attacks.

1.7 Organization of Rest of Thesis

The remaining chapters are organized as follows. Chapter 2, presents literature about the nature of deep learning and dataset. Chapter 3, introduces related works which are carried out for improvement of intrusion detection using different deep learning algorithm. Chapter 4, presents the detail of the proposed. Chapter 5, provides an extensive simulation study and evaluation of the proposed. Finally, the conclusions of the research and recommendations of future works are presented in Chapter 6.

Chapter 2

Literature Review

Throughout this thesis, we check with different IDS, machine learning models, and various terminology associated with the way to evaluate them. Here, we briefly provide a summary of machine learning, how each of the various algorithms works, and the way to calculate the relevant metrics needed. We also present the relevant networking theory and discuss the assorted styles of network attacks that the model is going to be ready to detect.

2.1 Overview Mobile Ad-hoc Network

Since their emergence within the 1970s, wireless networks have become increasingly popular within the computing industry.This is often particularly true within the past decades which has seen wireless networks being adapted to enable mobility. There are currently two variations of mobile wireless networks which are defined by IEEE 802.11 standards. The primary is thought of as infrastructure networks, i.e., those networks with fixed and wired gateways. The bridges for these networks are called base stations. A mobile unit within these networks connects to and communicates with, the closest base station that's within its communication radius. Because the mobile travels, out of ranging of 1 base station and into the range of another, a handoff occurs from the old base station to the new, and therefore the mobile can continue communication seamlessly throughout the network. Typical applications of this kind of network include wireless local area networks (WLANs) [18].

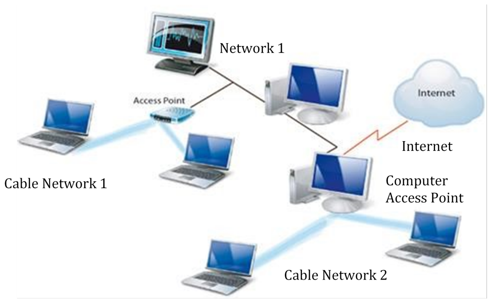


Figure 2.1: An Infrastructure Mobile Network
[18]

The second variety of mobile wireless networks is that the infrastructure-less mobile network commonly referred to as Ad-hoc network or Mobile Ad-hoc Network (MANET). Infrastructureless networks don't have any fixed routers; All nodes are capable of movement and may be connected dynamically in an arbitrary manner. Nodes of those networks function as routers that discover and maintain routes to other nodes within the network.

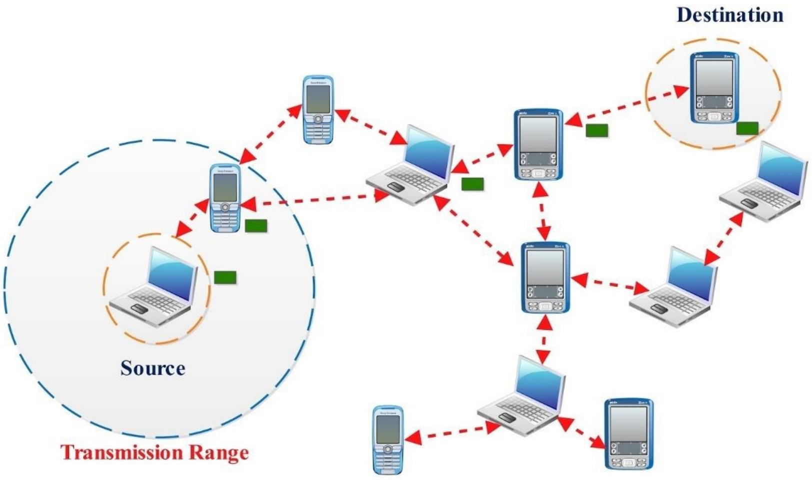


Figure 2.2: Mobile Ad-hoc Network
[18]

The whole life-cycle of mobile Ad-hoc networks might be categorized into the primary, secondary, and third-generation Ad-hoc networks systems. Present Ad-hoc networks systems are considered the third generation. The primary generation goes back to 1972. At that point, they were called PRNET (Packet Radio Networks) research for military purposes within the 1970s, which evolved into the Survivable Adaptive Radio Networks (SURAN) program within the early 1980s. In conjunction with ALOHA

(Areal Locations of Hazardous Atmospheres) and CSMA (Carrier Sense Medium Access), approaches for medium access control and a form of distance-vector routing PR-NET, were used on an effort basis to supply different networking capabilities in a very combat environmen [19]. The second generation of Ad-hoc networks emerged in the 1980s when the Ad-hoc network systems were further enhanced and implemented as an element of the SURAN program. This provided a packet-switched network to the mobile battlefield in an environment without infrastructure. This program proved to be beneficial in improving the radios' performance by making them smaller, cheaper, and resilient to electronic attacks.Within the 1990s, the concept of business Ad-hoc networks arrived with notebook computers and other viable communications equipment. The thought of mobile Ad-hoc networks has been under development from the 1970s and 1980s within the framework of Mobile Packet Radio Technology (PRNET-1973) and Survivable Adaptive Networks (SURAN-1983). Within the middle of the 1990s, with the definition of standards, commercial radio technologies have begun to seem, and therefore the wireless research community identified in Ad-hoc networks a challenging evolution of wireless networks [20]. Today's emerging standards and technologies for constructing a mobile Ad-hoc network are IEEE 802.11, Bluetooth, and ETSI Hiperlan/2. The deployment of mobile Ad-hoc networks opens a large range of potential utilization from military to miscellaneous commercial, private and industrial scenarios, [20].

A mobile ad-hoc network is a self-configuring infrastructure-less network of mobile devices connected by wireless links. Ad-hoc is Latin and means "for this purpose". Each device in a MANET is free to move independently in any direction, and will therefore change its links to other devices frequently. Each must forward traffic unrelated to its use, and therefore be a router. The primary challenge in building a MANET is equipping each device to continuously maintain the information required to properly route traffic. Such networks may operate by themselves or may be connected to the larger Internet. The growth of laptops and 802.11/Wi-Fi wireless networking have made MANETs a popular research topic since the mid1990s. Different protocols are then evaluated based on measures such as the packet drop rate, the overhead introduced by the routing protocol, end-to-end packet delays, network throughput, etc. The mobile ad hoc network has the following typical features

- Unreliability of wireless links between nodes.
- Constantlychanging topology. Per the IEEE802.11

Specification, a mobile Ad-hoc network is a network composed solely of nodes within mutual communication range of each other via the wireless medium. This definition implies that an Ad-hoc network is a complete one, in the sense that all nodes are in the neighbor of each other to form an arbitrary topology without a fixed infrastructure. The nodes move randomly and organize themselves arbitrarily in such a way that the wireless network topology may be subject to frequent changes. Such a network may operate in a standalone fashion or may be connected to the larger Internet. The nature of Ad-hoc networks makes them suitable for emergencies such as natural or human-induced disasters, military operations or emergency medical conditions, community networking, and interaction among meeting attendees or students during a lecture. In a multi-hop wireless Ad-hoc network, mobile nodes cooperate to form a network without using any infrastructure such as access points or base stations. Instead, the mobile nodes forward packets for each other, allowing communication among nodes outside the wireless transmission range. The nodes' mobility and the fundamentally limited capacity of the wireless medium, together with wireless transmission effects such as attenuation, multipath propagation, and interference, combine to create significant challenges on Ad-hoc routing protocols.

Figure 2.3: An Ad-hoc Network Example among Laptops

2.2 Mobile Ad-hoc Network Technologies

MANET devices are generally everyday devices such as Bluetooth and ZigBee, with application in Ultra Wide Band (UWB), WiFi and WiMAX enabled Laptop/Palmtop Computers, 3G/4G cellular phones as well as GSM Mobile phones and other electronic devices. The technology of these devices and how they compare with mobile ad hoc networks in terms of mode of operation, throughput, multiplexing, frequency, application and coverage are further discussed.

- A). Ultra-Wide Band is a communication method used in wireless networking to achieve high bandwidth connections with low power utilization. Originally designed for commercial radar systems, UWB technology has potential applications in consumer electronics and wireless personal area networks (PAN). UWB radios send short signal pulses over a broad spectrum, allowing UWB to commonly support high wireless data rates of 480 Mbps up to 1.6 Gbps at 5GHz, covering a distance of about 100m. At longer distances, UWB data rates drop considerably. UWB technology is used in consumer networks such as: wireless USB, wireless high-definition video, next-generation Bluetooth, peer-to-peer connections, etc .

- B). Wireless Fidelity (WiFi) broadband network is used to describe Wireless Local Area Network (WLAN) products that are based on the IEEE 802.11 (a/b/g/n) standards. The multiplexing mode uses both single carrier direct-sequence spread spectrum (DSSS) radio technology and multi-carrier OFDM (Orthogonal Frequency Division Multiplexing) radio technology. It can achieve data rate of up to 54 Mbps at frequencies of 2.4 and 5 GHz, covering 100 to 150 meters. The 802.11n variant can cover 300m and is capable of delivering data rate of up to 600 Mbps [19].

- C). Worldwide Interoperability for Microwave Access (WiMAX) is a broadband wireless network based on the IEEE 802.16 standard that is primarily tailored towards Wireless Metropolitan Area Network (WMAN). While the 802.16 d/g variants are used for fixed networks, the 802.16 e/n versions are for mobile networks. WiMAX operates at frequencies of 2 to 11 GHz, covering theoretical distance of up to 50 km (ideally 10 to 15 km). It has high data rate of up to 75 Mbps and uses Orthogonal Frequency Division Multiplexing (OFDM), offering fixed/mobile multimedia services.

- D). THIRD/FOURTH GENERATION CELLULAR NETWORKS (3G/4G) are broadband wireless mobile networks that has evolved from the 1st to the 2nd and 3rd generation networks. The still evolving 4th generation network is expected to be deployed in 2011. Only the 3rd and 4th generation networks are considered here. The IMT-2000, 3rd Generation (3G) mobile network is a broadband system with data capabilities of up to 2 Mbps, delivering multimedia (voice, data, audio entertainment, images, video clips, etc) applications at frequencies of 900, 1800, 2100 MHz. Generally, 3G technology can be split into: High Speed Downlink Packet Access (HSDPA), Code Division Multiple Access (CDMA2000), 1 x

EV-DV (Evolution - - Data and Voice) and WCDMA (Wideband-CDMA). The specification is mainly for worldwide area networks (WWAN), covering up to 5 km. The Fourth Generation (4G), IMT-Advanced (was previously called “Systems beyond IMT – 2000”) is a new generation of wireless broadband network intended to complement and replace the 3G systems, in the near future. Accessing information anywhere, anytime, with a seamless connection to a wide range of information and services, and receiving a large volume of information, data, pictures, video, and so on, are the key features of the 4G infrastructures. The future 4G infrastructures will consist of data rates of up to 200 Mbps, covering 10 km or more at frequencies of 2 to 8 GHz.

2.3 Types of MANET

2.3.1 Vehicular Ad-hoc Networks (VANETs)

: A Vehicular Ad-Hoc Network or VANET could be a technology that uses moving cars as nodes during a network to form a mobile network. VANET turns every participating car into a wireless router or node, allowing cars approximately 100 to 300 meters of every other to attach and, in turn, create a network with a good range. As cars fall out of the signal range and drop out of the network, other cars can take part, connecting vehicles to at least one another so that a mobile Internet is formed.It’s estimated that the primary systems that may integrate this technology are police and fire vehicles to speak with one another for safety purposes.

2.3.2 Internet Based Mobile Ad-hoc Networks (iMANET)

Internet-Based Mobile Ad-hoc Networks are ad-hoc networks that link mobile nodes and stuck Internet-gateway nodes. In such a style of network normal impromptu routing algorithms don’t apply directly. Wireless networks can generally be classified as wireless fixed networks, and wireless, or mobile ad-hoc networks. MANETs (mobile ad-hoc networks) are supported the concept of creating a network without taking any support from a centralized structure. Naturally, these forms of networks are suitable for situations where either no fixed infrastructure exists, or to deploy one isn’t possible.

2.3.3 Intelligent vehicular ad-hoc networks (InVANETs)

InVANET, or Intelligent Vehicular Ad-Hoc Networking, defines an intelligent way of using Vehicular Networking. InVANET integrates on multiple ad-hoc networking technologies like WiFi IEEE 802.11, WAVE IEEE 1609, WiMAX IEEE 802.16, Bluetooth, IRA, ZigBee for straightforward, accurate, effective, and easy communication between vehicles on dynamic mobility. Effective measures like media communication between vehicles are often enabled moreover methods to trace the automotive vehicles are additionally preferred. InVANET helps in defining safety measures in vehicles, streaming communication between vehicles, infotainment, and telematics. Vehicular Ad-hoc Networks are expected to implement a range of wireless technologies like Dedicated Short Range Communications (DSRC) which may be a sort of WiFi. Other candidate wireless technologies are Cellular, Satellite, and WiMAX. Vehicular Ad-hoc Networks are viewed as a component of the Intelligent Transportation Systems (ITS).

2.4 CHARACTERISTICS OF MANET

The characteristics of MANET are following:

- Autonomous terminal: Each node in MANET is autonomous and acts both, as router and host.
- Distributed: MANET is distributed in its operation and functionalities, like routing, host configuration, and security.
- Multi-hop routing: If the source and destination of a message are out of the range of 1 node, a multihop routing is formed.
- Dynamic network topology: Nodes are mobile and might join or leave the network at any time; therefore, the topology is dynamic.
- Fluctuating link bandwidth: the soundness, capacity, and reliability of a wireless link are always inferior to wired links.
- Thin terminal: The mobile nodes are often lightweight, with less powerful CPU, memory, and power.
- Spontaneous and mobile: Minimum intervention is required in the configuration of the network. The routing protocol should be an adapted one that enables users to speak within the network. It should also support security. Some existing security technologies for wired networks, like encryption, may be utilized in MANET. However, thanks to the mobile and circumstantial nature of MANET, the applications of MANET are limited. Other technologies, like firewall, don't apply to MANET, due to the dearth of a centralized authority. Same because the wired

network, MANET faces safety threats like passive eavesdropping, spoofing, and denial of service. At the identical time, thanks to its impromptu nature, it suffers from more security threats. Threats to MANET is classified into two groups:

- Vulnerabilities accentuated by the impromptu nature: The topology of MANET is principally determined by geographical locations and by the radio range of the nodes. Therefore, it doesn't have a clearly defined physical boundary. In a wired network, a centralized firewall can implement access control. However, in MANET, access-control can't be other attacks, like denial of service (DOS) still threat MANET, even worse than for wired network, since the routing and auto-configuration framework of MANET is more liable to such attack.

- Vulnerabilities specific to the impromptu nature: The routing and auto-configuration mechanism of MANET introduces the opportunity for more attacks because, in both mechanisms, all nodes have full trust between one another.

2.5 MOBILE AD-HOC NETWORK APPLICATIONS

With the rise of portable devices also as progress in wireless communication, Ad-hoc networking is gaining importance with the increasing number of widespread applications. Adhoc networking is often applied anywhere where there's little or no communication infrastructure or the prevailing infrastructure is pricey or inconvenient to use. Ad-hoc networking allows the devices to keep up connections to the network, in addition, to easily adding and removing devices to and from the network. The set of applications for MANETs is diverse, starting from large-scale, mobile, highly dynamic networks, to small, static networks that are constrained by power sources. Besides the legacy applications that move from the traditional infrastructure environment into the Ad-hoc context, an excellent deal of recent services can and can be generated for the new environment. Typical applications include:

1) Military battlefield:

Military equipment now routinely contains some type of computer equipment. Ad-hoc networking would allow the military to require advantage of commonplace network technology to keep up an information network between the soldiers, vehicles, and military information headquarters.The fundamental techniques of Ad-hoc networks came from this field [21].

2) Commercial sector:

Ad-hoc is employed in emergency/rescue operations for disaster relief efforts, e.g. In fire, flood, or earthquake. Emergency rescue operations must happen where nonexisting or damaged communications infrastructure and rapid deployment of a communication network is required. Information is relayed from one rescue team member to a different over a tiny low handheld. Other commercial scenarios include e.g. ship-to-ship Ad-hoc mobile communication, enforcement, etc.

3) Local-level:

Ad-hoc networks can autonomously link a second and temporary multimedia network using notebook computers or palmtop computers to spread and share information among participants at e.g. conference or classroom. Another appropriate local level application may be in-home networks where devices can communicate to exchange information. Similarly, in other civilian environments like a taxicab, construction, boat, and little aircraft, mobile Ad-hoc communications will have many applications.

4) Personal Area Network (PAN):

Short-range MANET can simplify the intercommunication between various mobile devices (such as a PDA, a laptop, and a cellular phone).

2.6 Challenge of MANET

Challenge of MANET **Limited Bandwidth** The wireless networks have limited bandwidth as compared to the wired networks. Wireless link has the lower capacity as compared to infrastructure networks. The effect of fading, multiple accesses, interference conditions is incredibly low in ADHOC networks as compared to the maximum radio transmission rate. **Dynamic topology** Due to dynamic topology, the nodes have less trust between them. We some settlements are found between the nodes then it also makes the trust level questionable. **High Routing** In ADHOC networks because of dynamic topology, some nodes change their position which affects the routing table. **The problem of Hidden terminal** The Collision of the packets is held because of the transmission of packets by that node which isn't within the transmission mechanism range of the sender-side but is in the range of the receiver side. **Transmission error and packet loss** By increasing collisions, hidden terminals, interference, uni-directional links, and by the mobility of nodes' frequent path breaks the next packet loss has been faced by ADHOC networks. **Mobility** Due to the dynamic behavior and changes within the configuration by the movement of the nodes.ADHOC networks face path breaks and it also changes within the route frequently. **Security threats** New security challenges brought by Adhoc networks thanks to their wireless nature. In Adhoc networks or wireless networks, the trust management between the nodes results in various security attacks.

2.7 SECURITY ATTACKS IN MANET

Understanding possible sort of attacks is often the primary step towards developing good security solutions.It's important for the secure transmission of knowledge. There is some basic class of attacks in MANET that may cause slow network performance, delay of messages, uncontrolled traffic, etc. Attacks are often categories into four types.

- **Internal Attack:** In an inside attack, the malicious node within the same network gains unauthorized access and impersonates as a real. It also participates in other network activities and analyses the traffic between other nodes.
- **External Attack:** In an external attack, the malicious node from another network gains unauthorized access and causes congestion sends false routing information or causes unavailability of services [22].

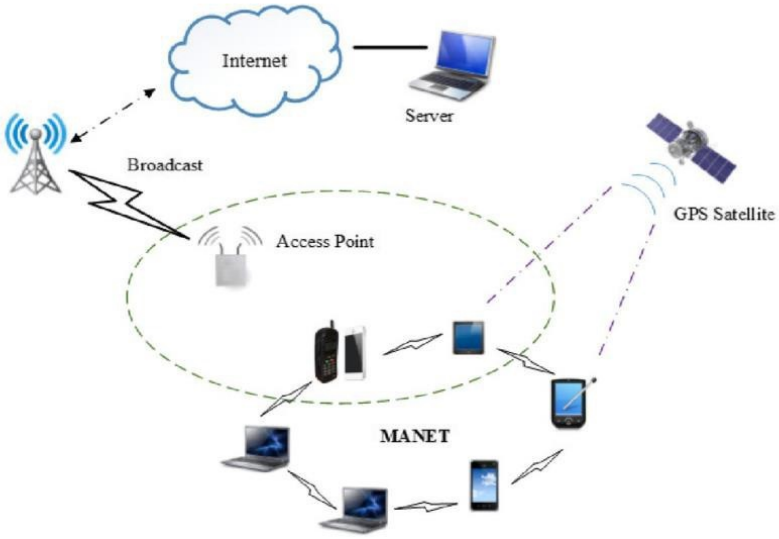


Figure 2.4: MANET Network Topology

- **Active Attack:** In an active attack, the malicious node from any network takes control of a communication between two entities and masquerades as one of them jamming, which causes channel unavailability by overusing it [21].
- **Passive Attack:** In a passive attack, the malicious node from any network, the attacker eavesdrops on packets and analyses them to pick up required information.

