Middlesex University Dubai

Faculty of Science and Technology

Computer Engineering & Informatics

DISSERTATION REPORT
CST4090

Cyber Threat Detection Using Machine Learning Techniques: A Performance Evaluation Supervised Machine Learning Algorithms In Intrusion Detection

Supervisor:

Dr. Maha Saadeh

Submitted By:

Suleyman Olushola Yahaya

MSc Data
Science
M00736597

1st May 2022

Abstract

The world is becoming increasingly reliant on the Internet, making Cyberspace a necessary component of all aspects of modern life. And judging by the reliance on software-based systems and the burgeoning growth in networking applications for day-to-day operations worldwide, there is an urgent need to develop more robust measures to combat the increasing number of cyber threats and attacks.

Entities worldwide are investing a ton towards the security of their assets making Machine Learning critical in Cyber Security due to its flexibility, scalability and adaptability to new and developing situations.

The current Intrusion detection systems struggle to detect stealthy Intrusions in Cyberspace. One of the alternative methods recommended for detecting unauthorised access or infiltration is Machine Learning.

Machine learning is a crucial phenomenon that has aided humanity in improving numerous industries, professional procedures, and everyday life.

In Cyber threat detection, Supervised Machine Learning algorithms have been a prominent approach. The detection of intrusions using simulated data has recently been identified as a possible application area for these approaches.

This research compares some of the most extensively utilised Supervised Machine Learning algorithms to detect intrusions using the NSL KDD dataset. The five machine learning techniques evaluated include Random Forest, k-Nearest Neighbor, Decision Tree, Naive Bayes, and Support Vector Machines.

The experimental results obtained showed that the Support Vector Machine produced the best results in classifying and identifying intrusion and routine network traffic with an average accuracy of 76%, an AUC of 0.84, an F1 score of 81%, and a Recall average of 70% o over the rest of the classification algorithms.

Table of Contents

List of Figures

Acknowledgement

Starting off by saying Alhamdulillah. I want to extend my deepest gratitude to my family for being my rock and support, my friends and everyone I have had the pleasure of meeting throughout my journey at Middlesex University. My sincere thanks and gratitude to my thesis advisor, Dr Maha Saadeh, for her patience and guidance throughout my dissertation, Dr Krishnadas Nanath, the rest of the faculty and the staff at Middlesex University for their unwavering support and help. I owe a huge part of my journey as a Data Scientist to you all. Thank you!

1. Introduction

The Internet, technically proficient users, system resources, data, and uneducated users are all part of Cyberspace. Cyberspace provides a worldwide platform for unrestricted access to knowledge and resources (Shaukat et al., 2020). This access comes with many risks and more common intrusions, allowing cybercriminals to profit from these flaws. Governments and businesses alike may suffer due to the resulting loss of revenue and credibility (Lambert, 2017).

Security breaches where enormous amounts of information are released online and then used by criminals to conduct financial fraud can also result from cyberattacks. When cybercriminals gain access to enough information to carry out destructive acts, these attacks directly impact individuals, organisations, and governments alike.

Intrusion detection has been a hot-button subject of cybersecurity research since the 1980s, with Malicious software (malware) evolving rapidly, posing a significant challenge to the design of intrusion detection systems (IDS). The most challenging task is identifying unknown and obfuscated malware since malware developers employ various evasion tactics for information concealment to avoid detection by an IDS (Khairat et al., 2019)

Given the continuous and high-tech changes, identifying the number of Intrusion attacks on cyberspace devices and their economic impact is challenging. The core goals of an intrusion detection system (IDS) have remained unchanged, despite early research focusing on host intrusion detection systems (HIDS).

An intrusion detection system that has been well-designed and executed should be able to detect a wide range of intrusions, possibly in real-time, with high discriminating power, improve itself through self-learning, and be adjustable in design and execution (Dhooge et al., 2019).

Given the continuous and high-tech changes, identifying the number of Intrusion attacks on cyberspace devices and their economic impact is challenging. Conventional approaches have proven to be inept in stemming these threats and attacks, necessitating cybersecurity and forensic specialists to develop creative strategies for detecting, analysing, and defending against cyber threats in real-time. In practice, dealing with such attacks is impossible without thoroughly examining the attack features and taking appropriate intelligent defensive actions. Cyberattacks have increased dramatically, with various industries experiencing penetrations and service disruptions since the COVID 19 pandemic. The protection of data and devices in Cyberspace has become the most talked-about topic in the I.T. industry. The vast amount of data collected and kept poses several hazards and security concerns. To ensure the security of Cyberspace, proactive measures to address these challenges are required. Enhancing the capabilities of intrusion detection systems is the first step in addressing these concerns.

1.1. Background

Intrusions Detection systems identify the vulnerabilities within a computer system or a network that attackers can exploit in future attacks. A modern security system consists of Endpoint security, data encryption, secure authentication, detection, and response. Machine Learning makes it easy to collect, analyse, interpret cyber assault evidence and defend against unauthorised access and malicious attacks. Security specialists have widely employed machine learning techniques to construct intrusion detection systems (IDS) that protect domains and networks from cyber threats in real-time and automatically (Wang et al., 2020).

Machine learning is a potential intrusion detection technology with low resource requirements. The supervised learning framework, in particular, can swiftly adapt and classify various sorts of attacks.

Massive cyberattacks occur when cybercriminals utilise the tools available to launch planned and politically driven assaults against endpoints, networks, data, and other I.T. infrastructures, resulting in data loss for individuals, businesses, and governments worldwide. There has been a surge in the frequency of cyber-crime as a result of the rapid development in digital transactions internationally across sectors. The market for intrusion detection and prevention systems is fueled by an increase in business data breaches or data leaks.

According to a publication released by MarketsandMarkets in July 2020, the worldwide Intrusion Detection And Prevention Systems (IDPS) market is expected to increase at a CAGR of 5.4%, from USD 4.8 billion in 2020 to USD 6.2 billion by 2025. The growing number of intrusions, evolving cyber ecosystem, and expanding requirements for conformity with numerous forthcoming laws are driving the industry.

The thesis provides a detailed evaluation of commonly used Supervised Machine Learning algorithms to assess their performance in detecting a well-known cybercrime, focusing on the efficiency and performance of these algorithms in detecting intrusions into Cyberspace.

Evaluating the highlighted Machine Learning techniques should guide new researchers, state actors, and private entities in improving existing Intrusion Detection systems.

1.1.1. Research Terminologies

- **Cyberspace**
  The interconnected network of information systems infrastructures, which includes the Internet, telecommunications networks, computer systems, and embedded processors and controllers, forms a worldwide domain inside the information environment.
- **Intrusion**
  An unlawful and unauthorised activity within a digital network.
- **Intrusion Detection**
  The capacity to monitor and respond to computer misuse is known as intrusion detection.
- **Intrusion Detection Systems**
  a network monitoring device or software program that looks for harmful activities or policy infractions.
- **Machine Learning**
  the study of computer algorithms that can learn and develop on their own through experience and data.

1.2. Problem Statement

By monitoring, detecting, and responding to unauthorised activities within the system, an IDS should be able to recognise any anomalous patterns and traffic. However, as the proliferation of online transactions increases internationally across industries, the frequency of cyber-crime rises, making it harder for intrusion detection systems to keep up and detect data breaches and prevent data leaking. Machine Learning approaches are being used to detect intrusions and address these threats.

1.3. Aims

Evaluate and Compare the performance of Classification algorithms on the NSL KDD Intrusion Dataset

This project aims to evaluate the performance of selected machine learning algorithms in differentiating between intrusions/attacks and normal activities within the NSL KDD dataset. Five supervised machine learning algorithms have been chosen to perform these tasks. The algorithms include Random Forrest, Decision Trees, Support Vector Machines, K-Nearest Neighbour, and Naïve Bayes.

This research is carried out to improve existing Intrusion Detection systems and further the work already being done in this domain. The following are the three research questions to consider while writing a literature review:

- Q1 – How practical have machine learning approaches been in detecting intrusions?