MANET attacks may be approximately classified into two main categories, namely passive attacks, and active attacks, betting on the means of attack [23, 24]. A passive attack gets data exchanged within the network without interrupting the communication operation, whereas a lively attack involves interruption of knowledge, Modification, or manufacturing, thus disrupting the MANET's normal functionality. The attacks may be classified into two categories in line with the domain of the attacks, namely external attacks, and internal attacks. External attacks are performed by nodes that don't seem to be a part of the network's domain. Internal attacks are from compromised nodes that are a part of the network after all. Internal attacks are more severe than external attacks.

2.7.1 Denial of Service Attacks (DoS)

Denial of Service Attacks (DoS) DoS attack could be a style of attack in which the hacker makes a computer or memory resource too busy or too full to serve legitimate networking requests, thus denying users access to a tool like an apache, Smurf, Neptune, death ping, back, mail bomb, UDP storm, etc are all DoS attacks [25].

2.7.2 Remote to User Attacks (R2L)

A remote user attack is an attack during which a user sends packets over the net to a machine that he/she has no access to show the vulnerabilities of the devices and utilize the rights that a neighborhood user would wear the pc, like xlock, client, xnsnoop, phf, send-mail dictionary, etc. [26].

2.7.3 User to Root Attacks (U2R)

These attacks are exploitations within which the hacker begins with a traditional user account on the system and tries to abuse system vulnerabilities to get superuser privileges like perl, xterm [27].

2.7.4 Probing

Probing is an attack during which the hacker scans a machine or networking device to spot weaknesses or vulnerabilities which will be exploited later to compromise the system. Usually, this method is employed in information mining like saint, portsweep, mscan, Nmap, etc [28].

2.8 An intrusion detection system (IDS)

An intrusion-detection system (IDS) is defined because of the tools, methods, and resources to assist in identify, assess, and report unauthorized or unapproved network activity. The intrusion detection technique is essentially independent of the architecture or environment. In other words, anomaly and misuse detection are often utilized in a very wireless environment even as they're in an exceedingly wired network. The difference in implementation is especially on what audit data to require as input to the algorithm. However, most IDS in MANET utilize anomaly detection thanks to the special nature of MANET. Intrusion detection is usually a part of an overall protection system that's installed around a system or device.It's not a stand-alone protection measure.Reckoning on the detection techniques used, IDS is classified into three main categories as follows:

- Signature or misuse-based IDS
- Anomaly-based IDS
- Specification-based IDS

The signature-based IDS uses pre-known attack scenarios and compares them with incoming packets traffic. There are several approaches within the signature detection, which differ in representation and matching algorithm employed to detect the intrusion patterns. The detection approaches, like expert system, pattern recognition, colored Petri nets, and state transition analysis are grouped on the misuse. The anomaly-based IDS attempts to detect activities that differ from the traditionally expected system behavior. This detection has several techniques, i.e. Statistics, neural networks, and other techniques like immunology, data processing, and Chi-square test utilization. The specification-based IDS are hybrid of both the signature and therefore the anomaly-based IDS. It monitors the present behavior of systems per specifications that describe desired functionality for security-critical entities. A mismatch between current behavior and therefore the specifications are reported as an attack.

2.8.1 Intrusion detection system IN MANET

An intrusion detection system is an alarm mechanism for a system. It detects the protection compromises that happened to an ADPS then issues an alarm message to an entity, like a site security officer so the entity can take some actions against the intrusion. An ID contains an audit data collection agent, which keeps track of the activities within the system, a detector that analyzes the audit data, and issues an output report back to the positioning security officer.Within the discussion of IDS in MANET, two concepts must be distinguished: intrusion detection techniques and intrusion detection architecture. Intrusion detection techniques discuss the concepts like an anomaly and misuse detection. They mainly solve the issues like, how an ID detects an intrusion with a specific algorithm, given some audit data as a computer file. The intrusion detection architecture deals with problems during a larger scope. It is essential for wireless ad-hoc networks to defend against malicious behavior, to secure the routing of MANET, and to address the restrictions of cryptographic systems IDSs, which are successfully employed in Mobile unintentional networks to detect attacks, can provide an appropriate second line of defense to spot malicious traffic and misbehaving nodes in wireless environments. Intrusion detection architecture has to employ certain intrusion detection techniques as a module. But it also contains many other modules, like a module on how the nodes in an exceeding network can collaborate in higher cognitive processes regarding intrusion detection. during a wired network, a node can usually make intrusion detection decisions supported by the info collected locally. Therefore, an intrusion detection technique can meet the requirement for intrusion detection once it's deployed on a node.In an exceedingly wireless network, however, it's very difficult for a node to create the choice just supported data collected locally. Nodes must collaborate or exchange data a minimum in making an intrusion detection decision. Therefore, an architecture to define the roles of various nodes and therefore the way they impart is extremely important in wireless IDS.

Distributed and Cooperative Intrusion Detection System

In this architecture, each node has an IDS agent that locally detects intrusions and collaborates with neighboring nodes for global detection whenever there's indeterminate evidence and a broader search is required. When the intrusion is captured, either an area response (e.g. alerting the local user) or a worldwide response is also issued by an IDS agent. Each node is involved within the method and response of intrusion detection as having an IDS agent running on that. An IDS agent is chargeable for detecting and collecting local information and data to spot any attack if an attack occurs within the network, in addition to taking an independent response.

Hierarchical Intrusion Detection System

Hierarchical IDS system Expand the distributed and cooperative IDS system functions and are implemented for multi-layer network infrastructures where the network is split into various small networks referred to as clusters. Usually, each cluster head has more functionality than other cluster members, like transmitting data packets to other clusters.We will therefore say that these cluster heads add how as central point's the same as wired network control devices like routers, switches, or gateways. The multi-layering concept applies to intrusion detection systems where there's a proposal for hierarchical IDS. Each IDS agent runs on a selected member node and is to blame for its node, i.e. monitoring and picking intrusions detected locally. A cluster head is answerable for their node locally furthermore as globally for his or her cluster, like monitoring network traffic and announcing a worldwide response when detecting network intrusion.

2.8.2 Intrusion Detection Models the MANET

An intrusion Detection System may be a security measure that may be installed on a network to forestall attacks from happening. The IDS allows network administrators to detect individuals trying to compromise the system in order that they retrieve information from it. There are various activities that the administrators can detect to spot them. This includes security policies violation.The IDS works best because it's designed in a manner that permits it to detect the vulnerabilities on the system within which it's installed.For instance, it can work supported previous attacks that affected the network and work backward to eliminate the possibilities of another similar attack. The IDS can detect attacks using various methods.For instance, it may be done through signature-based detection. These patterns are studied and compared to previous events or attacks to identifies new threats. As a result, the system administrators are often ready to identify new threats and different kinds of threats that the network is at risk of. An IDS is created of three basic components that include Network Intrusion Detection System (NIDS), Network Node Intrusion Detection System (NIDS), and Host Intrusion Detection System. Each of those components plays an awfully vital role in securing networks. The Network Intrusion Detection System works by first analyzing the traffic on the network. It then identifies possible threats with those attacks that are already registered in its library. The Host Intrusion Detection System, on the opposite hand, captures the image of the complete system file set and so compares it with the previous picture. If there's a difference in any respect, then it alerts the system's administrators who then come and stops the possible attack.There's also a Cloud Intrusion Detection system that's used for public environments. There are two general sorts of Intrusion Detection Systems.They're host-based intrusion systems and network-based intrusion systems. Each of these two systems has sensors that are aligned to the sort of intrusion system. There are sensors on a network-based IDS that monitor streams of traffic. The network-based IDS have various advantages and downsides.The subsequent are the samples of the benefits that are aligned to the present reasonably network; there's a lower cost of ownership when organizations and governments have network-based IDS.This is often because the traffic on the network is monitored as an entire.This means that the tendency of loading software on each host on the network is omitted.It's easier to deploy a network-based network.One of the benefits aligned to the current is that the installation of that network doesn't affect the present infrastructure. Detecting a network-based attack on this sort of network is simpler.This is often because the network-based IDS have sensors that check all the packets and identify any threat that will exist on the network.Employing a network-based IDS is additionally advantageous because it's real-time detection and quick responses to any reasonable attack which may face the network. The host-based intrusion detection system has various advantages that are tied to that [22].The quiet system may be ready to give feedback to the success or failure of the attacks thereon network.This can be because the Host-based intrusion detection system contains logs of all activities that have taken place.This sort of IDS also can be ready to monitor the activities that affect a given network where it's installed. Another advantage is that the host-based IDS is that it may be able to detect the attacks that are caused on the Network-based IDS and gone unnoticed. Network-based IDS sensors cannot, as an example, detect when an unauthorized user makes changes on a network. The sensors supported by the host are more superior to the opposite quiet sensors. These forms of IDS even have a near real-time detection and response to attacks and threats to attacks. this is often vital for administrators because they will be ready to handle attacks way before they're launched [21]. The host-based Intrusion detection sensors are installed inside the host servers or machines that play host to them.This suggests that there's no additional hardware that's required to put in this sort of IDS. When a comparison is formed between the host and network-based sensors, the host-based sensors are way cheaper. this implies that the price of entry to the current quiet IDS is cheaper. From the sound of the benefits of those two types of IDS, we are prompt to assume that each institution out there needs either of those IDS. However, each of those IDS comes with disadvantages that may limit its performance, and it's important to grasp each of them [23].The IDS technology is advancing daily, and thus organizations that acquire either of them should make sure that their system is up to this point so it may be ready to handle even the foremost recent styles of threats which will be launched on a given network. Having an IDS system on a network isn't the answer to preventing every kind of attack. The success of those systems depends heavily on the way the IDS sensors are deployed on a given network. Therefore system administrators should make sure that the deployment procedure is finished and achieved within the manner that they're presupposed to occur. The IDS technology is also a reactive activity, not a proactive one. This simply implies that the IDS technology heavily relies on previous attack patterns. The technology cannot work independently. However, IDS technology is extremely important for any organization that seeks to secure itself right from the network level [24] plenty of data is often secured through the method, and this can be what each organization seeks to realize at the tip of the day.It's important to place in better identification and robust authentication processes. Implementation of MANET Anomaly-based Intrusion Detection Model An anomaly-based intrusion detection system, monitors and alerts intrusions and misuse by observing activity that falls out of normal system operation.This is often in contrast to signature-based systems, which may only detect attacks that a signature has previously been created. Anomaly-based intrusion detection has the aptitude to spot unknown intrusions in addition to zero-day assaults. The strength of this emerges from the aptitude of ABIDS to model the standard operation disposition of a network and further identify deviations from the baseline. ABIDS may be specifically configured to suit a selected network thus making it challenging for a previously successful attack in one network to be replicated during a unique setting. Anomaly Based intrusion detection systems can implement different methodologies in either; artificial intelligent knowledge-based detection, statistical anomaly detection, data mining-based detection, or a machine learning-based detection algorithm.

1. Statistical ABIDS Model (SABIDS) A Statistical intrusion detection model could be a common technique for identifying attacks and intrusions in a very network. Statistical-based anomaly detection techniques employ statistical values and statistical assessments to conclude whether the observed performance departs considerably from the expected norm These statistical anomaly detection systems rely on supported a quasi-stationary activity, which is rare for many of the information processed by anomaly detection methods.

2. Operational Model or Threshold Metric Model Operational Model is also spoken because the Threshold Metric model is founded on the hypothesis that abnormal activity will be recognized by comparing a stream of activity against a predefined limit.Supported observed activity over a phase of your time, an alarm may be raised. The methodology is applied; particular statistics are commonly associated with network intrusions. An Adaptive Threshold Algorithm is combined during this case scenario as a toddler model. The Adaptive threshold algorithm may be a straightforward methodology that evaluates whether the quantity of knowledge over a given interval meets a group threshold.

3. Markovian Process Model or Marker Model The Markovian/Marker methodology is co-joined along with data values to conclude on the regularity of a particular occurrence, supported prior events. This model symbolizes every captured data as an isolated case and exploits a state transition method to determine whether the observed occurrence is often supported by prior events. This model is principally advantageous if the sequence of occurrences is predominantly significant. This model has predicated on two major procedures the Markov chains

and also the Markov models.

4. Statistical Moments or Mean and variance Model The Statistical Moments or Mean and variance Model implements statistical prediction and evaluation from the norm supported current values measured against a ramification of possible scenarios. The statistical moments sub-model evaluates and concludes that a selected occurrence that goes beyond a group alarm value is anomalous. This method comprehends device instances without prior knowledge of the device's traffic behaviors.This technique is incredibly flexible and has the power to see anomalous activity without prior briefing or configurations. It is, however, a really complex model to implement and build.

5. Multivariate Model This Multivariate Model is employed to watch and detect intrusions supported by two or more behavioral occurrences. The Multivariate Model thrives in occurrences of two or more behavioral occurrences that allow identification of possible irregularities in cases of complex conditions with multiple constraints. This Multivariate Model, when augmented with statistical methods like chi-square, produces improved results with low warning experienced in addition to a high detection rate. During this methodology, anomalous activity is identified fast. This model, however, is computer-intensive with large volumes of statistical processes required to gauge events accurately.

6. Statistic Model The TimeSeries Model identifies intrusions through a process of evaluating the sequence and time taken to perform various tasks during a networked system.A happening is marked as normal if the metrics are under the edge, while an incident is marked as anomalous if the metrics are observed to be above the edge. The statistic methodology is flexible since it adapts and modifies itself supported user actions. The alarm is ready off by activities that exhibit substantial departure from the regular disposition.

7. Data processing Based Approach the information Mining Based Approach is beneficial when wont to identify external malicious traffic coming into the network. It, however, maybe a weak mechanism for identifying internal cases of attacks. This data processing method is crucial in excluding regular occurrences from raising the alarm, thus enabling security admins to only dedicate time to managing actual network attacks.The info mining approach has the potential of identifying false alarms additionally as irregular signatures, thus enabling only the particular anomalous activity being identified and acted upon. This data processing-based

method uses two major techniques; clustering traffic into groupings and identifying normal occurrences while facilitating the invention of attacks.

8. Association Rule Discovery The Association Rule discovery could be a common methodology while albeit slow, uses the correlation between various elements to spot anomalous activity. This method is employed in a very ”market basket analysis” case scenario that identifies irregularities within the purchasing conduct of supermarket clients. Also named as Boolean association rules, this system tries to spot various arrays of things within the market that buyers regularly buy consequently in every purchase. The disadvantages of this system are that it exponentially proliferates the occurrences because the number of elements grows.

9. Knowledge-Based Detection Technique Knowledge-based detection methodology may be a flexible technique that’s applied in both anomaly-based intrusion detection systems moreover as signature-based systems.This system captures and stores known intrusions and network threats. This stored information is then employed to mitigate future intrusions additionally as raise the edge alarm. Occurrences that don’t trigger the edge alarm are treated as safe events.

10. State Transition Analysis The State transition analysis methodology is an open-source technique that reviews possible intrusions through objectives and transitions. These state transition diagrams provide pictorial illustrations of events that an attacker can successfully perform to attack a network. The sequence of occurrences undertaken during an attack on a network is recorded for each compromised state. State transition illustrations recognize the necessities for each intrusion yet as causes for the attack’s success. These illustrations enable the identification of key occurrences that enable an intrusion possible.

11. Expert System An experts system could be a variety of AI, through which an automatic data processing system uses to imitate the decision-making capability of somebody’s.This method integrates a knowledge-based intrusion-detection methodology. The expert system comprises a group of rules which describe attacks. Audit events are thereafter translated into facts carrying their semantic signification within the expert system, and therefore the inference engine draws conclusions supported these rules and facts. This method increases the extent of abstraction of the audit data by attaching semantic thereto. Expert systems are integrated into both signature-based and anomaly-based intrusion detection systems in addition.

12. Signature Analysis The Signature analysis techniques employ the tactic of consolidating information like expert systems methodology. The signature approach, however, uses the knowledge that's consolidated differently, by deciphering and breaking down data into a series of appraised events, thereby decreasing the alarm threshold of the intrusion system. Efficiency is that the utmost trait of this signature-based analysis technique which has enabled its implantation within the market as a viable enterprise security system.The answer, however, incorporates a major bottleneck within the requirement for normal updates to safeguard the network from newly discovered threats.

13. Machine Learning-Based Detection Technique Machine learning is that the capability of an application or network system to amass and advance its security capability by consolidating data and data and building a concrete algorithm as a result. Machine learning-based detection is reliant on the development of a system that expands and progresses the flexibility to guard the network supported by improving on prior performance.This system enables the machine-based system usable in wide and ranging case scenarios. However, this machine learning intrusion detection system gobbles up system resources thus can partake a large amount of memory, bandwidth, and CPU time. Machine learning intrusion detection systems use either in artificial intelligent mathematical logic, artificial neural networks, Bayesian techniques, genetic algorithms, support vector machines, and Bayesian systems.

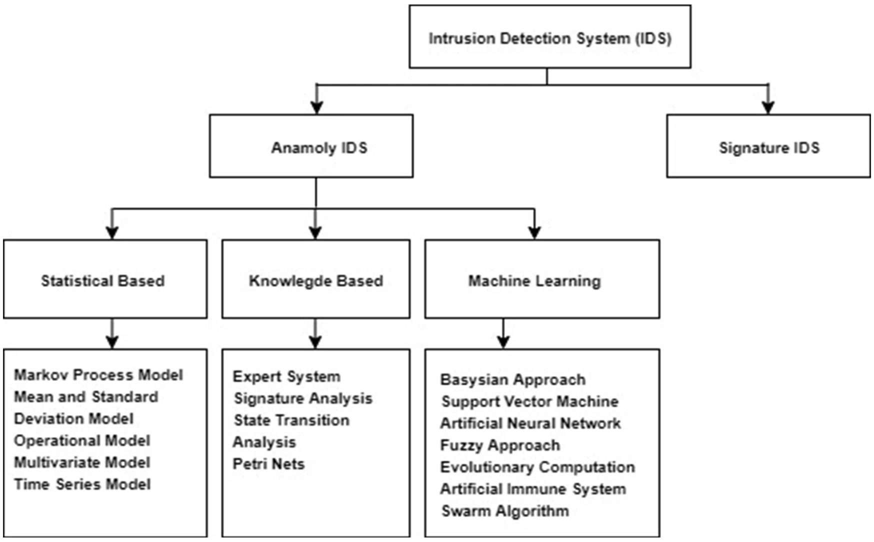


Figure 2.5: IDS taxonomy

2.9 AI

Artificial Intelligence (AI) was born within the 1950s when one or two pioneers from the nascent field of engineering science started asking whether computers can be made to think which continues to be being explored today. AI may be a general field that encompasses machine learning and deep learning, but it also includes more approaches that don't involve any learning [29].

2.9.1 Machine Learning (ML)

Machine Learning (ML) was born from the speculation that computers can learn without being programmed to perform specific tasks. It came into prominence perhaps within the 1990s when researchers and scientists started giving it more prominence as a sub-field of AI that performs much better compared to using fixed rule-based models requiring plenty of manual time and energy [30]. In traditional computing, a computer file is fed to a program to come up with the output. But in ML, computer file and output data are fed to the ML algorithm to get a function or program which will be wont to predict the output of input in keeping with the training done on the input/output dataset fed to the ML algorithm [31]. An ML system is trained instead of explicitly programmed and it has to learn from the historical data, optimize for better computations, and also generalize itself to supply proper results. Nowadays, machine learning algorithms are employed for classification, regression, clustering, or dimensionality reduction tasks of huge datasets with high-dimension. As a result, the task of detecting network intrusions comes under ML because it involves the classification of knowledge into normal and abnormal behavior. Network traffic data is increasing; thus, ML helps to acknowledge the complex patterns within the given large datasets by employing a learning mechanism to form decisions or predictions when new data instances are coming. ML algorithms are classified supported by the result of the algorithm and kinds of input fed during training as supervised, unsupervised, or semi-supervised learning [32].

Supervised Learning

Supervised learning may be a process by which a function springs with the assistance of labeled data samples also referred to as training data [33]. The algorithm analyzes the training data and produces a function, which can be used for input to output mapping of a replacement example. This learned knowledge is often then utilized in the long run to predict an output y' for any new computer file samples X which was previously unknown or unseen during the model training process. One among the challenges of supervised ML is that the labeling of an outsized number of information samples is usually done manually and it's time-consuming. But if we've got enough labeled data samples this sort of ML produces good prediction models. A supervised learning model has two major tasks to be performed, classification and regression. Classification: is worried about predicting output labels or responses that are categorical for input file for what the model has learned within the training phase [34]. Output labels, also called classes or class labels, are categorical which suggests they're unordered and discrete values. A classification problem will be binary classification (with two classes) or multiclass classification when there are three or more classes [35]. Common algorithms for performing classification include LR, NNs, support vector machine, ensembles like RFs and gradient boosting, k-nearest neighbors, DTs, Naïve Bayes, and then on. Regression: in contrast to classification, regression is worried about predicting the numeric value for the category label. Regression models use input file features and continuous numeric output values to be told specific relationships between the inputs and corresponding outputs [35]. With this data, the regression model can predict output responses for brand spanking new unseen data instances just like classification but with continuous numeric outputs. statistical regression and multivariate regression are several supervised regression algorithms.

Unsupervised Learning

Unsupervised learning deals with how systems can learn to represent particular input vectors in such a way that it reflects the statistical structure or pattern of a set of inputs [35]. In contrast to supervised learning, there are not any explicit target output labels related to each input. Here the task is to group unsorted information in step with patterns, similarities, and differences with no prior training data [36]. Clustering is one of the unsupervised learning methods which help to cluster or group data points into different groups or categories, without the supply of any output label within the input/training data.

Semi-supervised Learning

A semi-supervised learning method is another kind of ML that mixes supervised and unsupervised learning and is employed when a smaller number of labeled data is identified for a selected application [37]. It generates a function mapping from inputs of both labeled data and unlabeled data. The goal of this learning mechanism is to classify a number of the unlabeled data using the labeled set of data. The amount of unlabeled data should be over the amount of labeled data. Machine Learning Tasks Machine learning solves many alternative varieties of problems; we can group these problems into the subsequent tasks: [38]

- Classification outputs a predicted label from some predefined set.
- Regression outputs a predicted real-valued numerical result.
- Clustering groups objects such objects within the identical cluster have similar properties compared thereto of other clusters. There may or might not be a predefined number of clusters to suit the training data. Both classification and clustering are varieties of approaches that are suitable for this project. Classification aims to predict a category and assign the labels benign or attack. While clustering aims to divide the info into groups that are the same as one another. Here, we cluster the info into normal and anomalous cases.

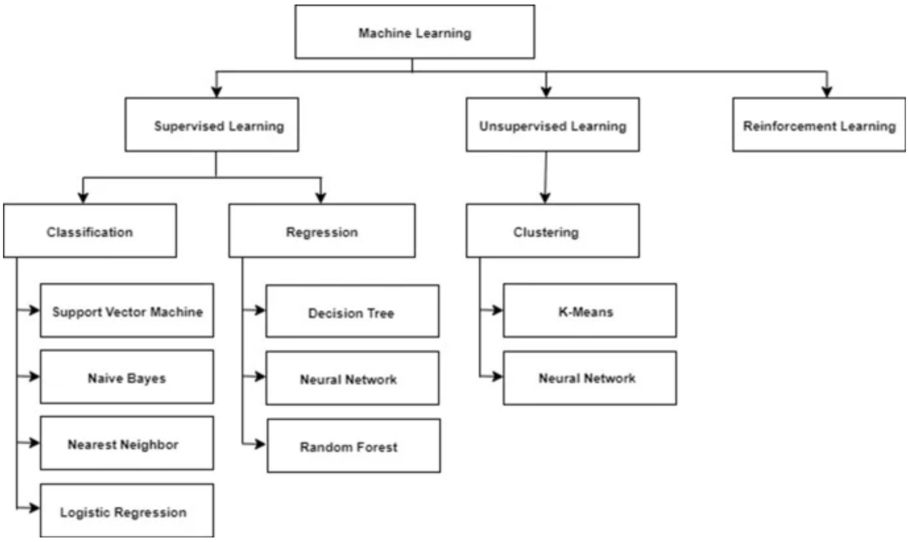


Figure 2.6: IDS taxonomy

Challenges Machine learning isn't without its challenges. The primary main challenge lies in choosing a useful dataset. It's vital that the dataset has relevancy to the task at hand and is sufficient in size so that the model has enough data to find out from. If performing supervised learning.

2.10 BACKGROUND THEORY

It adds the need to search out labeled data, which can be hard to seek out, or we may manually assign labels which might be time-consuming. **Overfitting** is another prominent issue in machine learning; it occurs when your model doesn't generalize well to new data. It can appear once you refine your model an excessive amount of that it also has learned all of the noise within the dataset, meaning that although the performances seem to be making improvements, it performs worse once you come to check it on new data. [39] The data may contain personally identifiable information, like IP addresses, so it must be anonymized, or the traffic simulated specified it contains no personal data. Furthermore, another issue faced is that there exists much variability in what's benign data, making it difficult for the model to be told what's 'normal' traffic, suggesting it's going to struggle to detect new attacks. Lastly, the value of miss-classification is high. If the IDS flags benign traffic as attacks too often, it may end up within the system being unusable. While if it classifies attacks as benign, this may cause severe damage since we don't seem to be registering that there has been an attack, leaving the network compromised. Thus, the system must have high performance. **Underfitting** is another issue that may occur with ML algorithms. Underfitting is that the inability of the model to predict well the labels of the info it was trained on. This problem usually happens after we have fewer data to create an accurate model and check out to coach a linear model using nonlinear data so the model produces plenty of wrong predictions. Such problems may be minimized by training models using more data and removing irrelevant features using feature selection mechanisms. Both underfit and overfit make the model too weak and unable to be generalized to any sample. So, model tuning helps models to generalize well on predicting any sample, whether training or testing.

2.10.1 Feature Selection

Feature selection is crucial in any machine learning model. It should be the case that the features included are those that improve the accuracy of the model [40]. Feature selection allows us to scale back the number of information points, which successively allows it to coach faster and reduces the possibility of overfitting. **Support Vector Machine** A support vector machine (SVM) could be a supervised, linear classifier. They create a hyperplane, and that we aim to seek out the maximum-margin hyperplane which divides our feature vectors of 1 class from the set of feature vectors of another class. This division could be a decision boundary to classify the information points [41]. **Decision Tree** Decision trees are a supervised method, often used for classification. To predict the category label for a given input, we start at the foundation node and chose the successor node that supported the worth of a selected feature. We repeat this until we reach a leaf node, which defines the expected class of this data. [41] **Naïve Bayes** The Naïve Bayes algorithm may be a supervised simple probabilistic classifier supported by applying Bayes' theorem. We assume that features are independent of every other, called the naïve assumption. To classify an input, we pick the category with the very best posterior probability, P(y—x), where x is that the feature vector and y is that the class [42]. **The K nearest neighbors (K-NN)** algorithm could be a supervised algorithm that works by assuming that data points of the identical class will exist nearby. The K refers to the number of neighbors we glance at the categorification of to see the class of computer file. [43] **K Means Clustering** K means the cluster may be an unsupervised clustering algorithm that groups object-supported their feature values into K disjoint clusters (where K is a positive integer specifying the number of clusters). It clusters data supported the Euclidean distance between the info, classifying objects into the identical cluster if they need similar feature values. [44]

2.10.2 Machine Learning Module

Machine Learning Module Machine Learning is a technique to train a model by providing the ample amount of past data to predict future patterns. So, if the model learns the pattern successfully, there are high chances it will predict correctly. Machine learning techniques are commonly applied to the problems which can not be solved by writing code or by mathematical means alone. Machine learning problems have two distinct approaches, and they are classified as Supervised Learning and Unsupervised Learning. Since we have used supervised learning, we will explain it further in brief. In supervised machine learning, algorithms are trained by providing them with pre-define collected data. This data is first labeled and based on these tags, the algorithm learns and then develops a model. The developed model then facilitates the accurate result when given a new data. The learning depends on the sophisticated algorithms and the training data. To give an idea about the working of training algorithm, we explain one of the simplest equation of the predictor function, where the predictor function or hypothesis function depends on two input values. It can be written in mathematical expression as follows:

$$h(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 \tag{2.1}$$

Where θ_0, θ_1 and θ_2 are coefficients and x1 and x2 is a independent variable or input data. General form of this equation can be written as follows:

$$h(x) = \sum_{i=0}^{n} \theta_i x_i \tag{2.2}$$

Where n is the total number of input data features or total number of input variables. When we train an algorithm, we actually find the values of coefficients. So if new data is passed to the function, based on the values of the input features, predictor function calculates the value. Depending on the value, whether it is below or above the threshold, new data is designated its class. We have shown the steps of supervised machine learning in figure 4.3. Further, under supervised machine learning, there are two important categories: 1) Regression and 2) Classification techniques. In regression method, the output of the hypothesis function lies on continuous spectrum whereas, in Classification technique, the output of the prediction function is in the form of distinct discrete classes. We have incorporated the machine learning module in our application to enhance periodically the anomaly detection rule. We have used Apache Spark scalable machine learning library MLlib for our purpose because of many of its advantages. It has built-in classification based algorithms like Decision Tree, Random Forest, and Naive-Bayes. MLlib has spark.ml higher-level API built on top of Spark dataframe, and it gives more versatility and is easy to use. Just like other Spark libraries, MLlib can run on Hadoop HDFS and because of its in-memory feature, Spark MLlib are quite fast. As mentioned earlier, packets features calculated in Spark Cluster is sent over to the monitoring system to detect any attack in the network, but at the same time, we store the collected features in HDFS for over a longer term. Since the pattern in the network traffic does not remain same and changes over the period, and hence there arises the need to update the trained model of a monitoring system. So whenever, there is a need to re-generate intrusion detection model or hypothesis function, we have enough data available to train. In our work, the five machine learning algorithms we have applied are

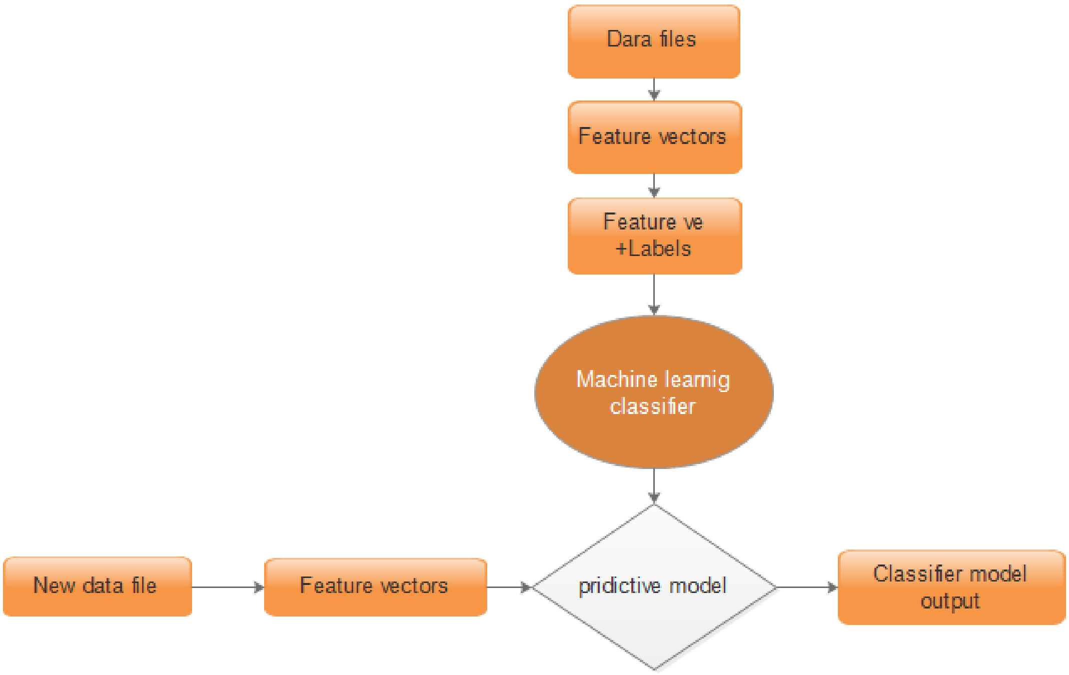


Figure 2.7: Flow diagram of various stages of supervised machine learning

(Support Vector Machine (SVM), Decision Tree, Naive Bayes, K Means Clustering, K Nearest Neighbor's. For training the model and then testing its accuracy, precision, and other parameters, we used the 70 : 30 ratio of test and training dataset. In the following, subsections, we will briefly describe the algorithm employed.

2.10.3 Naive-Bayes

It is a simple multiple-class classification algorithm with an assumption that a value of every feature is independent of the other regardless of any correlation between different features [24]. Despite being simple, Naive-Bayes algorithm has been quite successful in many physical world problems. It is based on Bayes' theorem, and it first calculates the conditional probability of each feature for a given label and then applies Bayes theorem to get the probability. Bayes theorem provides a method to find the posterior probability p(c/x), given P(c), P(x), and p(x/c) by following rule:

$$p(c/x) = \frac{(p(x/c) * p(c)}{p(x)} \tag{2.3}$$

where : P(c/x) it gives posterior probability for a class for given features x. P(c) is the prior probability of class. P(x/c) is the prior probability of features, given class. P(x) is prior probability of features. But in our case P(x) is not known, so there is another way round to solve this equation

$$P(c/x) = P(x0/c) * P(x1/c) * ...P(xn/c) * P(c) \tag{2.4}$$

Here, frequency table of each feature is drawn against the target and multiplied with the probability of class. Once, we get the posterior probability of each class; we compare them, and the class with higher probability gets labeled for particular data.

2.10.4 Support Vector Machines

Support Vector Machine (SVM) is a classification technique based on the concept of decision planes or hyperplanes that define class boundaries. A decision plane divides set of objects having different class into different categories. For a simple case, decision plane can be a straight line. However, for complex scenarios, a set of mathematical functions called kernels are applied to divide the data across a straight plane in a feature space of a higher order. New data points are then mapped to any one of the classes depending on the category they belong.

2.10.5 Decision Tree

Decision Tree builds classification models by splitting the training dataset based on values of the selected features. Features are divided across a value in a recursive manner, breaking the dataset into smaller subsets and in turn generating a tree called Decision Tree. Nodes in between the tree are called decision nodes; the terminal nodes are known as leaf nodes, and the topmost node is called root node. The algorithm to build decision tree is called ID3 and ID3 uses Entropy and Information Gain. Entropy is known as the measure of uncertainty.

$$E(S) = P - p_i log2p_i \tag{2.5}$$

where : pi is the probability of the class. The amount of Information gain or decrease in the entropy once data is split, indicates importance of an attribute in determining the target. Information Gain = entropy(parent node)-entropy(child node) In other words,

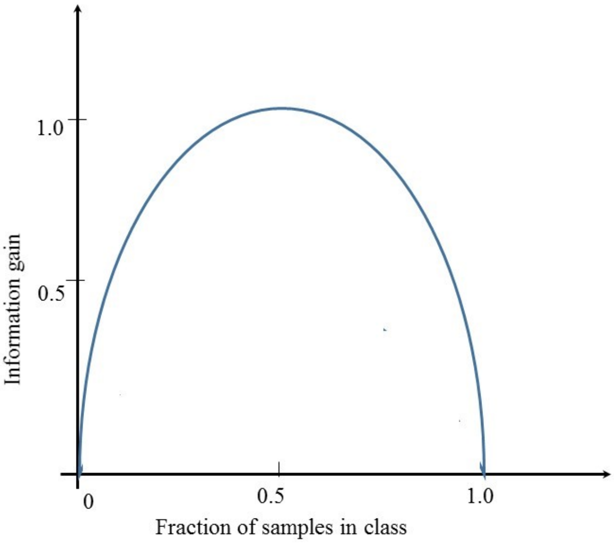


Figure 2.8: FInformation Gain curve with the variation in fraction of sample in complete classsize

we can say from the figure 4.4, when the fraction of samples in the data is exactly half, the information gain in selecting that node is maximum. On the other hand, when a sample is homogeneous, and there is only one kind of sample in data, Information gain is minimum because entropy is already minimum.

2.10.6 Random Forest

Random Forest is an algorithm of an ensemble of multiple Decision Trees. Each tree of a set gives its prediction result, and the label is given to the class with the most number of prediction in its favor. One of the main advantages of random forest is that it reduces the risk of over-fitting. More the number of trees in the random forest it's easier to tune its prediction. Randomness in a random forest is generated by selecting a random subset of training data to build trees. Subset data can have all the input attributes but a limited number of records or all the data records but a subset of attributes or subset of records and attribute both. Random Forest takes care of an unbiased estimate of the test set error, and it is determined internally during run-time.

2.10.7 k- nearest neighbor algorithm (k-NN)

k- nearest neighbor algorithm (k-NN): This is the most simple among all machine learning algorithms where the output is calculated based on k closest neighbours or k training patterns. The calculation of output varies depending on the task to be performed. For example in case of classifying an unknown pattern, the pattern is assigned as the class which appears frequently among the k nearest training patterns.

Chapter 3

Related work

3.1 Intrusion Detection Systems

Intrusion is any set of actions that try to compromise the integrity, confidentiality, or availability of a resource [45] and an intrusion detection system (IDS) could be a system for the detection of such intrusions. There are three main components of an IDS: data collection, detection, and response. The information collection component is chargeable for collection and pre-processing data tasks: transferring data to a standard format, data storage, and sending data to the detection module. IDS can use different data sources as inputs to the system: system logs, network packets, etc. Within the detection, component data is analyzed to detect intrusion attempts and indications of detected intrusions are sent to the response component. Within the literature, three intrusion detection techniques are used.The primary technique is anomaly-based intrusion detection which profiles the symptoms of normal behaviors of the system like usage frequency of commands, CPU usage for programs, and therefore the like. It detects intrusions as anomalies, i.e. deviations from the conventional behaviors. Various techniques are applied for anomaly detection, e.g. statistical approaches and computing techniques like data processing and neural networks. Defining normal behavior could be a major challenge. Normal behavior can change over time and intrusion detection systems must be maintained to this point. False positives – the traditional activities which are detected as anomalies by IDS – are often high in anomaly-based detection. On the opposite hand, it's capable of detecting previously unknown attacks. This is often important in an environment where new attacks and new vulnerabilities of systems are announced constantly.

Misuse-based intrusion detection compares known attack signatures with current system activities. It's generally preferred by commercial IDSs since it's efficient and contains a low false-positive rate. The downside of this approach is that it cannot detect new attacks. The system is just as strong as its signature database and this needs frequent updating for brand new attacks. Both anomaly-based and misuse-based approaches have their strengths and weaknesses. Therefore, both techniques are generally employed for effective intrusion detection.The last technique is specification-based intrusion detection.During this approach, a group of constraints on a program or a protocol are specified and intrusions are detected as runtime violations of those specifications.It's introduced as a promising alternative that mixes the strengths of anomaly-based and misuse-based detection techniques, providing detection of known and unknown attacks with a lower false-positive rate. It can detect new attacks that don't follow the system specifications. Moreover, it doesn't trigger false alarms when the program or protocol has unusual but legitimate behavior, since it uses the legitimate specifications of the program or protocol.It's been applied to ARP (Address Resolution Protocol), DHCP (Dynamic Host Configuration Protocol), and plenty of MANETS routing protocols. Defining detailed specifications for every program/protocol may be a really time-consuming job. New specifications also are needed for every new program/protocol and therefore the approach cannot detect some reasonable attacks like DoS (Denial of Service) attacks since these don't violate program specifications directly [46]. When an intrusion is detected, an appropriate response is triggered in keeping with the response policy. Responses to detected intrusions will be passive or active. Passive responses simply raise alarms and notify the correct authority. Active responses try and mitigate the consequences of intrusions and are divided into two groups: those who seek control over the attacked system, and people that seek control over the attacking system. the previous tries to revive the damaged system by killing processes, terminating network connections, and therefore the like. The latter tries to stop the attacker's future attempts, which might be necessary for military applications.

3.2 Intrusion Detection Issues in MANETs

Different characteristics of MANETs make conventional IDSs ineffective and inefficient for this new environment. Consequently, researchers are working recently on developing new IDSs for MANETs or changing these IDSs to be applicable to MANETs. There are new issues that ought to be taken under consideration when a replacement IDS is being designed for MANETs. Lack of Central Points MANETs doesn't have any entry points like routers, gateways, etc. These are typically present in wired networks and maybe accustomed monitor all network traffic that passes through them. A node of a mobile unexpected network can see only a little of a network: the packets it sends or receives along with other packets within its radio range. Since wireless unexpected networks are distributed and cooperative, the intrusion detection and response systems in MANETs might also have to be distributed and cooperative. This introduces some difficulties.For instance, distribution and cooperativeness of IDS agents are difficult in an environment where resources like bandwidth, processor speed, and power are limited. Furthermore, storing attack signatures in an exceedingly central database and distributing them to IDS agents for misuse-based intrusion detection systems isn't suited to the present environment. Mobility MANET nodes can leave and join the network and move independently, that the topology can change frequently.