- Q2 – How does this study compare to other studies of a similar nature?

1.4. Dissertation organisation

The focus of the proposed study is on assessing the performance of machine learning algorithms.

The research progresses in the following manner. The context and purpose of the study are covered in the first section of the introduction. The current information, as well as research methodologies and techniques that have been used in this sector, are discussed in the literature review section.

The methodology chapter describes the approach followed in the execution of the project. Next, The Findings Section describes the findings that support the project's original objectives. The Discussion chapter discusses similar research work and their findings and describes the results supporting the project's aims. The conclusion and Future work chapter conclude the research and further work.

The structure of the dissertation is depicted in the diagram below:

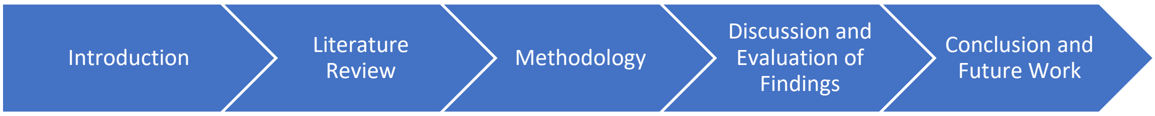

Introduction
Literature Review
Methodology
Discussion and Evaluation of Findings
Conclusion and Future Work

FIGURE 1: DISSERTATION STRUCTURE.

2. Literature Review

This chapter mainly introduces related concepts and research in Machine Learning and Cyber Threat Detection techniques.

2.1. Cyberspace

Civilisation has entered a new phase of information technology, following the electromechanical eras. The information market is growing to be the world's most significant resource in the context of technology. The way people live and work is changing due to information technology. For a large portion of the world's population, electronic information equipment such as computers, the Internet, television, and mobile phones has become indispensable.

People live in a three-dimensional environment of the physical world, human society, and information space in the information era. The world adopted the English term "cyberspace" to describe the information environment or information space in which humans live (Zhang et al., 2015).

Cyberspace is a constantly changing collection of interconnected information systems and the people who engage with them. The term "cyber" has meant nearly everything related to computers and networks, particularly in the security industry. Conflicts in Cyberspace, such as state-on-state cyber warfare, cyber terrorism, cyber militias, and so on, are another expanding subject of research. Regrettably, there is no agreement on what "cyberspace" is, let alone the consequences of cyberspace battles (Otis et al., 2010).

2.2. Cyber Threats and Vulnerabilities

Increasing worldwide reliance on Cyberspace, fast-evolving technology, long-term vulnerabilities, and advanced persistent attacks contribute to a rising societal vulnerability to cyberattacks. In a wide range of domains, including but not limited to communications, transportation, and commerce, ensuring a broader spectrum of infrastructure required for a well-functioning society is vital (Maybury et al., 2015).

Threats to society's essential systems can come from within or outside the country, causing direct or indirect harm to national systems and people. By exploiting weaknesses/vulnerabilities, threats might trigger asset loss or takeover. The three sorts of threats include physical, economic, and cyber threats. (Baker et al., 2012).

The table shows the examples of the Classification of threats;

Physical Threats	Economic Threats	Cyber Threats
• Natural disasters • Conventional warfare • Unconventional Warfare (Terrorism) • Technical disruptions	• National Depression. • Personal financial loss • Global Financial Market Crash	• Attacks on IoT Devices • Denial of Service attack or Distributed Denial of Service Attack (DDoS) • Ransomware.

		- Malware on Mobile Apps and Computer network systems,

TABLE 1: THREATS

Numerous threats include various forms of assaults and techniques, viruses, and physical dangers. The European Network and Information Security Agency (ENISA) uses a threat-based paradigm, which asserts that "any person or thing that acts (or can act) to originate, convey, transmit, or support a danger" is a threat. Corporations, cybercriminals, workers, hacktivists, nation-states, and terrorists are primary danger agents (Simola, 2019).

According to a report released by Broadcom in 2012, cyber-attacks and the time spent by businesses trying to recover from these attacks cost the United States a whopping $385 billion yearly. The number of people who cyber-attacks have harmed is likewise on the rise. Broadcom's poll also revealed that 69% of people in 24 countries said they had been the victim of cybercrime at some point in their lives. And also, 14 individuals are victims of a cyber assault per second, or over one million attacks every day.

Computer networks are vulnerable to various threats that jeopardise their availability, integrity, and secrecy. Some of these attacks include:

2.2.1. Denial of Service attack or Distributed Denial of Service Attack (DDoS)

DDoS (Distributed Denial of Service) attacks are common intimidation over the Internet. The network bandwidth in a DDoS attack indicates the victims' computer machines and resources are exhausted to transmit many packets to a targeted server and are launched from a network of zombie botnet computers that are remotely controlled, well-organised, and widely distributed. Many traffic or service requests are sent to the target system simultaneously or in a continuous stream. Due to the assault, the target system becomes useless, reacts slowly, or crashes entirely (Khalaf et al., 2019).

A denial of service (DoS) attack is an effort to prevent authorised users from accessing computer, server, or network resources by temporarily interrupting or suspending services. Denial of Service attacks is a subset of Distributed Denial of Service Attack (DDoS).

2.2.2. Phishing

Phishing schemes seek to steal consumers' passwords or sensitive information such as credit card numbers. In this situation, fraudsters utilise bogus URLs to send victims emails or text messages that appear to come from a reputable source (IBM, 2021). Basnet et al.(2019) described Phishing as a misleading tactic that combines social engineering and technology to get sensitive personal information such as passwords and credit card numbers by impersonating a trustworthy person or company in electronic contact.

Phishing has evolved into a serious hazard to both people and organisations, costing them millions in financial losses.

In July 2021, the Anti-Phishing Working Group (APWG) recorded 260,642 phishing assaults, the largest monthly total. Brands targeted increased significantly, from just under 400 per month to more than 700 in September, with phishing attempts in Brazil rising from 4,275 in Q2 to 7,741 in Q3.

APWG (2021) found that the software-as-a-service (SaaS) and webmail sectors were the most commonly targeted by phishing attempts, accounting for 29.1 % of assaults. In contrast, attacks against financial institutions and payment providers accounted for 34.9 % of all attacks. Phishing attempts against cryptocurrency exchanges and wallet providers accounted for 5.6 % of these attacks.

2.2.3. Ransomware

IBM (2021) Classifies Ransomware as malware that takes advantage of system flaws and encrypts data or system functions to keep it hostage. A ploy Cybercriminals utilise to extort money in return for the system's release. The addition of extortion methods to Ransomware is a recent development. With the ransomware-as-a-service concept enabling easy availability and deployment, and the possibility for huge earnings providing a viable criminal business model, Ransomware has become a serious global concern.

Alhawi et al. (2018) reported that Individuals, private enterprises, and governmental service providers, such as healthcare and utility corporations, can be targets of ransomware attacks, resulting in significant disruption and financial loss.

Gartner's press release in 2021 shows that the threat of new ransomware models is the top emerging risk facing organisations. In the third quarter of 2021, the danger of "new ransomware models" was the top concern among CEOs. In a study conducted on 294 senior executives from various sectors and regions, fears about Ransomware surpassed pandemic-related issues, including supply chain disruptions.

2.2.4. Malware

Malware is a type of harmful software that may make infected computers inaccessible. Most malware versions destroy data by destroying or erasing files required for the operating system to function (IBM, 2021).