The highly dynamic operation of a MANET can cause traditional techniques of IDS to be unreliable.For instance, it's hard for anomaly-based approaches to differentiate whether a node emitting out-of-date information has been compromised or whether that node has yet to receive the updated information. Another mobility effect on IDS is that IDS architecture may change with changes to the configuration. Wireless Links Wireless networks have more constrained bandwidth than wired networks and link breakages are common. IDS agents have to communicate with other IDS agents to get data or alerts and want to bear in mind wireless links. Because heavy IDS traffic could cause congestion then limit normal traffic, IDS agents have to minimize their data transfers. Bandwidth limitations may cause ineffective IDS operation.For instance, an IDS might not be ready to reply to an attack in real-time thanks to communication delay. Furthermore, IDS agents may become disconnected because of link breakages.

An IDS must be capable of tolerating lost messages whilst maintaining reasonable detection accuracy. Limited Resources Mobile nodes generally use battery power and have different capacities. MANET devices are varied, e.g. laptops, handheld devices like PDAs (personal digital assistants), and mobile phones. The computational and storage capacities vary too.The variability of nodes, generally with scarce resources, affects the effectiveness and efficiency of the IDS agents they support.For instance, nodes may drop packets to conserve resources (causing difficulties in distinguishing failed or selfish nodes from an attacker or compromised nodes) and memory constraints may prevent one IDS agent from processing a big number of alerts coming from others.

The detection algorithm must take into consideration limited resources.For instance, a misuse-based detection algorithm must take into consideration memory constraints for signatures and an anomaly-based detection algorithm has to be optimized to scale back resource usage. Lack of a transparent Line of Defense and Secure Communication Manets don't have a transparent line of defense; attacks can come from all directions.As an example, there are not any central points on MANETs where access control mechanisms are often placed. Unlike wired networks, attackers don't gain physical access to the network to use some varieties of attacks like passive eavesdropping and active interference (these require only radio contact). Furthermore, the critical nodes (servers, etc.) can't be assumed to be secured in cabinets, and nodes with inadequate protection have a high risk of compromise and capture. IDS traffic should be encrypted to avoid attackers learning how the IDS works [47]. However, cryptography and authentication are difficult tasks during a mobile wireless environment since they consume significant resources. In many cases, IDS agents risk being captured or compromised with drastic consequences in a very distributed environment.They will send false alerts and make the IDS ineffective. IDS communication may also be impeded by blocking and jamming communications on the network. Cooperativeness MANET routing protocols are usually highly cooperative.This could make them the target of the latest attacks.For instance, a node can pose as a neighbor to the opposite nodes and participate in decision mechanisms, possibly affecting significant parts of the network.

An intrusion detection system (IDS) is a direct monitoring system and provides a warning when it finds any abnormality. In recent years, many intrusion detection methods are proposed.During this section, we describe the related work of intrusion detection in ad-hoc networks.

On the opposite hand, with the age of massive data coming, many methods are proposed so as to resolve anomaly detection in large-scale datasets.During this section, a straightforward survey of major machine learning techniques utilized in IDS for MANETS presented [48].

Sun et al. [49] proposed a brand-new intrusion detection model that supported the Markov process against the disruption attack in Mobile unexpected networks. The performance of mobile unintended networks (MANETs) is significantly full of malicious nodes.One of the foremost common attacks in MANETs could be a denial of service (DoS); a sort of intrusion specifically designed to focus on service integrity and availability of a particular network node. Hence, it's important to use an efficient intrusion system (IDS) that detects and removes the malicious nodes within the network to enhance the performance by monitoring the network traffic continuously.

Wahab et al. [50] have presented an intrusion detection scheme using SVM over clustered vehicular ad hoc networks. The aim of this ID model is to reduce the size of the training set for the SVM classifier and its advantage is to support for high mobility environment. Various kernel functions are used to test the performance of SVM. Finally, the proposed method has proved that it improves the scalability of the network concerning a number of nodes (normal and malicious). A drawback of this work is SVM since it failed to tune the parameter set and very complex to obtain better results.

Singh and Bedi [51] have discussed multiclass extreme learning machine-based Smart Trustworthy IDS with a single hidden layer feed-forward neural network to categorize nodes into trustworthy, partially trustworthy, and malicious in KDD Cup Dataset. There are five agents are used in this paper such as data accumulation agent, preprocessing agent, trust degree computation agent, differentiation agent, and decision-making agent. ELM has proved that it suitable for intrusion detection in real-time, but it failed to improve the speed of attack detection evaluation.

Kolias et al. [72] have proposed IDS to detect the most popular attacks on 802.11 using several algorithms (Adaboost, J48, Naïvebayes, OneR, Random Tree, Random Forest, ZeroR). Aegean WiFi Intrusion Dataset (AWID) is used in this work and also it is suited for UMTS, LTE, WiMax technologies. It is showed that J48 and random forest classification algorithms provide a high detection rate and low false alarm rate. These two algorithms are simple and easy to use, but they failed to support large-scale datasets. Hence scalability is not achieved. Subba et al. [73] have discussed hybrid IDS with Bayesian game formulation to detect deprivation, flooding, DoS and foraging, blackhole attack, packet dropping attack by using unsupervised association rule mining (ARM) algorithms such as Apriori and Vickrey–Clarke Gorves (VCG). Furthermore, a threshold-based lightweight module and powerful anomaly-based heavyweight module is proposed to obtain lower power consumption. The proposed model is heavyweight and thus it provides low attack detection rate and high false alarm rate.

Ahmed et al. [54]have presented a new framework for DoS attack detection using finite state machines (FSM). An intrusion detection system with ad hoc on-demand distance vector (ID-AODV) protocol is proposed, which functions by FSM. There are three operational modules are involving in ID-AODV such as network monitoring, FSM, and DoS detection. In the simulation, ID-AODV shows that it obtained a better attack detection rate to show high security of mobile nodes in data transmission and collection, but the authors do not convey about detection delay.

Shanthi et al. [55]discussed the concept of intrusion detection and secure key management in MANET using trust metrics. For each mobile node direct and indirect is computed and hierarchical group key management is proposed for information access control. The base station is deployed in-network for group key generation, distribution, and management. Through this work, network lifetime and packet delivery ratio are improved when the presence of attackers, but attack detection rate with the use of trust metric is not investigated.

Khan et al. [56]discussed the detection and prevention of attackers in the network. In order to detect malicious nodes in the network, detection and prevention nodes are deployed in the network. If it determined any suspicious node, then broadcast this error message throughout the network. Data packets forwarded by the suspicious node are eliminated in the network. For intrusion detection and prevention, more statistical analysis and computation are required. This will result in high overhead and large energy consumption of the network.

Raja and Ganesh Kumar [57] have proposed a trusted cluster-based routing protocol for MANET. Trust management (TM) is concentrated in this paper where they compute a trust value for all mobile nodes. When a node has a high trust value, then those are considered to be trusted nodes. The goal of this paper is to establish TM based routing protocol to enhance QoS in MANET. Simulation results proved that it obtained better performance for succeeding metrics: energy consumption, throughput, packet delivery ratio, and delay. Mobile node's behavior is not a constant, which leads to given the wrong opinion of someone.

Anusha and Sathiyamoorthy [58] discussed an intrusions detection mechanism for MANET using trust-based authentication and bio-inspired optimization algorithms. In order to prevent intrusions, a certificate authority is deployed in MANET which generates public and private key pairs. Deep packet inspection is implemented in this paper to improve MANET security and hence packet features are extracted for deep packet inspection. When an attacker is deterred- mined in deep packet inspection, an error message is sent to a certificate authority for taking necessary actions. The asymmetric technique can be used for message encryption and signing (validation), but it is very resource-intensive and only supported and work well in small messages.

Luong et al. [59] proposed a new protocol named FAPRP, which is expanded as flooding attacks prevention routing protocol. This FAPRP is based on a machine learn- ing approach implemented and tested over MANET. FAPRP is an extended version of the AODV routing protocol created to mitigate flooding attacks. Experiments conducted and validated that FAPRP has reached a 99% of detection rate for flooding attacks. However, flooding is an initial attack, which is easily mitigated through packet header information, but several security attacks are still unsolved in MANET. One research work towards this idea i.e. detecting new security attacks in MANET is detailed in [28]. In this paper, the authors have proposed a node collusion method to classify normal and attacker nodes, which intends to mitigate two security attacks: wormhole and sinkhole attacks. For routing attacks prevention, the route reserve method is proposed.

Ahmad et al. [60]studied the performance comparison of RF, SVM, and ELM for network intrusion detection. Each technique is applied to detect intrusions with the trained NSL-KDD set. Finally, the authors have concluded that ELM is a suitable scheme for intrusions detection and validated for the large size of the dataset. This work tends to increase detection time since processing all preprocessed data with feature extraction and selection is time-consuming.

Yin et al. [61] tested the NSL-KDD dataset using a recurrent neural network (RNN) and the performance of RNN is compared with several classifiers such as J48, SVM, RF, and so on. It is supported for binary and multi-class classification. It shows a better accuracy rate in intrusion detection. The training time of RNN is higher and hence authors have suggested that in the future long short-term memory (LSTM) or gated recurrent unit (GRU) is used to address the issue.

Recently, Khan et al. [62]proposed convolutional LSTM and spark ML (machine learning) is proposed for intrusion detection. However, both convolutional LSTM and spark ML require a large amount of data for the training process, and also computations of this combined algorithm are very large.

Xu et al. [**?**]proposed a GRU for network intrusion detection. In this paper, RNN is integrated into GRU for improving intrusion detection performance. Two different datasets are tested such as KDD 99, and the NSL-KDD dataset. The high total detection rate is 99.42% and 99.31% for KDD and NSL-KDD datasets, respectively. Similarly, they obtained low false-positive rates such as 0.05 and 0.84 for KDD 99 and NSL-KDD dataset, respectively. The attack detection rate is very high, but detection time for intrusions becomes very high. It must be less to demon- state the system has obtained better performance.

Shams, E. A.& Rizaner, A. (11,2018) [64] have proposed a hybrid approach of IDS with Support vector machine (SVM) to identify DoS attack in MANET. The main contribution of this paper is the integration of an IDS into MANETs as a reliable and potent solution. A new approach to intrusion detection is developed based on support vector machine algorithm. The proposed IDS can detect the DoS type attacks at a high detection rate with a simple structure and short computing time. It is shown by extensive computer simulation that the proposed IDS improves the reliability of the network significantly by detecting and removing the malicious nodes in the system. The performance of the suggested approach is independent of the network routing protocol. The detection rate of the system is also not effected by node mobility and network size. The attack is detected with a detection rate of approximately 95%.

Sen et al. (12, 2018) [65] have presented a trust model that behaves similar to IDS and utilized for the detection of black hole attack in MANET. This paper looks at developing such a trust model which is applied to all the nodes in the network. The trust model works like an Intrusion Detection System (IDS), which seeks to detect blackhole attacks in the system, and then identify and mitigate the malicious attacker. It is possible to use this characteristic feature of MANET and devise a trust model to monitor network activity. Therefore, this document has looked at security measures in MANETs and devised a trust-based IDS system against blackhole attacks. The proposed mechanism is able to provide a substantial improvement in the affected network in terms of throughput and PDF, although it experiences higher end-to-end delays. 20931 Kbps throughput with a delay of 2260ms has been achieved.

Sowah et al. (17,2019) [66]have proposed a detection and prevention algorithm against Man-in-the-Middle named as Artificial neural network with a detection rate of 88.235%.In this paper, ANN classification methods in intrusion detection for MANETs were developed and used with NS2 simulation platform for attack detection, identification, blacklisting, and node reconfiguration for control of nodes attacked. The ANN classification algorithm for intrusion detection was evaluated using several metrics. The performance of the ANN as a predictive technique for attack detection, isolation, and reconfiguration was measured on a dataset with network-varied traffic conditions and mobility patterns for multiple attacks. With a final detection rate of 88.235%, this work not only offered a productive and less expensive way to perform MITM attacks on simulation platforms but also identified time as a crucial factor in determining such attacks as well as isolating nodes and reconfiguring the network under attack.
Amar Amouri [67] proposed a cross layer-based IDS with two layers of detection. It uses a heuristic approach which is based on the variability of the correctly classified instances (CCIs), which we refer to as the accumulated measure of fluctuation (AMoF). The current, proposed IDS is composed of two stages; stage one collects data through dedicated sniffers (DSs) and generates the CCI which is sent in a periodic fashion to the super node (SN), and in stage two the SN performs the linear regression process for the collected CCIs from different DSs in order to differentiate the benign from the malicious nodes. In this work, the detection characterization is presented for different extreme scenarios in the network, pertaining to the power level and node velocity for two different mobility models: Random way point (RWP), and Gauss Markov (GM). Malicious activity used in the work are the blackhole and the distributed denial of service (DDoS) attacks. Detection rates are in excess of 98% for high power/node velocity scenarios while they drop to around 90% for low power/node velocity scenarios
A Comprehensive Review [68] provides an honest overview of previous papers on this subject, the algorithms used, and their respective detection accuracy. The range of IDS accuracy for the papers listed within the survey was between 88.0% and 99.9%, with the median accuracy being 97.2%. The paper which had the very best accuracy uses the KDD Cup 1999 dataset, partitions the info into two classes (binary or attack), and trains employing a support vector machine. This review provides us witan a plan of the values that we should always expect to succeed in during this dissertation. the bulk of the papers referenced within the review use the KDD Cup 1999 dataset. It contains nearly five million data points, 41 extracted connection features, and 24 differing kinds of attacks. Labeled attack data accounts for 80.3% of the dataset. Its size makes it very appealing because it contains an enormous number and type of attacks. However, it's now twenty years old, making it outdated, irrelevant, and not suitable to be utilized in

this domain.Furthermore,it doesn't represent realistic network traffic since there's deficient benign labeled data, so mustn't be utilized in systems aimed to be deployed during a real network environment.

Mahendra Prasad1(B),Sachin Tripathi1, and Keshav [76] Dahal,"Intrusion Detection in Ad Hoc Network Using Machine Learning Technique"Blackhole attack is an important routing disruption attack that malicious node advertises itself as part of a path to the destination. In this paper, we have simulated blackhole attack in ad hoc network environment and collected data of essential features for attack behaviors classification. Then, many machine learning techniques have applied for classification of benign and malicious packet information. It suggests a new approach for select features, essential information collection, and intrusion detection in ad hoc network using machine learning techniques. We have shown comparative results of different machine learning techniques. Our results indicate that this approach can use with different classifiers and can extend it with other intrusions.Machine learning techniques are applied to this set of data which work in the supervised mode of training. Experiments show the simulated blackhole attack such activities, and various machine learning techniques provide their detection accuracy. Where MLP is shown the better result to other classifiers, it has shown 98.5% detection rate and 4.7% false alarm rate.

M. Islabudeen [69] proposed a Smart approach for intrusion detection and prevention system (SA-IDPS) to mitigate attacks in MANET by machine learning methods. Initially, mobile users are registered in Trusted Authority using One Way Hash Chain Function. Each mobile user submits their following information to verify authentication: finger vein biometric, user id, and latitude and lon- gitude. Intrusion detection is executed using four entities: Packet Analyzer, Preprocessing Unit, Feature Extraction Unit and Classification Unit. In packet analyzer, we verify whether any attack pattern is found or not. It is implemented using Type 2 Fuzzy Controller which considers information from packet header. In preprocessing unit, logarithmic normalization and encoding schemes are considered, which is time series and suitable for any application. In feature extraction unit, Mutual Information is used where we extracts optimum set of features for packets classification. In classification unit, Bootstrapped Optimistic Algo- rithm for Tree Construction with Artificial Neural Network is used for packets classifica- tion, which classifies packets five classes: DoS, Probe, U2R, R2L, and Anomaly, and then Association Rule Tree are used to classify whether the attack is Frequent or Rare. In this case, historical table is used for packets classification. Finally, experiments are conducted and tested for evaluating the performance of proposed SA-IDPS scheme in terms accuracy 99.74%.

Paper author	Title	Method/Algorithm	Result	Paper Gab
Xu, C., Shen, J., Du, X., & Zhang, F. (2018).	An intrusion detection system using a deep neural network with gated recurrent units	RNN is integrated to GRU for improving intrusion detection performance	High total detection rate is 99.42% and 99.31% for KDD and NSL-KDD dataset,respectively.	The paper test only on two dataset KDD and KSL-KDD which is not prepared from manet network
Shams, E. A.&Rizaner, A. (11,2018	hybrid approach of IDS with Support vector machine (SVM) to identify DoS attack in MANET	A new approach to intrusion detection is developed based on support vector machine algorithm	The detection rate of the system is also not effected by node mobility and network size. The attack is detected with a detection rate of approximately 95%.	The paper only detect know attack it does't deal with zero day attack and also the accuracy is low compare to other machine learning
Fang Feng et.	Anomaly detection in ad-hoc networks based on deep learning model: A plug and play device	Deep neural network (DNN) detection model to detect DoS attacks; DNN, Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM)	DoS attacks, Accuracy, Precision, Recall and of DNN detection model is respectively 0.9994, 0.9763, 0.9959, 0.986	The paper didn't describe how to perform the attack on the manet network.The paper work on NS2 to generate the dataset so its not effective for deep learning

M. Islabudeen1 · M. K. Kavitha Devi	A Smart Approach for Intrusion Detection and Prevention System in Mobile Ad Hoc Networks Against Security Attacks	In classification unit, Bootstrapped Optimistic Algorithm for Tree Construction with Artificial Neural Network is used for packets classification	Finally, experiments are conducted and tested for evaluating the performance of proposed SA-IDPS scheme in terms accuracy 99.74%.	The main drawback of the paper is tested for small size of real world dataset.
Sowah et al. (17,2019)	A scalable and hybrid intrusion detection system based on the convolutional-LSTM network. Symmetry	cross layer-based IDS with two layers of detection	The ANN classification algorithm for intrusion detection was evaluated using several metrics.	With a final detection rate of 88.235%

Mahendra Prasad1(B)et al.	Intrusion Detection in Ad Hoc Network Using Machine Learning Technique	many machine learning techniques have applied for classification of benign and malicious packet information	Machine learning techniques are applied to this set of data which work in the supervised mode of training. Experiments show the simulated blackhole attack such activities, and various machine learning techniques provide their detection accuracy. Where MLP is shown the better result to other classifiers, it has shown 98.5% detection rate and 4.7% false alarm rate	It took more training time and the dataset is prepared on NS2 not on real time%

Table 3.1: summary of related work

Chapter 4

Proposed Method

4.1 General Overview

This section presents an summary of the proposed machine learning approach applied in anomaly detection on MANET. We present the initial steps taken to urge the dataset ready for training, and the way we implemented the chosen machine learning algorithms. Moreover, because the development of the intrusion detection system, and the way to perform a spread of intrusion attacks.We have proposed and developed an IDS to detect and avert intrusion in MANET in real-time. In figure 4.1, we have shown the architectural diagram of the IDS system we have developed. IDSs can be placed in all node connected to Ad hoc network; installing it is a matter of mutual concession and compromise because, if we install the IDS at one node, it will lead to huge computational cost and may be increased latency. However, if we install IDS in a place in all node, then the number of IDS to be installed have to be increased to monitor the complete ad hoc network. So, IDS have to be placed carefully and judiciously in the each node, so that computing resources are utilized optimally and at the same time there is no or low latency in response while monitoring a critical system. The individual phases of the proposed framework are Initially, the first phase (Pre-process dataset) Apply data cleaning to the input local prepared dataset. In the second phase (Normalization) this phase will normalize local prepared datase by using the min-max technique. The third phase (Splitting) Split the Dataset into training, validation, and test sets. The fourth phase(machine Learning) Construct the classifier with enhanced the dataset from phase two and Training the machine learning models as classifiers in binary and multi-class modes. The last phase (Evaluation) Evaluate our proposed intrusion detection system based on machine learning techniques using accuracy and loss for each model and for both classifications binary and multi-class. The detailed description of each step in the proposed phase is explained below,

1. Phase 1 (Data cleaning): This stage will be divided into two steps because the first step is that the original dataset consists of many missing values in some feature columns which cannot be fed to machine learning. So that in this stage we filled all empty cells with “0” value which would represent the missing value. While the second step is that converting the stored cells as text needs to be converted to numerical value where each categorical value will be represented as a specific

numerical value.