Malicious software, commonly known as malware, is one of the most common online hazards today. Many dangerous tools, from traditional computer viruses to Internet worms and botnets, attack computers connected to the Internet. This threat is fueled by a criminal sector that methodically assembles networked servers for unlawful spam distribution and data collection (Rieck et al., 2011). The INSIKT Group, in August 2021, reported that Threat organisations would target these defects to transport, distribute, and execute harmful programs onto susceptible systems. Therefore trends in vulnerability exploitation and malware assaults frequently cross. Several noteworthy cyber events garnered media attention in the first half of 2021 due to their widespread impact and unique approaches utilised in assaults that highlight this junction. In these cases, threat actors used significant vulnerabilities to install malware on vulnerable systems such as Accellion FTA software, Microsoft Exchange Servers, macOS, and QNAP devices. These assaults show how hackers, ransomware operators, and state-sponsored organisations identify and exploit high-risk vulnerabilities.

2.2.5. Zero-day Exploit

Norton (2021) explained that Zero-day attacks start with zero-day vulnerabilities, with exposure being a flaw or fault in the security software. These can be caused by incorrect computer or security setups and programming faults made by developers. These vulnerabilities lead to exploits where hackers exploit the missed flaw for malicious intent, often through malware, to launch a cyberattack.

These vulnerabilities can exist for days, months or years before developers learn about the flaws, mainly because software defects are less predictable than hardware problems. And the process of uncovering such flaws and building exploits appears to be chaotic. The security risk against unknown zero-day attacks has been regarded as immeasurable (Wang et al., 2010).

For the past several years, the growing volume and variety of threats have caused severe security concerns and apprehensions in the security sector.

Traditional Intrusion Detection/Protection systems are ill-equipped to deal with these threats and attacks.

2.3. Intrusion and Intrusion Detection System

Intrusion is a severe security disclosure challenge since a single intrusion could be enough to erase or steal data from a computer. IDS has been increasingly important in recent years for dealing with security breaches and resolving issues of concern. A considerable number of research papers in intrusion detection are currently being presented. Different approaches aim to improve the system's ability to distinguish between regular and strange packets in network traffic (Saranya et al., 2020).

Intrusion Detection Systems (IDS) are a critical piece of technology for protecting humans from cyber-attacks. Every transaction and information processing takes place over the Internet, which is highly vulnerable to various fraudulent activities.

Rapid advancements in the internet and communication areas have resulted in a massive expansion of Cyberspace's network size and data. As a result, many new threats are being developed, making it difficult for cyberspace security to identify breaches effectively. Additionally, the presence of intruders in Cyberspace with the intent of launching various attacks cannot be overlooked. An intrusion detection system (IDS) is a tool that inspects network traffic to assure its confidentiality, integrity, and availability and thereby protects the infrastructure from possible invasions. Despite the researchers' best efforts, IDS continues to encounter difficulties in boosting detection accuracy while lowering false alarm rates and detecting unique intrusions (Ahmed et al., 2020).

Security concerns can affect almost every component of a network. The data node, for example, could be highly significant to a company. Any compromise of the node's information could significantly negatively impact the organisation's market reputation and financial losses. Existing IDSs have demonstrated ineffectiveness in detecting various threats, including zero-day attacks, and in lowering false alarm rates (FAR). This eventually leads to a demand for an efficient, accurate, and cost-effective network intrusion detection system (NIDS) (Prasad et al., 2019).

Anomaly-based, signature-based, and hybrid-based are three prevalent intrusion detection methods.

2.3.1. Anomaly-Based System

Anomaly-based intrusion detection is also known as behaviour-based detection. It models the behaviour of users, networks, and host systems and creates an alarm or alert for the administrator when the behaviour deviates from the norm. Signature-based IDSs are also known as knowledge-based detection systems.

This strategy is based on a database that comprises prior known attack signatures and known system flaws. An anomaly-based intrusion detection system and a signature-based intrusion detection system are combined in a hybrid detection system. Most Intrusion Detection Systems employ intrusion detection techniques, such as anomaly or signature. Because both intrusion detection systems have flaws, hybrid IDS can be used (Saranya et al., 2020).

2.3.2. Signature-Based System

Signature-based systems, sometimes referred to as misuse detection techniques detect previously identified intrusions by comparing their signatures to the data being processed. They work well in situations where most assault patterns are previously known. Their key drawbacks are the requirement for regular signature database updates and the inability to identify zero-day assaults and unknown threats (Rbah et al.,2022).

2.3.3. Hybrid-Based System

The principles of a Host Intrusion Detection System (HIDS) and a Network Intrusion Detection System (NIDS) are combined in a Hybrid Intrusion Detection System. It is created by obtaining data or information from both the host and the network for analysis and applying an analytical methodology (Chauhan et al., 2013).

In contrast to anomaly-based IDS, which is generated automatically, specification-based IDS requires developers to actively establish the constraints and features that reflect normal operating behaviour. As a result, the intrusion detection algorithm may detect suspicious behaviour by detecting the difference between the two patterns. Compared to anomaly-based solutions, approaches based on specification and signature have low overhead(Rbah et al., 2022).

2.4. Applications of Intrusion Detection Systems (IDS)

Intrusion Detection Systems (IDS) are a critical piece of technology for protecting humans from cyber-attacks. Every transaction and information processing takes place over the Internet, which is highly vulnerable to various fraudulent actions. As a result, there is a need to devote more attention to information security.

2.4.1. Intrusion Detection System for the Internet of Things (IoT)

The Internet of Things (IoT) is a network of interconnected devices that allows for seamless data exchange between physical items. Medical and healthcare equipment, autonomous vehicles, industrial robots, smart T.V.s, wearables, and smart city infrastructures may be remotely monitored and controlled. IoT devices are predicted to outnumber mobile devices in terms of usage, and they will have access to the most sensitive data, such as personal information. As a result, the attack surface area will grow, and the likelihood of attacks will rise. IoT intrusion detection systems must be created to secure communications enabled by IoT technologies, as security will be a critical supporting feature of most IoT applications (Granjal et al., 2015).

The Internet of Things (IoT) is fast growing to have a more significant impact on ordinary lives and massive industrial systems. However, this has attracted the attention of cybercriminals, who have turned IoT into a target for illicit operations, potentially exposing end nodes to attack. Due to this, various IoT intrusion detection systems (IDS) have been presented in the literature to combat attacks on the IoT ecosystem, which may be grouped into three categories depending on detection approach, validation strategy, and deployment strategy.

2.4.2. Intrusion Detection System for Cloud and Big Data Environment

In both industry and scientific institutes, the volume of data has expanded dramatically since the millennium. We are unlikely to be able to process the amounts and variety of data we are dealing with traditional software solutions. As a result, new big data processing technologies, which can disseminate and analyse data in a scalable manner, are being incorporated into or replacing traditional Business Intelligence (B.I.) systems. Data, which is crucial for decision-making, forecasting, and marketing-related competitiveness, is constantly at risk. Data is a valuable target for criminals, but they are also prized in industrial espionage, which specialises in tapping and manipulating company data. On the other hand, companies and research institutes face a variety of dangers. As a result, data security is critical for I.T. departments and the entire entity, such as a company or research institution (Azeroual et al., 2020).

2.4.3. Intrusion Detection System for Smart Grid

Countries and their economies rely on their respective electrical grid; a highly reliable technique for protecting against malicious incursion activities in these grids' physical and cyber layers is required. It is vital to detect intrusions to take the appropriate remedial steps as soon as possible to ensure the safe operation of the Smart Grid. A smart grid is a cyber-physical system that includes cyber (advanced metering infrastructure (AMI), control and automation systems, protocols, databases, and so on) as well as physical infrastructure (transmission lines, circuit breakers, relays, and generation units) to ensure the grid's reliability, efficiency, and resilience. Cyber-attacks aim to deceive power system operators by altering data collected from measurement devices (Jena et al., 2022).

2.5. Machine Learning

Mahesh (2018) defines Machine Learning as the scientific study of algorithms and statistical models that computer systems employ to complete a particular task without being explicitly taught. Much daily software involves learning algorithms in the "big data" era. Machine Learning is a branch of Artificial Intelligence. It accomplishes a particular objective by relying on the results of previous experiences rather than being expressly coded. As a result, machine learning does not need to be explicitly infused with data. When search engines like Google, Bing, Yahoo etc., are used to search the Internet, they perform so effectively because they utilise learning algorithms to rank web pages. These algorithms are used for various applications, including traffic prediction, image processing, product recommendations, speech recognition, predictive analytics, etc.

The majority of people identify only five to ten different attributes of data. A machine learning algorithm can process thousands of variables and parameters to uncover unique combinations and relationships in data. Because it allows computers to apply knowledge and extract value from massive data sets, Machine Learning is a helpful form of Artificial Intelligence for business. Streaming services like Netflix and Amazon Prime keep track of tens of millions of data points across hundreds of millions of members. Their algorithms can predict how individuals react to various programming types based on their ratings, preferences, prior viewership patterns, and clickstream histories (Baum, 2021).

Machine learning is a growing field of computational algorithms that mimic human intelligence by learning from their surroundings, and it is regarded as the base for all Artificial Intelligence projects. Pattern recognition, computer vision, spacecraft engineering, finance, entertainment, computational biology, and biological and medical applications have all benefited from machine learning techniques (El Naqa et al., 2015).

Machine Learning employs a variety of algorithms to solve different problems. Data scientists like to point out that there is no such thing as a "one-size-fits-all" algorithm for solving an issue. The type of algorithm used is determined by the problem and data available.

Machine learning algorithms are trained in various ways, each with its benefits and drawbacks. The data utilised must be considered to comprehend the advantages and disadvantages of every sort of Machine Learning; these data could be labelled and unlabelled.

Machine Learning and Artificial intelligence (A.I.) are the fastest-growing disciplines of computer science, and they provide a viable solution for achieving cyber security and combatting security threats. This cross-disciplinary research focuses on two elements when these phenoms meet cyber security. Machine Learning and other A.I. technologies may be used in cyber security to build intelligent models for intrusion detection, malware categorisation, and threat intelligence.

Furthermore, to counteract adversarial efforts and maintain privacy in Cyberspace. Machine Learning models require specific cyber security defence and protection solutions (Wu et al., 2018).

Intrusion detection systems play a critical part in Cyberspace security. The intrusion detection model is a predictive model that determines if data traffic is normal or intrusive. Machine learning methods are used to develop accurate models for Clustering, Classification, and prediction. In this research, machine learning classification algorithms such as Log Gaussian Naive Bayes, Support Vector Machines, Decision Trees, and Random Forests are used to create intrusion detection classification models (Belavagi et al., 2016).

Machine Learning algorithms are mainly divided into supervised Learning, Unsupervised Learning, Semi-supervised Learning, and Reinforcement learning (Sarker, 2021).

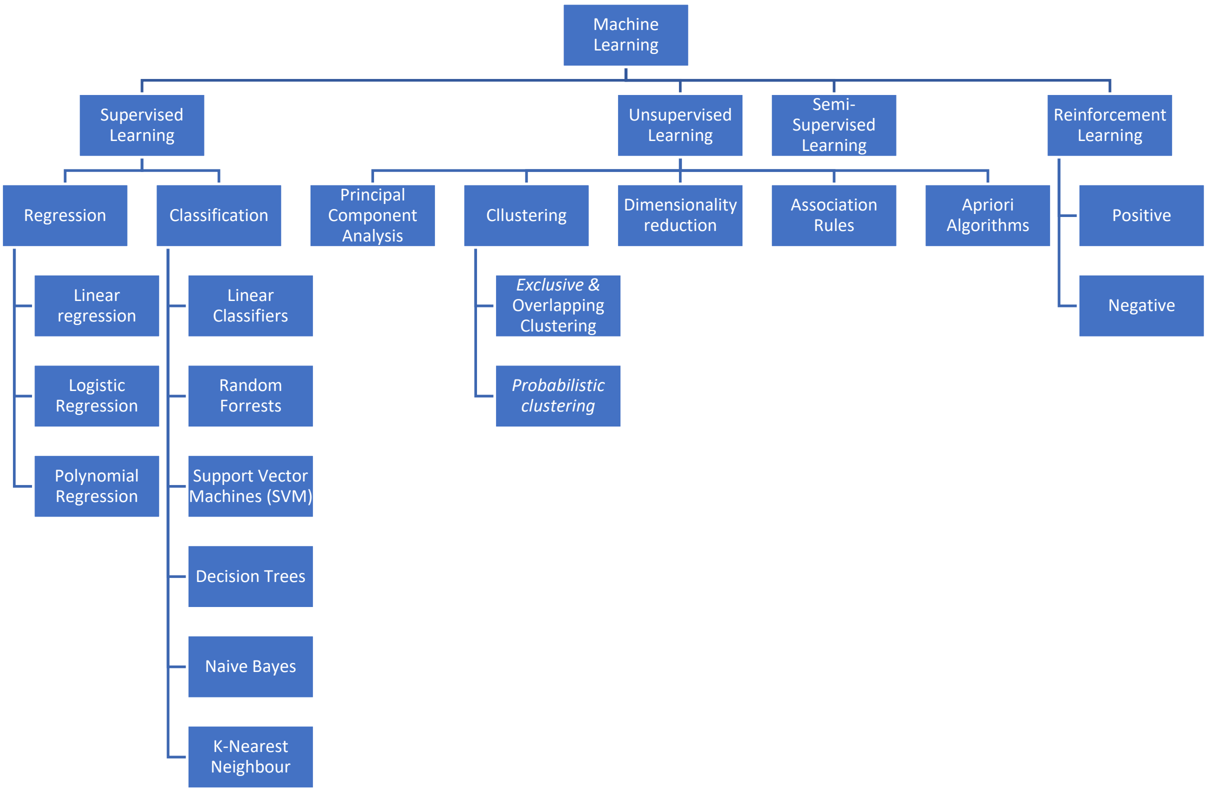


FIGURE 2: MACHINE LEARNING TYPES.

2.5.1. Supervised Learning

Supervised Learning in Machine learning aims to train a function that translates input to an output using sample input-output pairs. It uses labelled training data and training examples to predict an outcome. Supervised Learning is also a task-driven technique when specific goals are specified to be achieved from a particular set of inputs (Sarker, 2021).

The foundation for Supervised Learning is usually a set of data and an explicit knowledge of how that data is classified. The goal is to uncover data patterns used in an analytics process. This data includes labelled features that define the data's meaning (Hurwitz et al., 2018).

Supervised Learning may be divided into two categories: Classification and Regression.

2.5.1.1. Classification

Classification solves problems such as distinguishing apples from oranges by employing algorithms to accurately allocate test data into specific categories. Classification algorithms can be used in the real world to classify spam and place it in a distinct folder from your inbox.

These algorithms include Support Vector Machines (SVM), linear classifiers, decision trees, and random forests.

2.5.1.2. Regression

Another form of Supervised Learning is Regression, which employs algorithms to deduce the relationship between independent and dependent variables. These models help forecast numerical values based on various data sources, such as sales revenue estimates for a particular company. Linear, logistic, and Polynomial Regression are three popular regression algorithms.

2.5.2. Unsupervised Learning

Unsupervised Learning is a machine learning technique in which models are not supervised using a training dataset, as the name suggests. On the other hand, models use the data to uncover hidden patterns and insights. It is comparable to the Learning in the human brain while learning new things.

Unsupervised Learning analyses and clusters unlabeled datasets using machine learning techniques. Without user intercession, these algorithms uncover hidden patterns or data classifications. Unlike traditional observation methods, unsupervised Learning gives an empirical approach to viewing data, helping businesses to discover patterns in enormous volumes of data swiftly.

It is the best solution for exploratory data analysis, cross-selling techniques, consumer segmentation, and image identification because of its capacity to detect similarities and differences in information.

In Unsupervised Learning, we have the input data but no corresponding output data as it cannot be immediately applied to a regression or classification task. Unsupervised Learning aims to discover a dataset's underlying structure, categorise data based on similarities, and represent that dataset in a compressed fashion.