2. Phase 2 (Normalization): Normalization of data was especially helpful for systems in which the measurements are commonly represented on vastly different levels. Min-max normalization helps to build neural networks more consistently. This method for normalization has the advantage of accurately maintaining all data connections and therefore does not contribute to any prejudice. The increasing function is below the correct value range for the classification as min-max is added, but the respective distributions of the related features stay inside the current value range [10].

3. Phase 3 (Split Dataset): In this stage, we split the main dataset into a 70% training set, 15% for validation set, and 15% test set.

4. Phase 4 (Machine learning-Intrusion Detection System): we proposed an intrusion detection system based on machine learning techniques with five types of machine learning models which is (SVM, Decision Tree, Naive Bayes, K Means Clustering, K Nearest Neighbor's).

5. Phase 5 (Evaluation): This will evaluate machine learning models as intrusion detection techniques and measure their performance with normal machine learning techniques based.

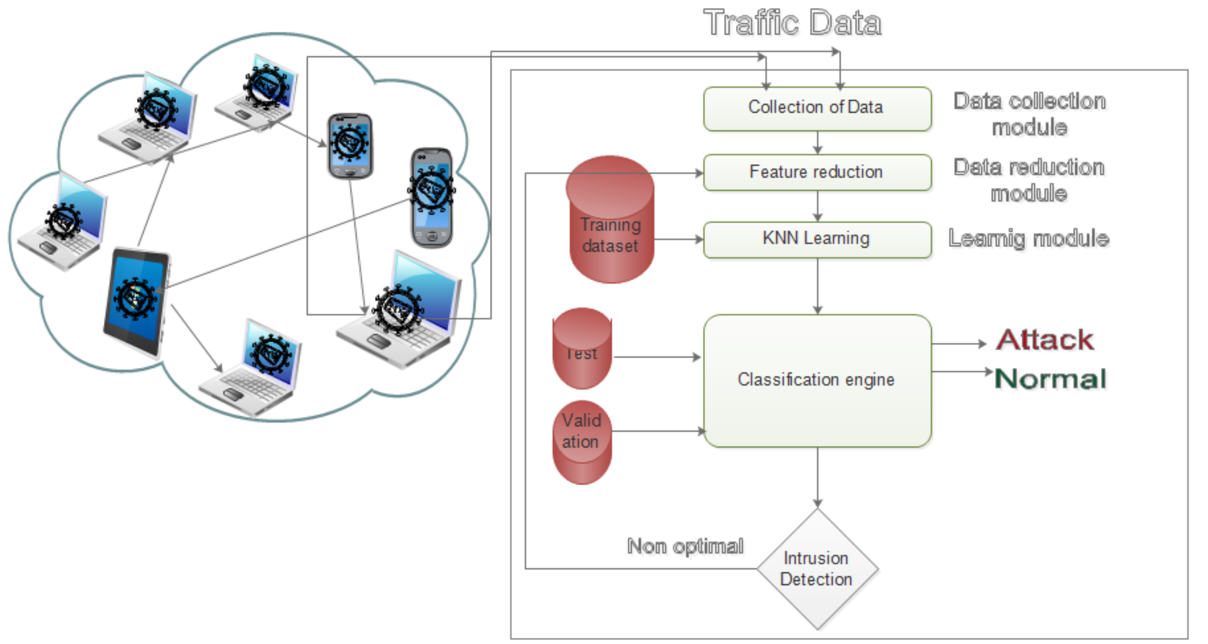


Figure 4.1: Architectural diagram of the developed IDS

As shown in figure 4.1, each node has multiple machines and these machines communicate among themselves.

4.2 Proposed Methodology

In this section, a brief description of our research methodology is given. We had started our work with the comparison of different machine learning-based classification models from a different perspective. Then we had proposed a hybrid machine learning approach for a better Intrusion Detection System. After that, we had also proposed a feature selection method which would help us to find out the important feature for improvement of our proposed hybrid machine learning algorithm for Intrusion Detection System.

4.3 Feature selection

Feature selection selects representative set of attributes from the set of original attributes. This representative set keeps only the relevant and important attributes, learning algorithm takes less time to learn and produces a more general classifier as it removes unnecessary and irrelevant attributes for the original set. Feature selection also facilitates data visualization and data understanding. Some of the popular feature selection techniques used in this paper is briefly presented below.

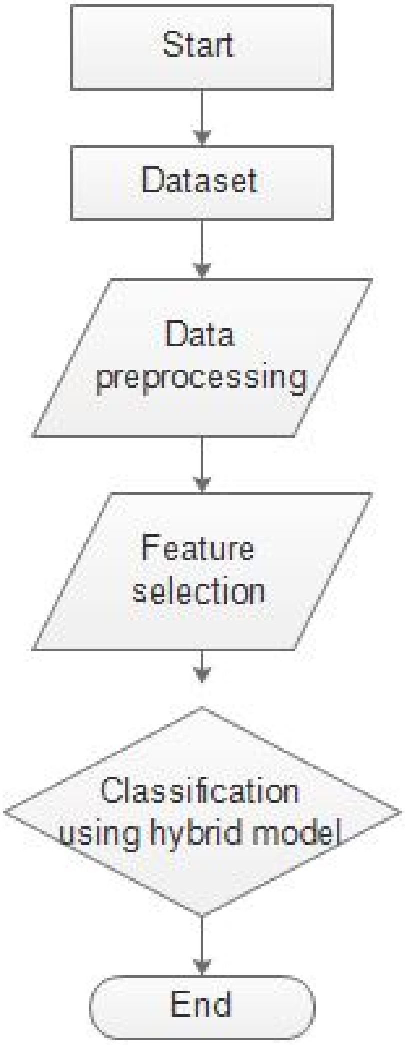


Figure 4.2: Flow chart showing various steps involved in Intrusion Detection System

1. Correlation based Feature Selection method: CFS works with hypothesis that is ”Good feature subsets contain features highly correlated with the class, yet uncorrelated to each other”

   Algorithm:

   (a) Select the dataset for pre-processing.

   (b) Calculate featurefeature and feature-class correlations.

   (c) Search through the feature subspace and calculate feature subset based on merit.

2. Principal Component Analysis: Principal component analysis (PCA) determines uncorrelated attributes called principal components.

   Algorithm:

   (a) Whole d-dimensional dataset is taken ignoring the class labels.

   (b) The d-dimensional mean vector is calculated.

   (c) Covariance matrix is found for the whole data set.

   (d) Eigenvectors and corresponding eigenvalues are calculated.

   (e) Eigenvectors by decreasing eigenvalues are sorted and l eigenvectors with the largest eigenvalues to form a d × l dimensional matrix M are chosen.

   (f) M is used to transform the samples onto the new subspace. Mathematically it can be written as:y = MT × p (Where p is a d × 1 dimensional representing one sample and y is the transformed l ×l dimensional sample in the new subspace).

3. Information Gain Ratio based feature selection: Features selected based on only information gain is biased towards attributes having many values. Information Gain Ratio (IGR) based Feature Selection removes this drawback by taking the splitting information of an attribute into account. Splitting information of an attribute is the entropy of pattern distribution into branches. Gain ratio of attribute decreases as value of split information increases.

   Algorithm:

   (a) Start with the full set of attributes (set containing all attributes of the dataset) and null selected feature set.

   (b) Calculate information gain ratio of each attribute.

   (c) Choose an attribute from the total set with the highest information gain ratio.

   (d) Split the dataset into sub datasets depending on the attribute values.

   (e) Add the attribute to selected feature set and remove from set of attributes.

   (f) Repeat step 2 to 5 for each of the sub-datasets with the set of attributes, if instance in a sub-dataset belongs to more than one class.

   (g) Output the selected feature set.

4. Minimum Redundancy Maximum Relevance: This method tries to penalize a feature's relevance based on its redundancy. The relevance of a feature set S

for the class c is defined by the average value of all mutual information values between the individual feature fi and the class c .

4.3.1 Building the Intrusion Detection System

Last few years, researchers have designed intrusion detection and prevention based on conventional approaches, which are not giving predominant results in the aspect of attack detection rate and false positive rate. To mitigate such issues, in this paper we proposed a machine learning intrusion detection system in mobile ad hoc environment. Our proposed comprised of Mobile Devices (MDs), Packet Analyzer, Preprocessing Unit, Feature Extraction Unit, and Classification Unit. According to the definition of MANET, mobile users are moved rapidly for several locations in ad hoc environment. Network traffic occurs when data packets are received from nearby mobile users. We introduced intrusion detection engines for mitigating attacks. Packet analyzer will scrutinize the packets based on packet arrival time, num. of packets per flow, packet counts, and packet size from its packet header. Threshold for classifying attack pattern and normal pattern is determined using T2FC, which improves uncertainty while classifying packets. Then attack pattern found packets are forwarded to preprocessing unit, which executes two steps: encoding and normalization. Then normalized packets are forwarded to feature extraction unit, where we extract most optimum set of features, and then classification unit is initiated for packets classification and further it is identified whether rare attack or frequent attack using KNN and KMeans.

This time, we use the entire dataset to coach the classifier since we don't must hold out any for testing and validation. Once training of the model is complete, the system starts analyzing the incoming traffic packet by packet. This traffic may either be live, using the machine's network interface, or from a supplied. Pcap file. A .pcap file contains packet data created during network capture and allows us to research previously collected data. This section elaborates the proposed method of intrusion detection. We assume that the ad hoc network comprises N bidirectional communication nodes in the network space that share packets or information over a shared wireless medium. This network space contains N-M normal nodes and M malicious nodes. Malicious nodes tune their behaviors and perform malicious activities. This method starts with feed data and simulates blackhole attack with malicious nodes. Subsequently, it gathers basic information of nodes which are in ad hoc network in a specified format. Then, this process selects essential features and collect data that build a dataset. Finally, we have applied many ML techniques for classification of information and provided the valid decision. A sequence of work is described in Algorithm 1.

Algorithm 1 intrusion detection

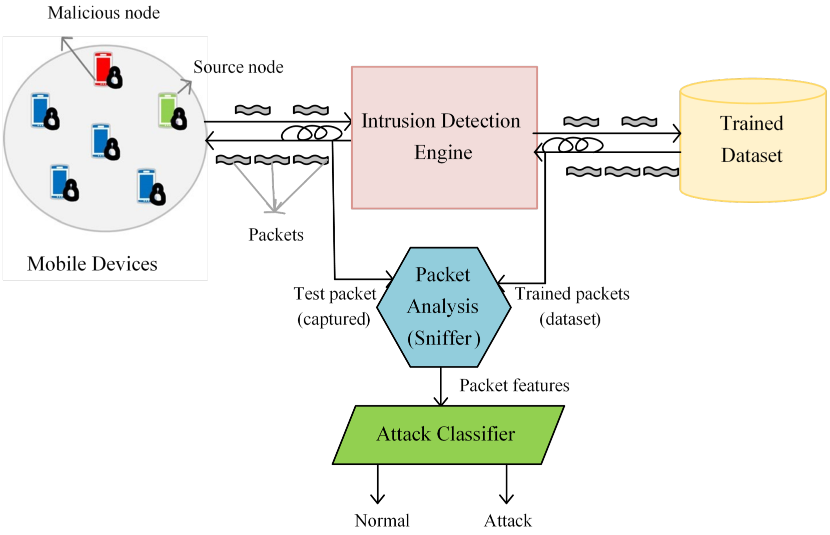


Figure 4.3: proposed Intrusion detection in MANET

1. input initial coordinate of nodes in the form of Xand Y.
2. simulate some nodes with malicious activities as blackhole attack that attracts packet and drops it and others as normal.
3. trace pcap file of each node at each stage of message transfer and receive.
4. export packet informations in required file.
5. select essential features.
6. data collection using selected features.
7. apply various ML techniques to classify normal and malicious information.
8. store outcome as a confusion matrix.
9. compute different statistical measures.
10. evaluate comparative results.

We have described details of simulation procedure such as essential feature selection, data collection process, statistical measures, and different ML techniques results in the next section. It is also shown simulation results and tabled comparative results of ML techniques.

Chapter 5

Result and Discussion

5.1 Experimental setup

A virtual environment is used to simulate the DDoS attacks source and target.To generate a DDoS attack and detect it using machine learning technology different tools and packages have been used which are described below.

5.1.1 Choice of Tools

Deciding on the correct tools to use is crucial for any project because it affects the issue of implementation and also the availability of other tools or libraries that will be required.

5.1.2 Programming Language

Python seemed the foremost appropriate language of choice for the project; its use is common for machine learning programs because of the provision of the many libraries, which makes implementing easier. Although it meant that we had to spend it slow at the beginning familiarizing myself with the language, it absolutely was the correct choice for this project.

5.1.3 CHOICE OF TOOLS

We used Collab and google drive to store and run all code. likewise as having the advantage of providing a backup of my code within the event of a hardware failure, it also allows you to revert to earlier versions just in case we make too many erroneous changes. Furthermore, all files were automatically saved to google drive. For reference, all testing was performed on my Toshiba satellite (core Intel i3, 8GB RAM) and google Collab.

5.1.4 Libraries

We used NumPy and Pandas when pre-processing the dataset. NumPy allows for multidimensional arrays and may perform fast mathematical operations on them. Pandas provide us with the info frame structure, allowing us to convert easily to and from CSV files. Both NumPy and Pandas operate efficiently on large data. For the educational phase of the project, we made a decision to use the Scikit-learn library, which is an open-source library that works with NumPy and Pandas. It offers a large range of tools and is straightforward to use, offering everything required for the training during this project.It had been not within the scope of this project to put in writing the algorithms from scratch, as we failed to require the fine-grained control gained from doing so. Hence, Scikit-learn's library was efficient and suitable to be used. Lastly, to retrieve information from the packets in my IDS, we used the Scapy library. The library has tools to be able to send, sniff, and dissect packets.

5.2 Dataset

We require a dataset to supply us with an input to the training algorithm, and further data to check the success of the model.When analyzing network traffic, you'll be able to either examine flow-level or packet-level data. The flow-level analysis provides an outline of activity on the network, where a flow may be a stream of knowledge between a source and destination employing a specific protocol. Packet-level data, on the opposite hand, captures the particular packets employed in the network layer, and can, therefore, provide more in-depth analysis since it obtained the particular payloads.We glance at flow-level data during this project. Although packet-level provides more data than flow-level, it'd likely be far overlarge to efficiently monitor, hence flow-level is more appropriate to be used here.We are using our local prepared dataset.It's a labeled dataset, containing nearly 17638 data points, 80 flow-based features, and 4 differing kinds of common attacks. This dataset was the result of a research paper [71] to make a useful, reliable dataset for attacks. It aimed to unravel the problems that the KDD99 dataset (and other accessible datasets) presented.We feel it had been an appropriate dataset to use since it absolutely was labeled and offered a large attack diversity across several different protocols. They decided which attacks to incorporate supported the foremost popular listed within the 2016 McAfee report, leading to relevant and up-to-date attack data.

5.3 Preprocessing

It allows the data extracted from the dataset to be trans-formed through a series of steps like deleting redundant records and normalizing data, to induce the information into the shape required for learning. the info for the dataset was collected over three days, and consists of 80% normal traffic, with the remaining 20% being the fourteen varieties of attacks. Table 5.1 shows the distribution of the various attacks during this dataset;

Label	Entries
Benign	8542
SSH-Bruteforce	3289
FTP-BruteForce	3200
Bot	1281
DDoS	1231

Table 5.1: Distribution of Labels in the dataset

it's clear that there's not sufficient data to coach for Heartbleed, SQL Injection, or Infiltration. So, we must drop these entries from the dataset.Furthermore, because the original attack labels provided, we grouped the remaining Eleven labels into more general categories using the mapping given in Table 5.2. we also assigned a binary (benign or attack) label toeach record. By evaluating each model on the three different grouping options, we are able to see if some varieties of attacks are more straightforward to predict than others.Or perhaps, if some models are better at predicting certain kinds of attacks than others, and whether grouping by the similarity of attack helps it to generalize well.Datasets occasionally contain missing values; this will occur from errors in recording or extracting the features To cater to missing values within the dataset, we made a decision to drop any rows that contained NaN, Null or Inf values, because the dataset is large enough this has almost no effect on the results.The subsequent step was to separate the dataset into training, validation, and testing datasets. The training dataset is what we use to coach the model; this is often the info that it learns from. Next, the validation dataset is what we use to perform initial testing and tuning of the model, while the test dataset is what we use to judge the performance after we've created the final

Botnet	Bot
Brute Force	FTP-Patator, SSH-Patator
DDoS	DDoS
DoS	DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS Slowloris
Probe	Port Scan
Web Attack	Web Attack: Brute Force, Web Attack: XSS

Table 5.2: Grouping of Original Attacks

model.We chose to separate the dataset employing a 60:20:20 ratio to training: validation: testing. Furthermore, because the dataset was unbalanced, we perform a stratified split of the labels. This ensures that the datasets produced to take care of the identical proportions of classes as within the original, as critical sampling, which splits the info randomly.Sampling allows us to avoid the case where sampling might not include enough instances of minority classes within the training dataset. It became apparent that there are some redundant columns which we then dropped from the dataset since they were providing no useful information towards classification. The last step in preprocessing is normalization. Normalization is important since the size of the feature values differs; some are between $[0,\infty]$ while others are between $[0; 1]$. So, by bringing all the features within the identical range, we make sure that they contribute an equal amount towards the classification. We performed min-max normalization using the scikit-learns library. Min-max normalization re-scales all of the features into the $[0,1]$ range, using the subsequent formula

$$\acute{x} = \frac{x - min(x)}{max(x) - min(x)} \tag{5.1}$$

where x is the original value

5.4 Development Approach

The project was split into three main areas of development; initial machine learning and evaluation, building the IDS interface, and final evaluation. We performed these tasks in sequential order, all of which used an iterative development approach. An iterative approach allows you to receive feedback about the system consistently. As an example, within the first section, an iterative workflow allowed me to incrementally make changes to my add an endeavor to enhance the performance of the classifier.

5.5 Starting Point

We had a theoretical understanding of Machine Learning from Part IB Machine Learning and Real-world Data, likewise as some general reading we had done. Other Tripos courses like Part IB Security and Part IB Computer Networking also provided useful background within the area. In terms of programming experience, we had some experience in Python and had used it for basic tasks, but never for an oversized project, machine learning, or during a networking context. Hence, we had to achieve an understanding of the libraries that we needed to use for the project.

5.5.1 Data Collection Phase

Data is usually required in machine learning to coach any algorithm to achieve knowledge. There are datasets available for network traffic classification like CAIDA dataset that's recorded in 2007. In CAIDA DDoS dataset, the author of the information doesn't guarantee that non-malicious data has been completely off from the dataset. Hence, to use that dataset, we might not get the most effective result since there's a break of including normal packets as DDoS packets. NSL-KDD is another dataset widely utilized by researchers. This dataset contains numerous attacks including six varieties of DDoS attack.The information is labeled with attack type and also normal traffic. However, authors haven't mentioned if the info is generated by IoT constrained devices. Therefore, we chose to come up with data supported own our requirement. To detect DDoS attack using machine learning technology, we would have liked DDoS and normal network traffic. We generated DDoS and normal traffic separately then combined them together. Normal and DDoS Traffic Collection To generate DDoS traffic, we've got used two Kali Linux running on Oracle VirtualBox machines on a laptop because the source and target of the attack. Both, attack source machine and victim machines are connected to MANET. Network traffic is recorded on victim machine using Wireshark. TCP SYN and UDP flood are generated using hping3 utility tool of Kali Linux. We've run DDoS attack roughly 1.5 minutes for every of the protocols and captured 17543 packets.To gather normal traffic, we've used 3 laptop devices that often interact for around 20 minutes and also one tablet and three itinerant for 30 minute. **Normal and DDoS Traffic Collection** To generate DDoS traffic, we have used two Kali Linux running on Oracle VirtualBox machines on a laptop as the source and target of the attack. Both, attack source machine and victim machines are connected to MANET. Network traffic is recorded on victim machine using Wireshark. TCP SYN and UDP flood are generated using hping3 utility tool of Kali Linux. We have run DDoS attack roughly 1.5 minutes for each of the protocols and captured 17543 packets. To collect normal traffic, we have used 3 laptop devices that regularly interact for around 20 minutes and also one tablet and three mobile phone for 30 minute.