Unsupervised learning models are utilised for three main tasks—Clustering, association, and dimensionality reduction.

Most of today's IDS techniques cannot deal with the dynamic and complicated nature of cyber attacks on computer networks. As a result, practical adaptive approaches, such as machine learning techniques, can lead to higher detection rates, lower false alarm rates, and cheaper computing and communication costs (Zamani et al., 2013).

The ongoing need for up-to-date definitions of the attacks is one of the most agonising pains in the intrusion and virus detection fields. This is due to the employment of a "misuse detection" strategy, in which the goal is to define what is abnormal rather than what is expected. While this method has been extensively successful and is used in almost all recent antivirus and intrusion detection products, its fundamental shortcoming is that misuse-based systems are mainly useless when confronted with an unexpected threat(Zanero et al., 2004).

2.5.2.1. Clustering

IBM (2020) defines Clustering as a data-mining technique that groups unlabeled data into categories based on similarities and differences. Clustering algorithms are used to organise raw, unclassified data objects into groups that are represented by information structures or patterns. Several types of clustering algorithms are exclusive, overlapping, hierarchical, and probabilistic.

2.5.2.2. Association

An association rule is an unsupervised learning strategy for discovering the relationship between variables in an extensive database. It recognises clusters of objects in the collection that appear to be related. Marketing efforts are more effective as a result of the association regulations. The Apriori method is the most extensively used of several algorithms for generating association rules, including Apriori, Eclat, and FP-Growth (IBM, 2020).

2.5.3. Semi-Supervised Learning

Semi-supervised Learning builds prediction models using a small number of labelled training data and many unlabeled ones. It allows leveraging the massive volumes of unlabelled data accessible in many use cases in conjunction with generally smaller labelled data sets. It is fundamentally a cross of Supervised and Unsupervised Learning (van Engel et al., 2020).

In practice, most Semi-Supervised Learning operations rely on extending either supervised or unsupervised Learning to incorporate information that is unique to one. SSL has gotten a lot of interest in machine learning and pattern recognition. The fundamental goal of Semi-Supervised Learning is to create a classification model that works with both labelled and unlabeled data. Semi-Supervised Learning has increased in prominence as labelled data becomes more challenging to come by and unlabeled data becomes more common in various sectors (Jha et al., 2021).

2.5.4. Reinforcement Learning

Reinforcement learning is the process of learning what to try to do in order to maximise a numerical reward signal. Through trial and error, reinforcement learning could be a tool for improving performance. Reinforcement learning can be conceived as a sequenced decision-making problem in which individuals interact with their environment at discrete time points.

One of the challenges that reinforcement learning encounters that other types of Learning do not is striking a balance between exploration and exploitation. It's based on a formal framework that defines how the agent interacts with the environment in terms of states, actions, and rewards (Jha et al., 2021).

2.5.4.1. Positive Reinforcement

Positive reinforcement learning entails doing anything to make the desired behaviour more likely to occur again. It appears to have a favourable effect on individuals' behaviour and strengthens their conduct. Although this form of Positive Reinforcement can last a long time, too much of it can result in a chain reaction of positive events, limiting its effectiveness.

2.5.4.2. Negative Reinforcement

Negative Reinforcement is the polar opposite of positive reinforcement learning in that it enhances the likelihood of a specific behaviour occurring again by avoiding the dire situation. It may be preferable to positive Reinforcement depending on the situation and conduct, but it merely reinforces the minimum essentials of the activity.

There are two crucial learning models in reinforcement learning:

- Markov Decision Process
- Q learning

2.6. Related Work

Tsai et al. (2009) examined 55 similar research published between 2000 and 2007 to see what methodologies were employed, what experiments were undertaken, and what might be addressed for future work from a machine learning standpoint. They concluded that It would be valuable if different ensemble and hybrid classifiers were compared in prediction accuracy.

Choudhury et al. examined the performance of various classifiers in WEKA in 2015 and concluded that RandomForest and BayesNet are suitable for network traffic monitoring. They also compared machine learning algorithms, concluding that Boosting is the most effective method in this procedure. They believe that the improvised algorithms can be used to create effective network intrusion detection devices for use in an organisation's security.

Agrawal et al. (2015) did a survey on anomaly detection for intrusion detection using data mining approaches. They grouped anomaly detection methods into three categories: clustering-based, classification-based, and hybrid approaches. K-means. Under clustering-based techniques, K-Meoids, EM clustering, and Outliers detection algorithms were described. Under classification-based algorithms, they explained the Naive Bayes Algorithm, Genetic Algorithm, Neural Networks, and Support Vector Machine. Hybrid approaches refer to a mix of machine learning techniques.

Ahmed et al. (2016) surveyed network anomaly detection systems that used several machine learning methodologies, such as Classification, Clustering, statistical, and information theory approaches. They summarised some of the challenges with various network intrusion detection datasets. They suggested collaborative Intrusion Detection systems as a potential research avenue, although their review lacks a complete description and analysis of existing IDS ideas based on machine learning. They differentiated regular instances from anomalous instances by using various machine learning algorithms in intrusion detection.

By analysing 20% of the KDD NSL dataset, Ashraf et al., 2018, compared and assessed the classification performances of Naive Bayes, J48, and Random Forest. They discovered that Random Forest outperformed Naive Bayes and J48 in accuracy and detection rate. All three classifiers obtained up to 90% accuracy in terms of precision and recall.

Dobson et al. 2018 employed Apache Spark, a big data processing engine known for performing tasks at high speeds to process network packet data. Random Forest, Support Vector Machines (SVMs), Logistic Regression, Nave Bayes, Gradient Boosted Trees, and a Deep Multilayer Perceptron, a Spark implementation of a deep learning method, were all implemented using the Spark libraries. They compared the outcomes of typical machine learning algorithms to Deep learning results. They discovered that deep learning algorithms yield better accuracy, precision, and recall than traditional machine learning algorithms but take longer to study data.

Liu et al. (2019) developed an IDS taxonomy that presents the numerous machine learning techniques utilised in Cyber Security using data sources as the main thread. Using this Classification, they examined Intrusion Detection Systems applied to various data sources, such as logs, packets, flow, and sessions. They deployed deep learning methods and application scenarios for Intrusion Detection Systems that used these various data types. They assessed and refined the problems and prospects in the field by reviewing current representative studies and providing referrals to other researchers performing in-depth studies.

Mishra et al. (2019) described numerous forms of network and host-based assaults and a summary of their attack characteristics. According to the research, if a strategy works well for detecting one type of assault, it may not work to see other types of attacks. As a result, by classifying numerous machine learning algorithms for each type of assault, the importance of a methodology for specific attacks has been demonstrated.

More recently, a plethora of research has been carried out in Machine Learning to tackle unauthorised intrusion into restricted cyberspaces. Salloum et al. conducted extensive surveys on machine learning and deep learning algorithms for intrusion detection network analysis in 2020. They focused on the challenges of applying Machine and Deep Learning to cybersecurity. They made recommendations for future studies with a short tutorial description of each Machine and Deep Learning method.

Ch et al. introduced a unique classification and feature selection approach in 2021, combining Regression Trees (CART) with Random Forest. The Hybrid Anomaly-based Intrusion Detection System is the name of this system (HAIDS). Instead of using a single method, the hybrid technique improves the model's efficiency. Furthermore, to combat the problem of high dimensionality, eliminating unimportant characteristics is used. The suggested approach was used to pick the top thirteen features from the UNSW-NB15 dataset. With a false alert rate of 11.86 per cent and an accuracy rate of 87.74 per cent, the hybrid technique had the best performance and accuracy.

In their study, Kilincer et al. (2021) discovered that the classifiers employed for all data sets had similar or better performance than the literature. The Decision Tree classifier was more successful than the other classifiers in terms of classifier performance. Dini et al. (2009) demonstrated various machine-learning models to solve anomaly categorisation in LAN traffic monitoring. The use of a K-nearest neighbours (KNN) algorithm and an artificial neural network (ANN) to create an intrusion detection system method (IDS).

Tapsoba et al. (2021) used a binary and multi-class classification model to predict future intrusions for their research project. To compare the best-supervised classification algorithms, they examined and developed K-Nearest Neighbour, decision trees, Logistic Regression, Naive Bayes, Random Forest, Support Vector Machines, and Neural Networks classifiers. The Support Vector Machine classifier proved to be the most effective in predicting intrusions in their study, achieving an accuracy of 80.4%.

Maseer et al. 2021, looked at prior AIDS research and used a set of criteria with various datasets and sorts of assaults to come out with benchmarking results that can disclose the best AIDS algorithms, parameters, and testing criteria. They used ten supervised and unsupervised machine learning techniques to identify effective and efficient ML-AIDS in networks and computers. The artificial neural network (ANN), decision tree (D.T.), k-nearest neighbour (k-NN), naive Bayes (N.B.), random forest (R.F.), support vector machine (SVM), and convolutional neural network (CNN) algorithms are among the supervised machine learning algorithms, while the expectation-maximisation (E.M.), k-means, and self-organising maps (SOM) algorithms are among the unsupervised machine learning algorithms.

Various strategies for detecting assaults on the Internet of Medical Things (IoMT) system were provided by Rbah et al. in 2022. They looked at, compared, and assessed various machine learning (ML) and deep learning (DL)-based mechanisms for preventing and detecting IoMT network assaults, focusing on the proposed methodologies, performance, and limits. Their work revealed potentially available research related difficulties and orientations for designing those systems for the Internet of Medical Things (IoMT) networks based on a detailed review of current defensive security measures.

3. Research Methodology

This chapter details the overall plan and reasoning for the research work. It goes over dataset selection, data preprocessing steps, and evaluation approaches employed in this research.

As stated in the introductory chapter, this study aims to assess five Supervised Machine Learning classification algorithms used in Cyberspace intrusion detection. The selected Machine Learning models would be evaluated on their ability to correctly identify if an event is an intrusion or just routine network traffic termed as "normal".

These models are used to identify the test data's labels. The actual labels are compared to predicted labels, and the evaluation metrics, including the Accuracy, Recall, Precision, F1 Score and Area Under the Receiver Operating Characteristic, are all calculated. The models' performance is compared using these parameters.

Various machine learning techniques have been used to detect and combat these attacks and assaults. The models used in this study are Random Forest, k-Nearest Neighbour, Decision Tree, Support Vector Machines, and Naive Bayes.

The approach adopted for this research is depicted in the figure below.

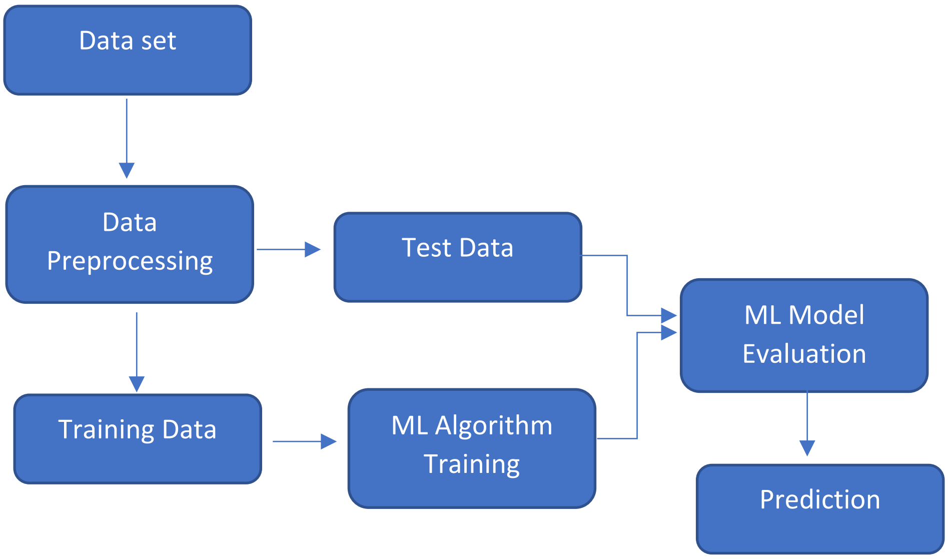


FIGURE 3: MACHINE LEARNING APPROACH.

The entire experiment was carried out using Google Colab resources.

3.1. Experiment Tools and Materials

3.1.1. Google Colab

Google Colab is a cloud-based version of the Jupyter Notebook that provides free computing and dedicated resources such as GPUs and TPUs for personal machine learning projects. The collaboration feature of Google Colab allows for sharing completed projects. Google Colab is equipped with pre-installed data libraries, such as Pandas, NumPy, Matplotlib, Keras, TensorFlow, and PyTorch.

The libraries utilised in this project include the following:

Matplotlib

Matplotlib is a Python library that allows you to create static, animated, and interactive visualisations (Hunter et al., 2007).

- ### Seaborn

Seaborn is a high-level interface for using Matplotlib to create statistical visuals. Its goal is to make visualisation a key component of analysing and comprehending large datasets.

- ### Pandas

Pandas is an open-source, BSD-licensed library for the Python programming language that provides high-performance, easy-to-use data structures and data analysis tools. The library is written in the Python Web framework and is used for numerical data and time-series data manipulation. It defines three-dimensional and two-dimensional data using data frames and series (Pandas, 2022).

- ### Scikit-learn

Scikit-learn is primarily concerned with data modelling topics such as Regression, Classification, Clustering, and model selection. The library is written on the top of Numpy, Scipy, and matplotlib. It's an open-source, commercially usable library that's also simple to grasp. It integrates easily with other machine learning frameworks such as Numpy and Pandas for analysis and Plotly for presenting data in a graphical style for visualisation. Both Supervised and Unsupervised Learning is aided by this resource (Pedregosa et al., 2022).

- ### Numpy

NumPy is an essential Python library for scientific computing.

Numpy is a multi-dimensional data and sophisticated mathematical functions library built on top of an earlier library called Numeric. Numpy is a rapid computing toolkit that can perform various jobs and operations, including introductory algebra, Fourier transformations, random simulations, and shape manipulation.

3.2. Dataset

The lack of representative publicly available datasets constitutes one of the biggest challenges for intrusion detection. The NSL KDD data may not perfectly represent existing real networks. However, it is the closest to actual network traffic and is widely used as an adequate benchmark data set to help researchers compare different intrusion detection methods(Ring et al., 2019). NSL-KDD is a data set proposed to address some of the KDD'99 data set's intrinsic flaws. In the NSL-KDD data set, there are no duplicates or redundant records, and the number of chosen records from each challenging level group is inversely proportional to the percentage of records in the original KDD data set. All of these benefits eliminate the possibility of the classification algorithm producing biased findings.

Furthermore, the NSL-KDD train and test sets have a reasonable quantity of records. This advantage makes it possible to execute the tests on the entire collection without picking a tiny sample at random. Consequently, the assessment outcomes of various research projects will be uniform and comparable. The table below details the data files present in the NSL KDD dataset.

	Data File	Description
1.	KDDTrain+.ARFF	The full NSL-KDD train set with binary labels in ARFF format.
2.	KDDTrain+.TXT	The complete NSL-KDD train set includes attack-type labels and CSV format difficulty levels.
3.	KDDTrain+_20Percent.ARFF	A 20% subset of the KDDTrain+.arff file.
4.	KDDTrain+_20Percent.TXT	A 20% subset of the KDDTrain+.txt file.
5.	KDDTest+.ARFF	The full NSL-KDD test set with binary labels in ARFF format
6.	KDDTest+.TXT	The complete NSL-KDD test set includes attack-type labels and difficulty levels in CSV format.
7.	KDDTest-21.ARFF:	A subset of the KDDTest+.arff file does not include records with the difficulty level of 21 out of 21.
8.	KDDTest-21.TXT	A subset of the KDDTest+.txt file does not include records with the difficulty level of 21 out of 21.