This time, we use the entire dataset to train the classifier since we don't must hold out any for testing and validation. Once training of the model is complete, the system starts analyzing the incoming traffic packet by packet. This traffic may either be live, using the machine's network interface, or from a supplied. Pcap file. A .pcap file contains packet data created during network capture and allows us to research previously collected data.

5.6 Identifying a Flow

A flow is identified by its flow ID, given by the subsequent tuple: (source IP, destination IP, source port, destination port, protocol) the primary packet within the flow is assumed to be within the forward direction, and then defines the source and destination directions, making a future packet with these values switched a packet within the backward direction. There are only two transport layer protocols that we consider here; UDP and TCP. A UDP flow terminates on day trip, given here as 600 seconds. A TCP flow, yet as day out, may additionally terminate via the standard connection raze (FIN flag), or reset (RST flag). When a replacement packet is received, we've to test whether it belongs to an existing flow, and if not, create a brand new flow instance. If the packet belongs to an existing flow, we must check if it absolutely was a terminating packet. We store all current flows in an exceedingly dictionary, with the keys being the flow-ID, and also the value is that the regard to the flow object.

5.6.1 How to Launch different attack on MANET

hping3is pre-installed package on Kali Linux. It is a command-line based packet analyzer. It can be used for Firewall testing, advanced port scanning, network testing using different Internet protocols, advanced traceroute, TCP/IP stacks auditing, etc. With hping3 options users can specify the target server, a number of packets to send to the target, target port, spoofing attack source,selecting a random source, random destination, flooding to send requests to the target as fast as possible, protocol types such as TCP, UDP, ICMP and many more options. We have used hping3 to launch UDP and TCP flood on the server running on another machine. **Attack Parameters** To generate a DDoS attack with hping3, the following parameters were used.

- -flood: This command sends packets as fast possible.
- -rand-source: This command spoofs the attack source with random IP address
- -c -count: This command represents packet counts
- -d -data: Using -d we have set the packet size to send to the victim server

Bot We use Ares1, which may be a Python-based botnet and backdoor. It offers tools like keylogging, remote shell, and file transfers. We host the command-and-control server on our Kali attacker machine and launch the agent on our compromised victim.

DoS and DDoS Four differing kinds of DoS attacks exist during this dataset. To perform GoldenEye, Hulk and Slowloris, and run the scripts on the attacker machine, providing the victim's URL as a parameter.The ultimate DoS attack, Slowhttptest is preinstalled in Kali, and as before, given the target URL to launch the attack. A Distributed DoS (DDoS) attack requires multiple devices (bots) to be ready to overwhelm the target machines. As we'd not personally have permission to use enough devices to form the quantity of traffic required for a DDoS attack, but we try with eight devices connected through the MANET network and generate different attacks, using Wireshark for sniff packets and convert to CSV file using CICFlowMeter-4.0. **FTP and SSH Patator** are both brute force attacks on a server, within which we make many repeated guesses at a password to achieve unauthorized access to the system. Here, we use the Patator package to perform both attacks. To perform brute force, the program is supplied with a dictionary of passwords to try; we use the rock your password dictionary (Preinstalled on Kali) which contains over 14 million unique passwords. We run both FTP and SSH servers on the victim machine, and so launch the attack from the attackers' machine. **Probe** A probe attack aims to find weaknesses or vulnerabilities during a system.The probe the attack we perform uses Nmap (Network Mapper) to perform a port scan on the victim machine.

Web Attack To perform the assorted web attacks, we host the Damn Vulnerable Web App (DVWA) on our victim machines. DVWA may be a PHP/MySQL-based web application that's susceptible to many common web attacks. To perform brute force, we first use Burp Suite offered by Kali to intercept the HTTP requests so we are able to see the shape they take, such as the login parameters and therefore the cookies. Once we've this data, we use THC-Hydra to brute force the login. We use XSSer to perform cross-site scripting attacks. XSSer could be a framework that may detect and exploit web vulnerabilities. We pass it the URL for the DVWA on our victim machine, together with the desired cookies, and also the program automates the testing for XSS. **Scapy** is a powerful Python-based packet manipulation program. Scapy allows the user to transmit, forge, dissect and sniff the network packets. It is also used for tracerouting, scanning, attack, probing, unit testing and network discovery purposes. Using Scapy a user can send invalid frames, inject their own frames and combine different parts of the packets. Scapy allows the user to specify his own packet or set of packets, layers and you can set packet field and values based on your requirement. We have used Scapy to generate GTP-U packet and glow GPT-U packet with the inner packet that contains the DDoS attack.

Scikit-Learn Scikit-learn6 is an open source machine learning tool for Python programming languages. It is an efficient and simple tool for data mining and data analysis. Scikit-learn contains the implementation of different algorithms for supervised and unsupervised learning. In addition to classification,regression and clustering algorithm, this package also contains features for model selection, dimensionality reduction and data preprocessing Scikit-learn has extensive use and is being used by different researchers and big industries like Spotify, booking.com, change.org and IBM Watson to integrate the machine learning module into their platform. The reason why many industries and researchers are selecting Scikit-learn in their artificial intelligence and machine learning tool is its ease of use that allows accomplishing plenty of processes with a collaborative library, open API with proper documentation and free of cost. The mentioned classes in the table below have been used from Scikit-learn library for classification.

Python is a high-level and general-purpose open source programming language. Due to the code readability philosophy of Python, ease of learning, efficient code and easy communication feature of Python it has been the favorite programming language for the majority of data scientists. Python has vibrant scientific libraries and many great environments such as Spyder and Jupyter notebook. Python Matplotlib library is a powerful 2D-graphic library that helps machine learning scientists in plotting graphs. Due to these strengths, we have chosen Python languages for machine learning experiment. Due to these strengths we have chosen Python languages for machine learning experiment.

5.7 Feature Selection

Feature Selection Feature selection may be a process where you automatically select those features in your data that contribute most to the prediction output variable. Having irrelevant features in your data can decrease the accuracy of the many models, especially linear algorithms like linear and logistic regression. Three benefits of performing feature selection before modeling your data are: Reduces Overfitting: Less redundant data means less opportunity to form decisions supported noise. Improves Accuracy: Less misleading data means modeling accuracy improves. Reduces Training Time: fewer data means algorithms train faster. feature selection allows us to cut back the dimensionality of our data, and choose the features which are relevant to the classification. We apply the Chi-Squared test to the training data which tests the independence of a feature variable with the category label. If the feature variable and sophistication label are independent, then the feature variable isn't relevant to decide the category label, then we will discard it. The SelectKBest function from the scikit-learn library allows us to use the chi-squared algorithm, and so select the highest 'K ' most dependent features for our dataset. Figure 5.1 shows the cumulative scores for the sorted feature scores produced by the libraries chi-squared algorithm. The line represents 99%, meaning that the features which lie above that time are adding little information, then we are able to remove them. This gives us a worth of k=40 to supply to the SelectKBest function. Table 4.3 lists the 40 chosen features, with an outline of every. Most of the features chosen relate to either the statistics of the packet length sizes, the flags utilized in the flow (if TCP), the inter-arrival time between packets, or the number of your time the flow spent idle.

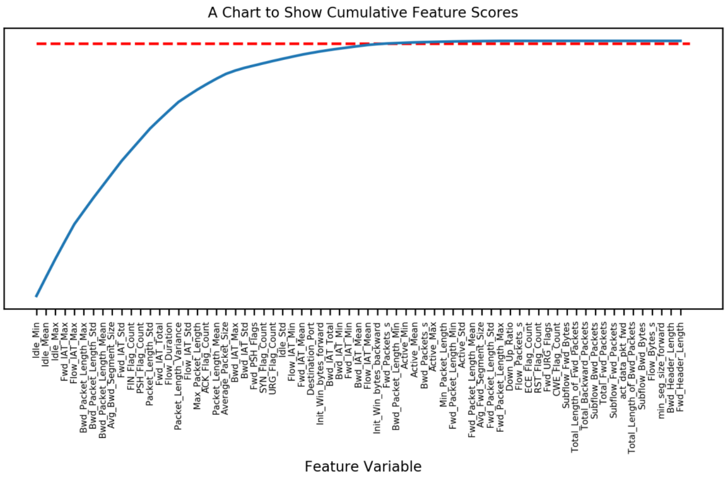


Figure 5.1: shows the cumulative scores for the sorted

5.8 Classification

We have preprocessed the information and transformed it into the chosen features. Now, we can fit our training data to the classifier.

5.8.1 Implemented Models

For the chosen five learning algorithms we've got implemented, we provide a quick reason on why we believed they were suitable to be used during this research. **Support Vector Machine** We chose to implement an SVM since a paper referenced in my project proposals starting point [72] evaluated network intrusion detection using an SVM and achieved a detection accuracy of 88.03% on the UNSW-NB15 [73] dataset. SVMs create a choice boundary to discriminate between the classes, then will divide the hyperplane into the various benign and attack class labels.

Decision Tree The decision tree could be a popular choice for classification problems and is why we made a decision it was appropriate to use here.They're easy to implement and might work with large data. However, they will suffer from overfitting.To make our decision tree, we fit the dataset, which creates decisions to be made at each node.This can be performed by randomly splitting the dataset and assessing the split using the Gini impurity. Formally, the Gini impurity measures the frequency of which a randomly chosen element from this split should be incorrectly labeled if it were randomly labeled consistent with the distribution of labels during this split. [74] Gini impurity ranges between zero, when all elements belong to the same class, and reaches the utmost value when the values are equally distributed across all of the classes. We aim to reduce the Gini impurity, such this splits perfectly separates the category. **Naive Bayes** Naive Bayes may be a simple and commonly applied machine learning algorithm.We are trying to classify network traffic as normal or malicious, so it seems appropriate to undertake a Bayes classifier.Naive Bayes makes the naive assumption that the features are independent of every other. However, this might not hold with the set of features chosen, and then might lead to poor results. **K Nearest Neighbor** K-NN classifies employing a majority voting on nearby points, so if we assume that similar types of attack feature vectors are nearby to every other, it should produce an honest performance. Also, no training time is needed; instead, it stores the training data in memory to be used at test time. However, this implies it's computationally intensive.

K Means Clustering Clustering allows us to divide the info into similar groupings, specified data points within the same cluster is comparable. Hence, by implementing clustering, we aim to take advantage of the differences between benign and adverse traffic within the hope that the algorithm assigns them to different clusters. However, unlike all of the supervised methods we've previously discussed, K means clustering is unsupervised, meaning that there aren't any labels for it to predict. As a result, we've lost the notion of what's a 'correct' result. Instead, we can try to map the clusters to the various label groupings to determine if there's a correlation.

5.8.2 Implementing Classification

For each of the algorithms mentioned, we train and evaluate them thrice, each with the different label classes (original labels, grouped labels, and binary labels) and compare how each performs using metrics.The primary three models were straightforward to implement, while the last two required choosing the K parameter.To settle on the worth of K for the K-NN, we are able to evaluate the algorithm on the validation dataset across different values of K, and see which performs. Figure 5.2: KNN Precision, Recall,

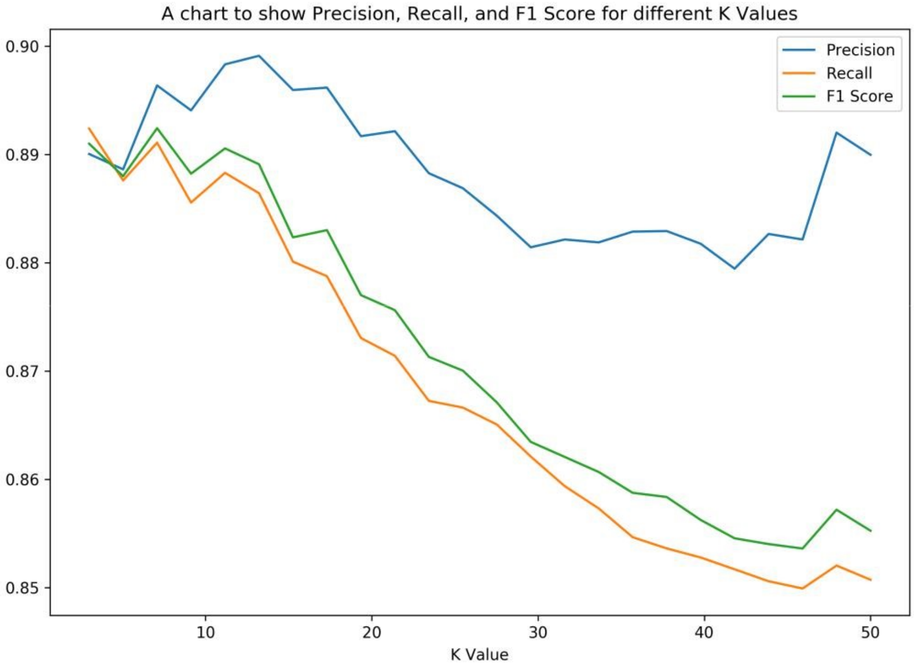


Figure 5.2: KNN Precision, Recall and F1 Score

and F1 Score for various Values of K, of the initial Attack Labels on the Validation Dataset. best. Figure 5.2 shows the precision, recall, and F1 scores for odd K values up to 50. K is odd to avoid ties when deciding the category. After gazing the graph, we commit to set K=7. Although precision is highest when K=13, we also need a high recall, therefore we accompany the best F1 score, which is when K is ready to 7. Deciding the worth of K for K means cluster was more natural since the matter predefined it; the amount of clusters is that the same because the number of labels within the dataset since we would like the clusters to represent the various classification labels. Principal component analysis (PCA) will be accustomed visualize the output of the K means cluster algorithm. PCA transforms high-dimensional data into lower dimensions, allowing us to work out the clusters easier [73].Finally we implement the hybrid method.

5.9 Results

5.9.1 Evaluation Metrics

We use metrics to evaluate the standard of the various machine learning algorithms that we've presented. We explore the suitable metrics for this thesis. A confusion matrix allows us to visualize the performance of the classification algorithm; it's a table whose results are divided into actual and predicted classes. [75] Table 5.3 shows a confusion matrix for a binary classification algorithm. **Accuracy** is the number correct classifications, divided by the total number of classifications.

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5.2}$$

Although this may seem like a simple metric to use, it can fail to describe the true performance if we have highly imbalanced classes. We define precision as the ratio of the number of class predictions that truly were that class, over the number that was predicted that class.

Table 5.3: Confusion Matrix for Binary Classification

	Predicted Attack	Predicted Benign
Truly Attack	True Positive (*TP*)	False Negative (*FN*)
Truly Benign	False Positive (*FP*)	True Negative (*TN*)

$$Precision = \frac{TP}{TP + FP} \tag{5.3}$$

A low precision indicates that we have a large number of false positives in our results. Classifying truly benign data as an attack too often, the false-positive case, can render the system unusable, a situation we want to avoid. Recall is the ratio of the number of class predictions that truly were that class, over the number of true occurrences of that class.

$$Recall = \frac{TP}{TP + FN} \tag{5.4}$$

A low recall indicates that we have a large number of false negatives in our results. As previously mentioned, we wish to avoid classifying attacks as benign; this is the false negative case and would lead to undetected network attacks. So, we must ensure the recall value of our system is as close to 100% as possible to avoid this. The F1 score allows us to combine the precision and recall values, and is their harmonic mean.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recal} \tag{5.5}$$

By combining both the precision and recall into a single metric, it provides a balance between the two values. Since precision and recall values are important to consider in this context, we use the F1 score to evaluate the performance of our classifier. The previous chapter described the preprocessing phase of our machine learning experiment and this chapter would describe the results regarding classification algorithms (KNN, Decision Tree, Naïve Bayes,K Means Clustering, K Nearest Neighbor's) that were used to detect DDoS attacks in a packet core network. The collected data were trained to predict DDos, Bot, SSH-Bruteforce, and FTP-BruteForce using the algorithms, and the performance of each of the algorithms was evaluated separately, and the results were stored and shown in the tables.

5.9.2 Requirements Analysis

First, We prepare the dataset, making sure it is the right shape and there are no null values. Next we perform feature selection, using a filter approach. Lastly, we then train and test the different models.We perform initial testing on the proportion of the dataset held out for validation. We can split the evaluation of the algorithms into the supervised approaches and the unsupervised approach since they have different evaluation approaches. The success criteria of this project were defined

- Implemented a minimum of one machine learning algorithm and evaluated its precision on a proportion of the dataset reserved for testing.
- Successfully developed an interface for the intrusion detection system, which might sniff and monitor traffic and alert on a suspected attack.
- Evaluated the IDS detection accuracy when tested on simulated network traffic.

Using these, we are able to construct the tasks that require to be completed, Table 5.4 shows these alongside their priority, difficulty, and risk class. Priority rankings were assigned supported how important the step was in terms of constructing progress (i.e. Do future tasks depend upon this step).The issue was assigned supported whether we currently knew the way to perform the step. Deciding the chance level was a mix of current knowledge on a way to complete the task, and the way important the task was to the general success criteria.

No.	Task	Priority	Difficulty	Risk
1	Find a suitable dataset	High	Medium	High
2	Preprocess the dataset	Medium	Low	Low
3	Perform feature selection on the dataset	Medium	Medium	Low
4	Perform the chosen five machine learning algorithms on the data, andevaluate using the validation set	High	High	Medium
5	Compare the different models to create the final model	Medium	Low	Low
6	Evaluate the final performance using the test set	Medium	Low	High
7	Build a system that can extract the chosen features from real-time traffic	High	High	High
8	Combine the machine learning model from task 5 and system from task to create the intrusion detection system	Medium	High	Medium
9	Evaluate the performance of the IDS created in task 8 by simulating avariety of network attacks	Medium	High	High

Table 5.4: Requirements for my Project to be a Success

5.10 Methodology

The datasets were generated and collected as described in detail in section 5.1 Both normal datasets and DDoS,Bot, SSH-Bruteforce, and FTPBruteForce datasets are labeled, and the required features are extracted. Then the data is converted into the format which is acceptable for Scikit-learn using CICFlowMeter-4.0. Several experiments have been performed to check the accuracy and performance of the classifier on different data combinations and sizes. It should be noted that our DDoS,Bot, SSH-Bruteforce, and FTPBruteForce detection experiment is based on only TCP SYN attack and UDP protocol attack as a sample since the thesis aim is to provide only a DDoS,Bot, SSH-Bruteforce, and FTPBruteForce detection method. The data is split to train set,validation, and test set as follows: We have followed the below procedure throughout our machine learning-based intrusion detection experiment:

1. Reading the datasets
2. Selecting the features and the target
3. Splitting the dataset into train set,validation and test set
4. Training the model with the classifier
5. Predicting the data coming from the test set
6. Checking accuracy of the classifier prediction through 10-fold cross-validation

5.11 Initial Testing Results

We are perform, and analyse the following 5 machine learning algorithms on the local Dataset:

- Support Vector Machine (SVM)
- Decision Tree
- Naive Bayes
- K Means Clustering
- K Nearest Neighbours
- Hybrid model

To evaluate the performance we perform the three group mechanism of each algorithm

1. All labels:

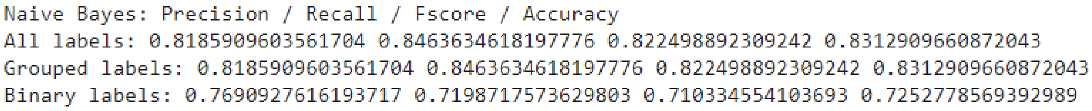
2. Grouped labels

3. Binary labels:

5.11.1 Supervised Algorithms

Naive Bayes

For the chosen supervised algorithms, we can compare their metrics directly.Figure 5.3 shows the precision, recall and accuracy that the supervised algorithms achieved. This graph reveals why accuracy itself does not provide enough information about the performance of an algorithm; naive Bayes on the original labels achieved an accuracy of 0.83, yet had a recall of 0.84. This recall and accuracy value would mean that the true positive attacks and true negative more, which indicates that naive Bayes is not suitable for use in an IDS for Manet.The graph shows us that precision and recall rise as the attack groupings get less specific,giving the binary case the highest performance. However, we want to be able to provide more information than just ‘attack’; hence there is a trade-off between the specific class provided and the result.We can see that naive Bayes has the worst performance rates of any algorithm across all the three different groupings. The reason that naive Bayes likely performed so bad in comparison due to the other algorithms is due to the naive assumption we made; we assumed that all the features are independent, which is not necessarily valid in this case.For example, Packet Length Variance and Packet Length Std are features that are not independent. Hence the assumption fails, and we achieve poor results.