TABLE 2: NSL KDD DATASET.

The table below has an explanation of each attribute as well as a breakdown of the data set.

#	Feature Name	Description	Type	Value Type	Ranges (Between both train and test)
1	Duration	Length of time duration of the connection	Continuous	Integers	0 - 54451
2	Protocol Type	Protocol used in the connection	Categorical	Strings	
3	Service	Destination network service used	Categorical	Strings	
4	Flag	Status of the connection – Normal or Error	Categorical	Strings	

5	Src Bytes	Number of data bytes transferred from source to destination in a single connection	Continuous	Integers	0 - 1379963888
6	Dst Bytes	Number of data bytes transferred from destination to source in a single connection	Continuous	Integers	0 - 309937401
7	Land	If source and destination IP addresses and port numbers are equal then, this variable takes value 1 else 0	Binary	Integers	{ 0 , 1 }
8	Wrong Fragment	Total number of wrong fragments in this connection	Discrete	Integers	{ 0,1,3 }
9	Urgent	The number of urgent packets in this connection. Urgent packets are packets with the critical bit activated	Discrete	Integers	0 - 3
10	Hot	Number of "hot" indicators in the content, such as: entering a system directory, creating programs and executing programs	Continuous	Integers	0 - 101
11	Num Failed Logins	Count of failed login attempts	Continuous	Integers	0 - 4
12	Logged In	Login Status : 1 if successfully logged in; 0 otherwise	Binary	Integers	{ 0 , 1 }
13	Num Compromised	Number of "compromised” conditions	Continuous	Integers	0 - 7479
14	Root Shell	1 if root shell is obtained; 0 otherwise	Binary	Integers	{ 0 , 1 }
15	Su Attempted	1 if "su root'' command attempted or used; 0 otherwise	Discrete (Dataset contains ‘2’ value)	Integers	0 - 2
16	Num Root	Number of "root" accesses or number of operations performed as a root in the connection	Continuous	Integers	0 - 7468
17	Num File Creations	Number of file creation operations in the connection	Continuous	Integers	0 - 100
18	Num Shells	Number of shell prompts	Continuous	Integers	0 - 2

19	Num Access Files	Number of operations on access control files	Continuous	Integers	0 - 9
20	Num Outbound Cmds	Number of outbound commands in an ftp session	Continuous	Integers	{ 0 }
21	Is Hot Logins	1 if the login belongs to the "hot'' list i.e., root or admin; else 0	Binary	Integers	{ 0 , 1 }
22	Is Guest Login	1 if the login is a "guest'' login; 0 otherwise	Binary	Integers	{ 0 , 1 }
23	Count	Number of connections to the same destination host as the current connection in the past two seconds	Discrete	Integers	0 - 511
24	Srv Count	Number of connections to the same service (port number) as the current connection in the past two seconds	Discrete	Integers	0 - 511
25	Serror Rate	The percentage of connections that have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in count (23)	Discrete	Floats (hundredths of a decimal)	0 - 1
26	Srv Serror Rate	The percentage of connections that have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in srv_count (24)	Discrete	Floats (hundredths of a decimal)	0 - 1
27	Rerror Rate	The percentage of connections that have activated the flag (4) REJ, among the connections aggregated in count (23)	Discrete	Floats (hundredths of a decimal)	0 - 1
28	Srv Error Rate	The percentage of connections that have activated the flag (4) REJ, among the connections aggregated in srv_count (24)	Discrete	Floats (hundredths of a decimal)	0 - 1
29	Same Srv Rate	The percentage of connections that were to the same service among the links aggregated in count (23)	Discrete	Floats (hundredths of a decimal)	0 - 1
30	Diff Srv Rate	The percentage of connections that were two different services	Discrete	Floats (hundredths of a decimal)	0 - 1

		among the links aggregated in count (23)			
31	Srv Diff Host Rate	The percentage of connections that were too different destination machines among the links aggregated in srv_count (24)	Discrete	Floats (hundredths of a decimal)	0 - 1
32	Dst Host Count	Number of connections having the same destination host IP address	Discrete	Integers	0 - 255
33	Dst Host Srv Count	Number of connections having the same port number	Discrete	Integers	0 - 255
34	Dst Host Same Srv Rate	The percentage of connections that were to different services, among the connections aggregated in dst_host_count (32)	Discrete	Floats (hundredths of a decimal)	0 - 1
35	Dst Host Diff Srv Rate	The percentage of connections that were too different services among the links aggregated in dst_host_count (32)	Discrete	Floats (hundredths of a decimal)	0 - 1
36	Dst Host Same Src Port Rate	The percentage of connections that were to the same source port among the links aggregated in dst_host_srv_count (33)	Discrete	Floats (hundredths of a decimal)	0 - 1
37	Dst Host Srv Diff Host Rate	The percentage of connections that were to different destination machines, among the connections aggregated in dst_host_srv_count (33)	Discrete	Floats (hundredths of a decimal)	0 - 1
38	Dst Host Serror Rate	The percentage of connections that have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in dst_host_count (32)	Discrete	Floats (hundredths of a decimal)	0 - 1
39	Dst Host Srv Serror Rate	The percent of connections that have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in dst_host_srv_count (33)	Discrete	Floats (hundredths of a decimal)	0 - 1

40	Dst Host Rerror Rate	The percentage of connections that have activated the flag (4) REJ, among the connections aggregated in dst_host_count (32)	Discrete	Floats (hundredths of a decimal)	0 - 1
41	Dst Host Srv Rerror Rate	The percentage of connections that have activated the flag (4) REJ, among the connections aggregated in dst_host_srv_count (33)	Discrete	Floats (hundredths of a decimal)	0 - 1
42	Class	Classification of the traffic input	Categorical	Strings	

TABLE 3: FEATURES IN THE DATASET.

3.3. Algorithms

3.3.1. Decision Trees

Decision Trees (D.T.s) are a non-parametric supervised learning approach for Classification and Regression. The objective is to learn basic decision rules from data attributes to develop a model that predicts the value of a target variable. A tree is an approximation of a piecewise constant.

Decision trees are robust algorithms that may be utilised in various domains, including machine learning, image processing, and pattern recognition. They are a sequential model that effectively and cohesively connects a series of fundamental tests in which a numeric characteristic is compared to a threshold value in each trial. Each tree comprises nodes and branches. Each subset specifies a value that the node can take. Each node represents features in a category to be categorised, and each subgroup defines a decision that the node can take. Decision trees have a wide range of applications due to their easy analysis and precision across numerous data types (Charbuty et al., 2021).

The figure below illustrates the structure of a decision tree.

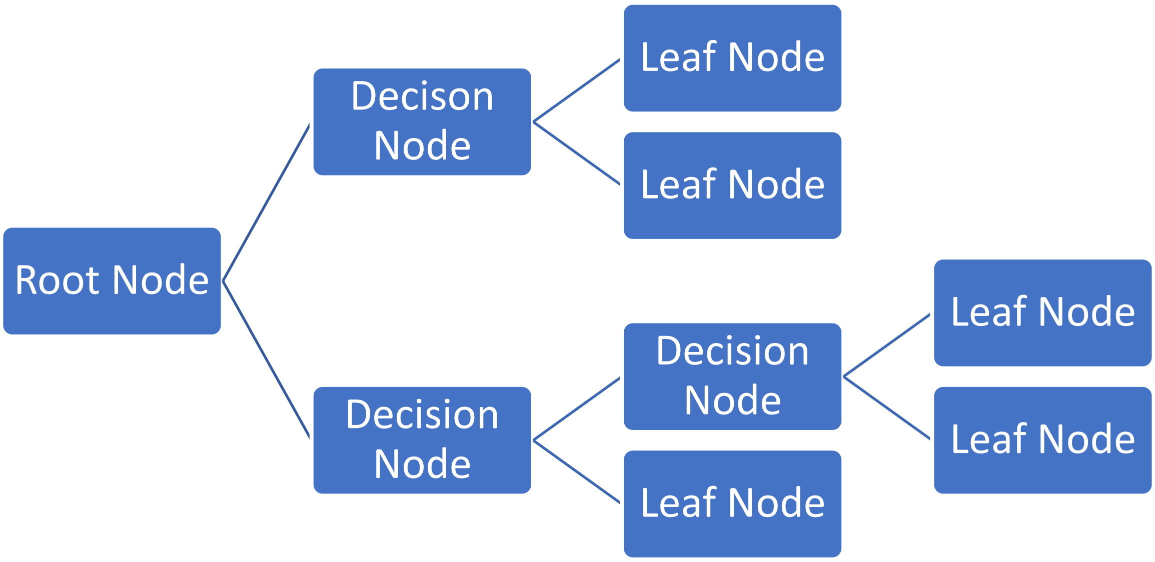


FIGURE 4: DECISION TREES.