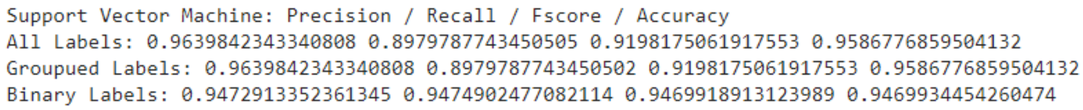
```
Naive Bayes: Precision / Recall / Fscore / Accuracy
All labels: 0.8185909603561704 0.8463634618197776 0.822498892309242 0.8312909660872043
Grouped labels: 0.8185909603561704 0.8463634618197776 0.822498892309242 0.8312909660872043
Binary labels: 0.7690927616193717 0.7198717573629803 0.710334554103693 0.7252778569392989
```

Figure 5.3: Naive Bayes Precision, Recall, F1 Score and accuracy

SUPPORT VECTOR MACHINE

The SVM is already referred to as the most effective learning algorithm for binary classification. The SVM, originally a kind of pattern classifier supported a statistical learning technique for classification and regression with a spread of kernel functions, has been successfully applied to variety of pattern recognition applications. Recently, it's also been applied to information security for intrusion detection. Support Vector Machine has become one among the favored techniques for anomaly intrusion detection thanks to their good generalization nature and therefore the ability to beat the curse of dimensionality. Another positive aspect of SVM is that it's useful for locating a world minimum of the particular risk using structural risk minimization, since it can generalize well with kernel tricks even in high-dimensional spaces under little training sample conditions. The SVM can select appropriate setup parameters because it doesn't rely on traditional empirical risk like neural networks. one among the most advantages of using SVM for IDS is its speed, because the capability of detecting intrusions in real-time is incredibly important. SVMs can learn a bigger set of patterns and be ready to scale better, because the classification complexity doesn't rely upon the dimensionality of the feature space. SVMs even have the flexibility to update the training patterns dynamically whenever there's a replacement pattern during classification.

Limitation of Support Vector Machine in IDS SVM is essentially supervised machine learning method designed for binary classification. Using SVM in IDS domain has some limitation. SVM being a supervised machine learning method requires labelled information for efficient learning. Pre existing knowledge is required for classification which can not be available all the time. SVM has the intrinsic structural limitation of the binary classifier i.e. it can only handle binary-class classification whereas intrusion detection requires multi-class classification. Although there are some improvements, the quantity of dimensions still affects the performance of SVM-based classifier. SVM treats every feature of knowledge equally. In real intrusion detection datasets, many features are redundant or shorter. it might be better if feature weights during SVM training are considered. Training of SVM is time-consuming for IDS domain and requires large dataset storage. Thus SVM is computationally expensive for resource-limited spontanepous network. Moreover, SVM requires the processing of raw features for classification which increases the architecture complexity and reduces the accuracy of detecting intrusion. **SVM** was chosen for consideration due to its use in a paper [**?**] found which achieved a detection accuracy 96.39% on the 5 different labels, hence outperforming their classifier. Support Vector Machine (SVM):Finally all the result look like as follow in fig

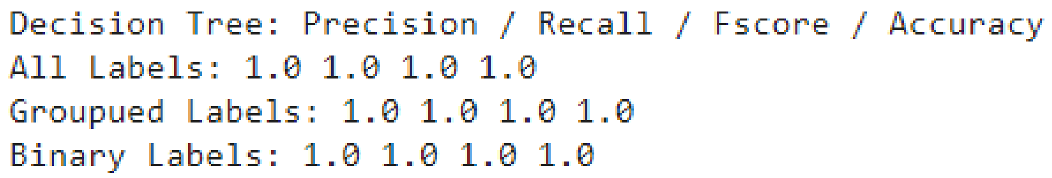


```
Support Vector Machine: Precision / Recall / Fscore / Accuracy
All Labels: 0.9639842343340808 0.8979787743450505 0.9198175061917553 0.9586776859504132
Grouped Labels: 0.9639842343340808 0.8979787743450502 0.9198175061917553 0.9586776859504132
Binary Labels: 0.9472913352361345 0.9474902477082114 0.9469918913123989 0.9469934454260474
```

Figure 5.4: finally all the SVM results

Decision Tree

Decision Tree (DT) algorithms are well-known tool for classification and prediction tasks.It builds a model that predicts the output of a pattern based on different input attribute values of the pattern. The construction of DT does not require any domain knowledge or parameter setting, just the given data set is learnt and modelled. It consists of three basic elements: Decision node, edges or branch and leaf node From the graph, it is clear that the decision tree and K-NN classifiers are suitable to use within the IDS since they achieve high precision and recall.Finally all the result look like as follow in fig

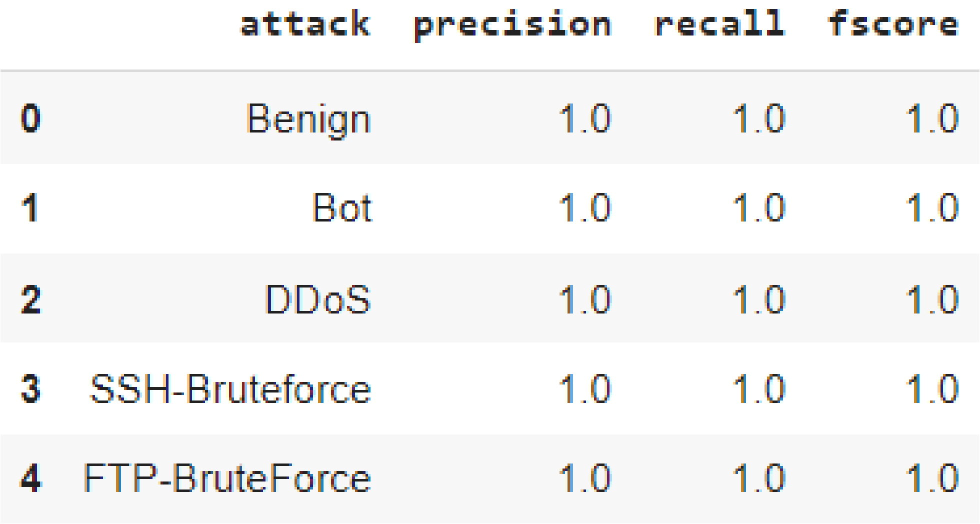


```
Decision Tree: Precision / Recall / Fscore / Accuracy
All Labels: 1.0 1.0 1.0 1.0
Groupued Labels: 1.0 1.0 1.0 1.0
Binary Labels: 1.0 1.0 1.0 1.0
```

Figure 5.5: finally all the Decision Tree results

Random Forest

The random forest is a collection of decision trees, building each tree by randomly sampling the features and the training data, and using majority voting amongst the trees to assign the class label. As a result, they tend to outperform decision trees since they overcome the overfitting problem that decision trees may suffer from and are less sensitive to outlier data.



	attack	precision	recall	fscore
0	Benign	1.0	1.0	1.0
1	Bot	1.0	1.0	1.0
2	DDoS	1.0	1.0	1.0
3	SSH-Bruteforce	1.0	1.0	1.0
4	FTP-BruteForce	1.0	1.0	1.0

Figure 5.6: Random Forest

5.11.2 Unsupervised Algorithms

Unsupervised methods lack the well-defined evaluation metrics that supervised methods have due to the lack of correct labels provided to assess the results. However, as we are working with a labelled dataset, we can see if the clusters obtained from the K means clustering algorithm match the attack labels. Also, we can use principal component analysis (PCA) to compare the output of the clustering algorithm visually. Figure 5.8 shows the result of PCA on both the actual labels and the predicted clusters on the validation data. It is clear from these graphs that our prediction lacks the correct structure that the actual labels hold, indicating poor performance.

In the supervised models, we saw the trend that the grouped labels usually performed better than the original labels, with the binary labels performing best. However, the grouped labels also lack significant structure when compared with the correct labelling. The results reveal that cluster 0 is likely to be benign data and cluster 1 DoS attacks. The remaining attacks all seem to belong to cluster 2, alongside a fifth of the benign data. Overall, there is no clear result from the original labelled data, nor the grouped labels.Lastly, the binary classification shows no signs of improvement either.The results are given in Figure 5.6 We conclude here that K means clustering is not suitable for use as it has not produced any promising results.

k-nearest neighbor algorithm (k-NN)

k- nearest neighbor algorithm (k-NN): This is the most simple among all machine learning algorithms where the output is calculated based on k closest neighbours or k training

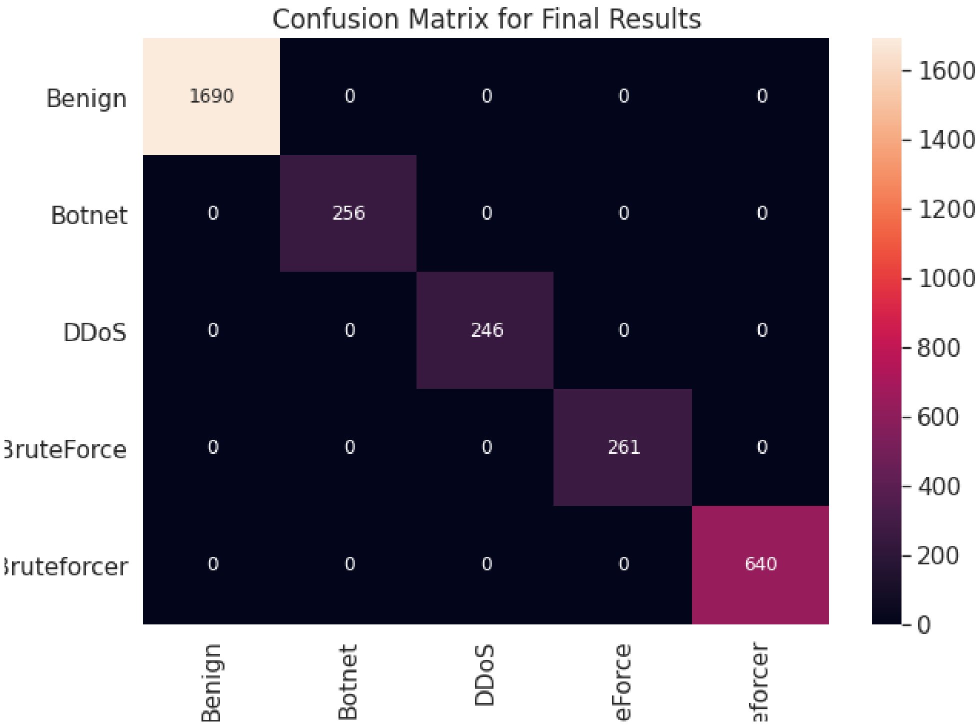


Figure 5.7: Random Forest

patterns .The calculation of output varies depending on the task to be performed. For example in case of classifying an unknown pattern, the pattern is assigned as the class which appears frequently among the k nearest training patterns.Finally all the result look like as follow in fig

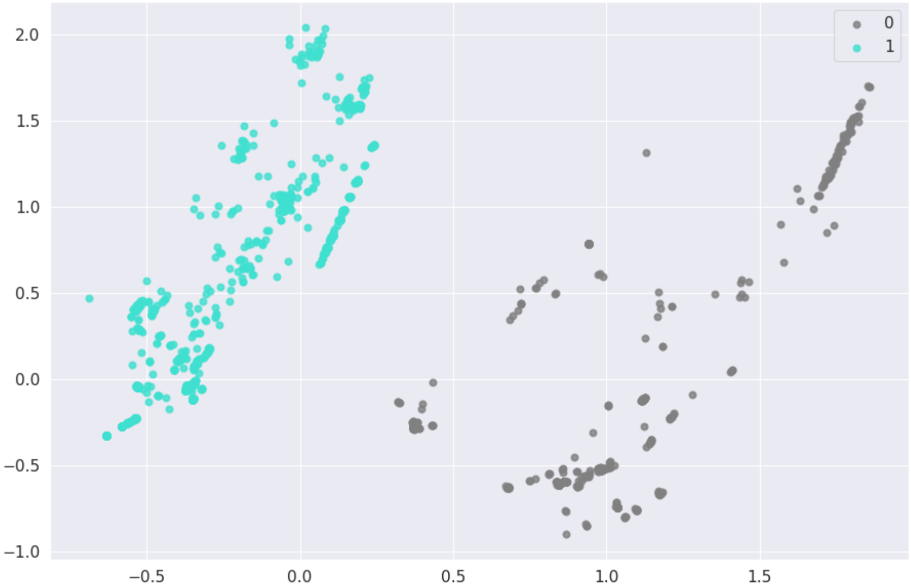
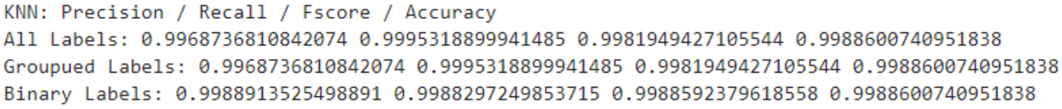


Figure 5.8: Finally all the PCA algorithm results

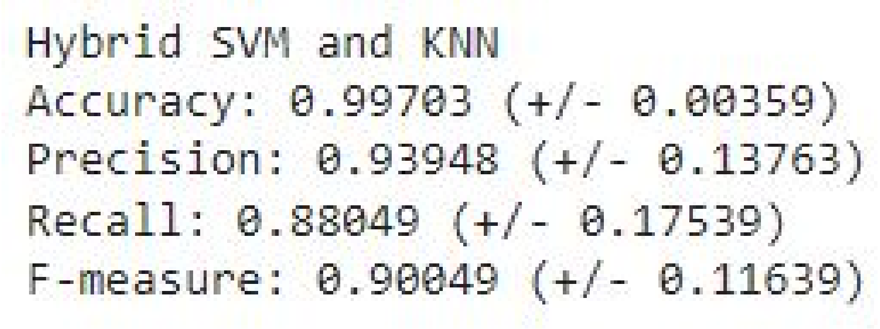


```
KNN: Precision / Recall / Fscore / Accuracy
All Labels: 0.9968736810842074 0.9995318899941485 0.9981949427105544 0.9988600740951838
Groupued Labels: 0.9968736810842074 0.9995318899941485 0.9981949427105544 0.9988600740951838
Binary Labels: 0.9988913525498891 0.9988297249853715 0.9988592379618558 0.9988600740951838
```

Figure 5.9: Finally all the k-nearest neighbor algorithm results

Hybrid machine learning model

The EML method creates multiple instances of traditional ML methods and combines them to evolve a single optimal solution to a problem. This approach is capable of producing better predictive models compared to the traditional approach. The top reasons to employ the EML method include situations where there are uncertainties in data representation, solution objectives, modeling techniques, or the existence of random initial seeds in a model. The instances or candidate methods are called base learners. Each base learner works independently as a traditional ML method, and the eventual results are combined to produce a single robust output. The combination could be done using any of the averaging (simple or weighted) methods and voting (majority or weighted) for regression and classification methods, respectively.To compare its performance with other popular machine learning algorithms such as Random Forest, SVM, KNN and other machine learning methods SVM score the accuracy 95% and KNN scores 99.98%. The experimental results show that ensemble learning is a proper technique for classifying attacks than other existing methods in term of Accuracy, Precision, Recall and F-score also, the proposed system shows better accuracy compare to Random Forest and even the two SVM and KNN individually.

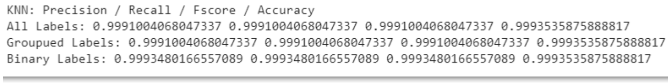


```
Hybrid SVM and KNN
Accuracy: 0.99703 (+/- 0.00359)
Precision: 0.93948 (+/- 0.13763)
Recall: 0.88049 (+/- 0.17539)
F-measure: 0.90049 (+/- 0.11639)
```

Figure 5.10: Finally hybrid model algorithm results

5.12 Model Refinement

After initial analysis of the five different implemented models, we now begin to look more in depth at the well performing ones, allowing us to finalise details of our model. **Decision Tree vs K Nearest Neighbours** Due to the success of decision trees and K-NN in the previous subsections, we now compare the two algorithms further. Firstly, we can look at how they perform in classifying the attacks individually. Figure 5.7 shows the F1 scores for both decision tree and Figure 5.8 K-NN on each of the different labels. Decision tree offers a higher F1 score than K-NN in all the attack labels other than brute force and DoS slowloris, meaning that decision tree is the best performing algorithm out of all the five considered so far. Cross-site scripting (XSS) performs very poorly for both the algorithms, while bot and brute force also perform slightly worse than the rest of the labels. Using decision tree and KNN to achieve a accuracy of 99.99% and 99.99% respectively. They extracted URL based features, such as the presence of special characters and request for cookies, as well as JavaScript-based features like the number of script functions and the number of references to a JavaScript file for example. Hence, it could be possible to achieve better results for XSS if we instead had different features. As well as the evaluation metrics, we also need to compare the time taken for training and classification. The IDS classifies in real-time hence the classification time must be fast. On the other hand, training time is less important since we can train the system before it goes live, although the user sees this as wasted time.The training and classification time for both decision tree and K-NN. We see that decision tree takes far less time in both cases. The large difference in times arises because the K-NN algorithm has a query complexity of O(dn2) while the decision tree is O(log n), for n samples in d dimensions feasible. K Nearest Neighbours



```
KNN: Precision / Recall / Fscore / Accuracy
All Labels: 0.9991004068047337 0.9991004068047337 0.9991004068047337 0.999353587588817
Groupued Labels: 0.9991004068047337 0.9991004068047337 0.9991004068047337 0.999353587588817
Binary Labels: 0.9993480166557089 0.9993480166557089 0.9993480166557089 0.999353587588817
```

Figure 5.11: KNN

Decision Tree vs Random Forest From the five different algorithms we have tested, it is apparent that the most suitable for use on the chosen dataset and the selected attacks is the decision tree,KNN, Kmean. Due to its success, we suggest considering a random forest classifier. The random forest is a collection of decision trees, building each tree by randomly sampling the features and the training data, and using majority voting amongst the trees to assign the class label. As a result, they tend to outperform decision trees since they overcome the overfitting problem that decision trees may suffer from and are less sensitive to outlier data. the original labels on the validation dataset. We can see that they achieve similar results,which we would expect given that the random forest is a collection of decision trees. The random forest has a higher F1 score than the decision tree in 7 out of the 12 labels for the graph presented. It also had higher F1 scores in 5 out of the 7 grouped labels, and in both of the binary labels.As before, we compare the training and classification times for both algorithms, Unlike the K-NN case, they are now in the same order of magnitude. We see that the random forest trained slightly faster, while the decision tree had a faster classification. Nonetheless, these times are not different enough to significantly advantage either model. Overall, the random forest was the most appropriate model to use going forward due to the results it achieved on the validation data. Unfortunately, none of the attacks performed better at classifying the labels where it offered lower results (bot, brute force, DDos). If that was the case, we could have combined the models with having a hybrid model and implementing a voting rule to classify the traffic.

Final Grouping of Labels We need to consider what labels we want our classifier to predict. Throughout the evaluation in the last sections, we considered three different options; the original 12 from the dataset, our custom 7 which grouped the 12 labels by type of attack, or the binary label case. We saw that in general, the binary case achieved the highest F1 scores and accuracy. However, We rule this out as a potential option due to wanting to make the system provide a useful output. We want to be able to tell the user what type of attack occurred, rather than just ‘attack’. As a reminder to the reader, Despite the low F1 scores brute force achieved in the last subsection for the random fores and Naive Bayes when they are combined into the web attack label, it achieves an F1 score of 98.83%, compared to the previous 73.60% and 31.22% respectively. A Similarly, 69 of the 301 brute force attacks are incorrectly classed as XSS. Hence, when grouped under one label, these errors go away, and we correctly classify 424 of the total 431 web attack labels. This further hints that we have not selected features specific enough to identify web attacks; we have enough to know that a web type attack occurred but do not have enough information to differentiate on type of attack.Although the four different types of DoS attacks already performed well, having scores above 99% for precision and recall, we want to group these into one category. Since there exist many different types of DoS attack, it makes sense to generalise to that case, rather than fit new ones into one of the four we have trained on. I initially proposed grouping FTP-Patator and SSH-Patator into one label since they are both types of brute force.

However, they both performed exceptionally well on the random forest classifier when considered individually, achieving F1 scores of 99.99% and 99.99% respectively.

When looking at the confusion matrix, the only miss-classification either attack had was to a benign label, so we do not group these in the final grouping, and instead, consider them as two separate cases. There are no other relevant or credible groupings to consider, and so we have decided upon the final grouping of labels. Optimising Parameters Machine learning models also take a collection of parameters as input, these values are known as hyperparameters and control the behaviour of the model (such as the depth of the tree in decision trees and random forests). We can tune these hyperparameters to find the values which yield the optimal results for our problem. For our model, a random forest, listed are the parameters which we will tune, and a description of them according to the scikit-learn library:

- n estimators: The number of trees in the forest
- max depth: The maximum depth of the tree
- min samples split: The minimum number of samples required to split a decision node
- min samples leaf: The minimum number of samples required to be a leaf node
- max features: The number of features to consider when looking for the best split
- bootstrap: Whether bootstrap samples are used when building trees.

If false, the whole dataset is used to build each tree First, we consider the number of estimators parameter. This controls the number of decision trees that we evaluate in our ensemble. We speculate the reasoning for this misclassification is due to the different operational phases of a botnet. Botnets communicate with a command and control server, which can launch a variety of commands. Hence it is likely that our algorithm has learnt to detect botnet when malicious network traffic is produced due to this botnet operation (such as being used in a DDoS attack). However, bots may also sit idle, or the adversary can limit the intensity of the attacks to disguise as 'normal' traffic, and this is probably where our model is classifying botnet traffic as benign.Table 4.7 compares the results of the testing with the percentage of the dataset that label occupied. We see the two lowest scoring labels, botnet and web attack, each account for less than 0.1% of the dataset. The lack of data for these two attack labels may be why they are the only attacks to have an F1 score less than 99The benign case had a precision of 99.97%. The precision measures the false positives, which in the case of the IDS would be benign traffic classified as an attack.

The high precision for the benign case shows that the system is usable and will not classify all the users normal data as attacks. Similarly, the benign case achieved a recall of 99.91%. The recall measured the false negatives, the case where we classify attacks as benign. This is the situation we wanted to avoid since it means we have not identified a network attack and is a big problem. However, the 99.91% recall leaves us confident in classifying traffic appropriately.The range of accuracy achieved was between 88.0% and 99.9%, with the median accuracy being 97.2%. We exceeded this median value, achieving an accuracy of 99.9%, matching that of the highest accuracy in the survey. Unfortunately,the survey does not include precision, recall or F1 metrics so we cannot compare our values. Overall, we can be confident that our system can detect the types of attacks it was trained on. An average precision of 99.98% and recall of 99.98% ensures that our system can differentiate between benign and adverse traffic with minimal errors.The decision tree and K-NN classifiers are suitable to use within the IDS since they achieve high precision and recall.

Chapter 6

Conclusion and Future Work

This thesis addresses two main questions described in chapter one and also it provides the results that fulfill the thesis objectives. In addressing the primary research question, the task was to look into the MANET security challenges proper attribute selection for unknown attack. To achieve this objective, comprehensive research was carried out on security threats in the mobile adhoc network. Also, a detailed study was carried out on adhoc network protocol vulnerabilities that are the threat that the majority of the attackers can exploit to launch an attack. To address the second research question and achieve the thesis objectives,real time dataset was prepared and apply machine learning techniques were used to detect DDos, Bot, SSH-Bruteforce, and FTP-BruteForce attacks generated by insecure MANET devices.To detect DDos, Bot, SSH-Bruteforce, and FTP-BruteForc attacks, we proposed to perform a machine learning to classify attack. Then, through a supervised machine learning technique attack detection has been done using two classification algorithms as Decision Tree, Naive Bayes,SVM and for unsupervised machine learning algorithm as kmeans clustering. To determine the performance of the algorithms, the experiments were conducted against different grouping, k-fold cross-validation, confusion matrix, and ROC curves.

The results show KNN (99.99% accuracy), K Means Clustering (99.99% accuracy), Decision Tree (99.99% accuracy) and hybrid machine learning 99.98% accuracy in three evaluation All,grouped,binary performs with high accuracy while but Naive Bayes (80.12% accuracy) with sufficient accuracy.The proposed system shows better accuracy compare to Random Forest and even the two SVM and KNN individually. This thesis focused only on FTP,TCP and UDP attacks because they are the most widely used protocols to launch an attack, and also due to lack of time in the scope of this thesis. We believe experimenting and testing with four types of attacks provides an insight into how to detect all types of attacks in the MANET network. To the best of our knowledge, this is the earliest work carried out to detect DDos, Bot, SSH-Bruteforce, and FTP-BruteForce in the packet flow MANET through the unsupervised machine learning technique .

6.0.1 Future Work

In this thesis work, the aim was to detect DDos, Bot, SSH-Bruteforce, and FTP-BruteForce attacks in the MANET network, and the objective was to propose efficient intrusion detection system for MANET that should lead to a concrete implementation in the future. Therefore, a DDos, Bot, SSH-Bruteforce, and FTP-BruteForce detection method in the MANET network is proposed in detail throughout the thesis, and offline data were used for all the experiments in training the models and testing the models. For future work, we would like to recommend testing this method in a real test environment. Secondly, this thesis focused only on the FTP,UDP flood and TCP SYN flood. In the future, we would like to include all possible DDos, Bot, SSH-Bruteforce, and FTP-BruteForce attacks in order to protect the MANET services. Lastly, we would like to perform transfer learning for intrusion detection for manet.

Bibliography

[1] Hoebeke J, Moerman I, Dhoedt B, Demeester P. Anoverview of mobile ad hoc networks: applications and challenges. Journal-Communications Network 3:60–66, 2004.

[2] H. Jeroen, M. Ingrid, D. Bart, and D. Piet, ”An overview of Mobile Ad hoc Networks: Applications and Challenges,” Journal of the Communications Network,vol. 3, pp. 60–66, 2004.

[3] Perkins CE. Ad Hoc Networking. Addison-Wesley Professional: Boston, MA, USA, 2008.

[4] Chlamtac I, Conti M, Liu JJN. Mobile ad hoc networking: imperatives and challenges. Ad Hoc Networks 1:13–64 ,2003.

[5] Stallings W. Wireless Communications and Networks. Pearson Education: India, 2009.

[6] Murthy CSR, Manoj BS. Ad Hoc Wireless Networks: Architectures and Protocols. Pearson Education:Delhi, India, 2012.

[7] Mishra A, Nadkarni KM. Security in wireless ad hoc networks. In The Handbook of Ad Hoc Wireless Networks.CRC Press: Boca Raton, FL, USA, 499–549,2003.

[8] Yang H, Luo H, Ye F, Lu S, Zhang L. Security in mobile ad hoc networks: challenges and solutions. IEEE Wireless Communications 11:38–47,2004.

[9] Earle AE. Wireless Security Handbook. CRC Press / Auerbach Publications: Boston, MA, USA, 2005.

[10] Patcha A, Park JM. An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer Networks 51(12):3448–3470 2007.

[11] Liao HJ, Richard Lin CH, Lin YC, Tung KY. Intrusion detection system: a comprehensive review. Journal of Network and Computer Applications 36(1):16–24,2013.

[12] R Bace and P Mell. Intrusion detection systems, national institute of standards and technology (NIST). Technical Report 800-31, 2001.

[13] SP NIST. 800-94, a guide to intrusion detection and prevention systems (IDPs). Information Technology Laboratory, National Institute of Standards and Technology, USA, 2007.

[14] D. E. Denning, An Intrusion Detection Model," IEEE Transactions in Software Engineering, vol. 13, no. 2, pp. 222- 232, USA, 1987

[15] Y. Lecun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) 436,2015.

[16] S. K. Wagh, V. K. Pachghare, and S. R. Kolhe, "Survey on intrusion detection system using machine learning techniques," Int. J. Comput. Appl., vol. 78, no. 16, 2013.

[17] Muhammad Ashfaq Khan et. A Scalable and Hybrid Intrusion Detection System Based on the Convolutional-LSTM Network ,22 April 2019.

[18] Chaudhary, A., & Shrimal, G. Intrusion Detection System based on genetic algorithm for detection of distribution denial of service attacks in MANETs. Retrieved from SSRN 3351807,(2019).

[19] Nwabueze, C. and Akaneme, ; Wireless Fidelity (WiFi) Broadband Network Technology: An Overview with Other Broadband Wireless Networks, Nigerian Journal of Technology, Vol. 28, No. 1, pp 71 - 78 S. (2009).

[20] Gai, K., Qiu, M., Tao, L., & Zhu, Y. Intrusion detection techniques for mobile cloud computing in heterogeneous 5G. Security and Communication Networks, 9(16), 3049- 3058,(2016).

[21] Jose, S., Malathi, D., Reddy, B., & Jayaseeli, D. A Survey on anomaly-based host Intrusion Detection System. In: Journal of Physics: Conference Series 1000(1), 012049. IOP Publishing,(2018).

[22] Liu, M., Xue, Z., Xu, X., Zhong, C., & Chen, J. . Host-based Intrusion Detection System with system calls: Review and future trends. ACM Computing Surveys (CSUR), 51(5), 98,(2018).

[23] Marteau, P. F. Sequence covering for efficient host-based intrusion detection. IEEE Transactions on Information Forensics and Security, 14(4), 994-1006,(2019).

[24] Taher, K. A., Jisan, B. M. Y., & Rahman, M. M. Network intrusion detection using supervised machine learning technique with feature selection. In 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST) (pp. 643-646). IEEE, (2019).

[25] Singla, N., Singh, R.: Wormhole attack prevention and detection in MANETs using HRL Method. Int. J. Adv. Res. Ideas Innov. Technol. 3(2) (2017).

[26] Singh, S., Mishra, A., Singh, U.: Detecting and avoiding of collaborative black hole attack on MANET using trusted AODV routing algorithm. In: Symposium on Colossal Data Analysis and Networking (CDAN), 18–19 Evaluation of Deep Learning Approaches for Intrusion Detection System in MANET 997,March 2016.

[27] Jamal, T., Butt, S.A.: Malicious node analysis in MANETS. Int. J. Inform. Technol. 11,859–867 (2018)

[28] Moudni, H., Er-rouidi, M., Mouncif, H., El Hadadi, B.: Performance analysis of AODV routing protocol in MANET under the influence of routing attacks. In: International Conference on Electrical and Information Technologies (ICEIT), 4–7 May 2016 (2016).

[29] Sibanjan Das, Umit Mert Cakmak, Hands-on Authomated Machine Learning: A Beginners Guide to Building Automated Machine Learning Systems Using AutoML and Python, Packt Publishing Ltd, 2018.

[30] Garrido, a. P. J. Gabriel, OpenCV 3.x with Python by Example, Birmingham:, Packt Publishing Ltd, 2018.

[31] Rajanarayanan Thottuvaikkatumana, Apache Spark 2 for Beginners, Packt, 2016.

[32] Birmingham, Hands-On Automated Machine learning, Packt Publishing Ltd, 2018.

[33] Mark Ryan M. Talabis, D. Kaye, ”Supervised Learning,” Available at: https://www.sciencedirect.com/topics/computer-science/supervised-learning, Last Accessed on 3 April 2021.

[34] Dipanjan Sarkar, Raghav Bali and Tushar Sharma, Practical Machine Learning With Python, Apress, 2018.

[35] Dipanjan Sarkar, Raghav Bali and Tamoghna Ghosh, Hands-On T ransfer Learning with Python, Packt Publishing, 2018.

[36] Sunshine10,”Unsupervised Learning,”Available at: https://whatis.techtarget.com/definition/unsupervised-learning, Last Accessed on 3 April 2021.

[37] Dipanjan Sarkar, Raghav Bali , and Tushar Sharma, Practical Machine Learning with Python : A Problem-Solver’s Guide to Building Real-world Intelligent Systems, Apress, 2018.

[38] Girish Chandrashekar, Ferat Sahin, “A Survey on Feature Selection Methods,” Computers Electrical Engineering, Volume 40, Issue 1, pp 16-28, ISSN 0045-7906, 2014.

[39] Flach, Peter, “Machine Learning: The Art and Science of Algorithms That Make Sense of Data”, Cambridge: Cambridge University Press, doi:10.1017CBO9780511973000,2012.

[40] Clarence Chio, David Freeman, “Machine Learning and Security: Protecting Systems with Data and Algorithms”, 1st ed., O’Reilly, 2018.

[41] Shalev-Shwartz, Shai, and Shai Ben-David, “Understanding Machine Learning: From Theory to Algorithms”, Cambridge: Cambridge University Press, doi:10.1017/CBO9781107298019,2014.

[42] scikit-learn.org/stable/modules/naive bayes [Accessed 27 Feb. 2020]

[43] Zaki, Mohammed J., and Wagner Meira, Jr. “Probabilistic Classification.” Chapter. In Data Mining and Machine Learning: Fundamental Concepts and Algorithms, 2nd ed., pp 469–82. Cambridge: Cambridge University Press, doi:10.1017/9781108564175.023,2020.

[44] Panwar, Shivendra S., Shiwen Mao, Jeong-dong Ryoo, and Yihan Li. “TCP/IP Overview.” Chapter. In TCP/IP Essentials: A Lab-Based Approach, pp 1–25. Cambridge: Cambridge University Press, doi:10.1017/CBO9781139167246.003,2004.

[45] Heady R, Luger G, Maccabe A, Servilla M The architecture of a network level intrusion detection system. Technical Report, Computer Science Department, University of New Mexico,(1990).

[46] Huang Y, Lee W Attack Analysis and Detection for Ad Hoc Routing Protocols. In Proc of Recent Adv in Intrusion Detect LNCS 3224:125-145,(2004).

[47] Smith AB An Examination of an Intrusion Detection Architecture for Wireless Ad Hoc Networks. In Proc of 5th Natl Colloq for Inf Syst Secur Educ,(2001).

[48] Amouri, A. Cross Layer-based Intrusion Detection System Using Machine Learning for MANETs, USF, Tampa, FL, USA, April 23, 2019.

[49] B. Sun, K. Wu, U. W. Pooch, Routing anomaly detection in mobile ad hoc networks, in: The International Conference on Computer Communications and Networks, . ICCCN 2003. Proceedings, 2003, pp. 25–31,2003.

[50] Wahab, O. A., Mourad, A., Otrok, H., and Bentahar, J. . CEAP: SVM-based intelligent detection model for clustered vehicular ad hoc networks. Expert Systems with Applications,50, 40–54,(2016).

[51] Singh, D.,& Bedi, S. S. Multiclass ELM based smart trustworthy IDS for MANETs. Arabian Journal for Science and Engineering, 41(8), 3127–3137, (2016)..

[52] Kolias, C., Kambourakis, G., Stavrou, A., & Gritzalis, S. Intrusion detection in 802.11 net- works: Empirical evaluation of threats and a public dataset. IEEE Communications Surveys and Tuto- rials, 18(1), 184–208,(2016).

[53] Subba, B., Biswas, S., & Karmakar, S.Intrusion detection in mobile ad-hoc networks: Bayes- ian game formulation. Engineering Science and Technology, an International Journal, 19(2), 782–799,(2016).

[54] Ahmed, M. N., Abdullah, A. H., & Kaiwartya, O. FSM-F: Finite state machine based frame- work for denial of service and intrusion detection in MANET. PLoS ONE, 11(6), e0156885,(2016).

[55] Shanthi, K., Murugan, D., & Ganesh Kumar, T. Trust-based intrusion detection with secure key management integrated into MANET. Information Security Journal: A Global Perspective, 27, 1–9,(2018).

[56] Khan, F. A., Imran, M., Abbas, H., & Durad, M. H. A detection and prevention system against collaborative attacks in mobile ad hoc networks. Future Generation Computer Systems, 68, 416–427,(2017).

[57] Raja, R., & Ganesh Kumar, P. QoSTRP: A trusted clustering based routing protocol for mobile ad hoc networks. Programming and Computer Software, 44(6), 407–416,(2018).

[58] Anusha, K., & Sathiyamoorthy, E. A new trust-based mechanism for detecting intrusions in MANET. Information Security Journal: A Global Perspective, 26(4), 153–165,(2017).

[59] Luong, N. T., Vo, T. T., & Hoang, D. FAPRP: A machine learning approach to flooding attacks prevention routing protocol in mobile ad hoc networks. Wireless Communications and Mobile Computing, 1–17,(2019).

[60] Sasirekha, D., & Radha, N. Secure and attack aware routing in mobile ad hoc networks against wormhole and sinkhole attacks. In 2017 2nd international conference on communication and electronics systems (ICCES),(2017).

[61] Ahmad, I., Basheri, M., Iqbal, M. J., & Rahim, A. Performance comparison of support vec- tor machine, random forest, and extreme learning machine for intrusion detection. IEEE Access, 6, 33789–33795,(2018).

[62] Yin, C., Zhu, Y., Fei, J., & He, X. A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access, 5, 21954–21961,(2017).

[63] Khan, M. A., Karim, M. R., & Kim, Y. A scalable and hybrid intrusion detection system based on the convolutional-LSTM network. Symmetry, 11(4), 583. https://doi.org/10.3390/sym11 040583,(2019).

[64] Xu, C., Shen, J., Du, X., & Zhang, F. An intrusion detection system using a deep neural network with gated recurrent units. IEEE Access, 6, 1,(2018).

[65] E. A. Shams, & A. Rizaner, "A novel support vector machine-based intrusion detection system for mobile ad hoc networks," Wireless Networks, Vol.24, No.5, pp.1821-1829,2018.

[66] B. Sen, M. G. Meitei, K. Sharma, M. K. Ghose, & S. Sinha, "A Trust-Based Intrusion Detection System for Mitigating Blackhole Attacks in MANET," In Advanced Computational and Communication Paradigms , Springer, Singapore, pp. 765-775,2018.

[67] R. A. Sowah, K. B. Ofori-Amanfo, G. A. Mills, & K. M. Koumadi, "Detection and Prevention of Man- inthe-Middle Spoofing Attacks in MANETs Using Predictive Techniques in Artificial Neural Networks (ANN)," Journal of Computer Networks and Communications, Vol. pp. 1-14,2019.

[68] Amar Amouri Cross Layer-based Intrusion Detection System Using Machine Learning for MANETs 2019.

[69] Kunal and M. Dua, "Machine Learning Approach to IDS: A Comprehensive Review," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, pp. 117-121,2019.

[70] M. Islabudeen1 · M. K. Kavitha Devi A Smart Approach for Intrusion Detection and Prevention System in Mobile Ad Hoc Networks Against Security Attacks 2020.

[71] scikit-learn.org/stable/modules/naive bayes [Accessed 24 April. 2021]

[72] Md Nasimuzzaman Chowdhury, Ken Ferens and Mike Ferens, "Network Intrusion Detection Using Machine Learning", International Conference on Security and Management, 2016.

[73] Moustafa, Nour, and Jill Slay. “UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 network data set).”Military Communications and Information Systems Conference (MilCIS), 2015. IEEE, 2015.

[74] T. Zhi, H. Luo and Y. Liu, “A Gini Impurity-Based Interest Flooding Attack Defence Mechanism in NDN,” in IEEE Communications Letters, vol. 22, no. 3, pp.538-541, March 2018.

[75] Watt, Jeremy, Reza Borhani, and Aggelos K. Katsaggelos, “Linear Two-Class Classification.” Chapter. In Machine Learning Refined: Foundations, Algorithms, and Applications, 2nd ed., pp 125–73. Cambridge: Cambridge University Press. doi:10.1017/9781108690935.010,2020.

[76] Mahendra Prasad1(B),Sachin Tripathi1, and Keshav Dahal,”Intrusion Detection in Ad Hoc Network Using Machine Learning Technique”,Conference Paper · December 2020