Decision trees can be categorical, they require minimal to no data preparations, and their clarity and interpretability make them a popular choice. Other models frequently need feature scaling, the creation of dummy variables, and the elimination of null or missing values.

3.3.2. Gaussian Naïve Bayes

The Naïve Bayes algorithm is a simple approach for making predictions that use the Bayes rule to determine the probability of each character belonging to each class. It simplifies probability calculations by assuming that the features are independent based on the labels of the other attributes. The conditional probabilities are the probabilities of a class value if the attribute value is known. Naïve Bayes algorithms have been surprisingly accurate for classification tasks in several studies, albeit with smaller datasets. The probability of data instances may be determined by multiplying their attribute conditional probabilities. Calculating the probabilities of each class occurrence and picking the highest probability class value can be used to make predictions.

The simple form of the calculation for the Bayes Theorem is as follows:

$$P(A|B) = P(B|A) * P(A) / P(B)$$

The posterior probability P(A|B) is known as the posterior probability, while the marginal probability of the occurrence P(A) is recognised as the prior probability.

3.3.3. K-Nearest Neighbours (k-NN)

K-nearest neighbours (k-NN) is a pattern recognition technique that finds the k closest relatives in the following situations using training datasets. The K-NN technique is straightforward to comprehend and implement. It searches the whole dataset for K's nearest neighbours to categorise

The method employs 'feature similarity" to estimate the values of new data points. The procedure assigns an unknown classification to an input sample vector y by allocating it to its nearest neighbour K class.

Belgrana et al. (2021) explained that it could be expanded to include immediate neighbours. The vector is assigned to the class with the most significant number of nearest neighbours. When the number of nearest neighbours between classes is equal, the lowest distance is used as the judge.

3.3.4. Random Forests

The Random Forest Algorithm is a prominent algorithm of Machine Learning that is well-known for its simplicity and efficacy. It may be categorised as a Decision Tree-Based Classifier that uses voting to select the best classification tree as the algorithm's final Classification. Because of its excellent characteristics, such as Variable Importance Measure, Out-of-bag error, Proximities, and others, Random Forest is the most often used group classification system. There is a wide range of applications in image processing right now, including intrusion detection, content information filtering, and sentiment analysis (Abdulkareem, 2021).

Random forest combines many decision trees (referred to as the forest) to provide a more accurate and consistent forecast. The forest it creates is made up of Decision Trees trained using the bagging approach. The figure below illustrates the structure of a Random Forest.

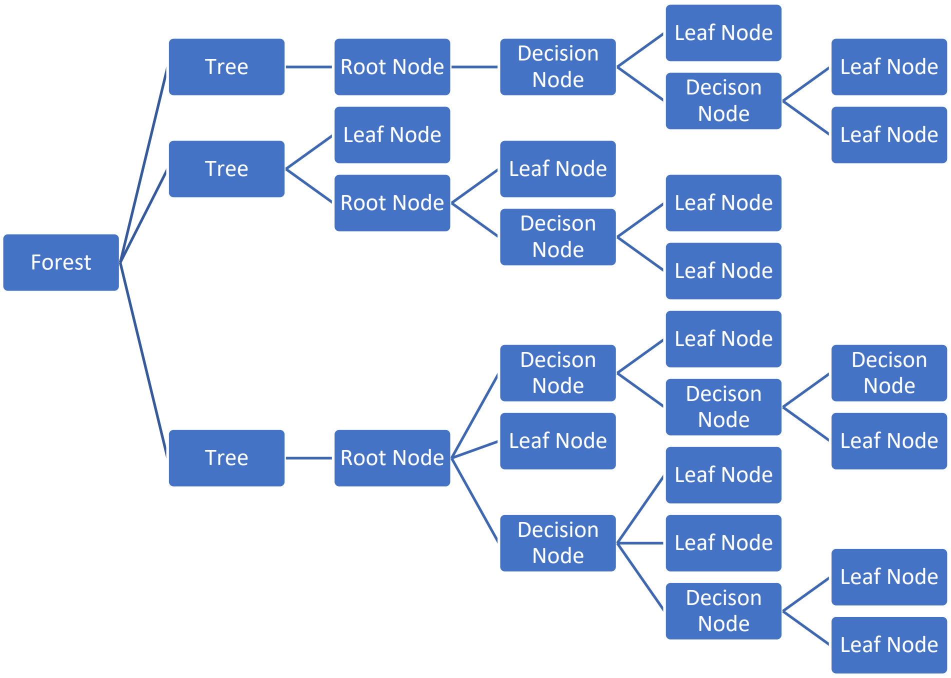


FIGURE 5: RANDOM FORESTS.

3.3.5. Support Vector Machines

Telles et al. in 2021 described Support Vector Machines (SVMs) as a class of machine learning algorithms that distinguish between variables that a distinct hyperplane has determined at the outset. In plain terms, the algorithm takes labelled training data as input and returns an optimum hyperplane classifying the data based on objective criteria as output. This hyperplane is a line that divides a two-dimensional plane into two classes. Support vector machines are effective in high-dimensional spaces and when the number of dimensions exceeds the number of samples. Face recognition, target recognition, object identification, speaker identification, and handwritten digit recognition are just a few machine learning applications that employ Support Vector Machines.

The image illustrates how Support Vector Machines operate. (Telles et a., 2021).

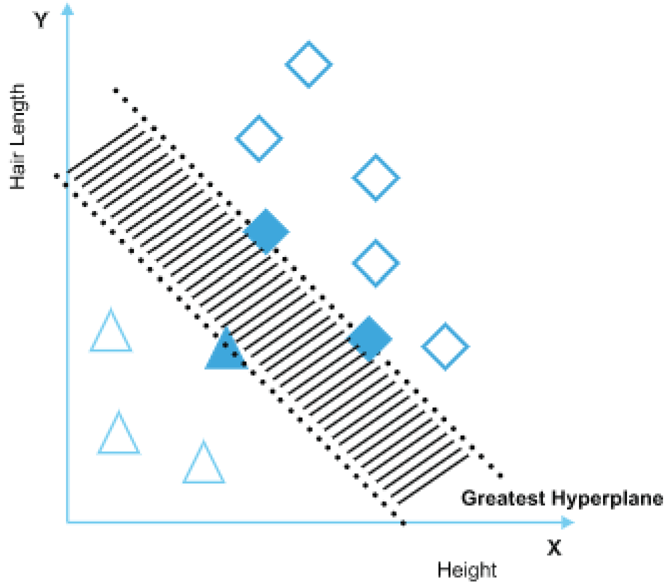


FIGURE 6: SUPPORT VECTOR MACHINES.

3.4. Data Preprocessing

In Machine Learning, data preprocessing is a critical step that improves the data quality and facilitates the Extraction of relevant insights. In Machine Learning, data preprocessing refers to organising and managing raw data to make it appropriate for creating and training Machine Learning models. In basic terms, data preprocessing is a data collection approach used in Machine Learning that turns raw data into a legible and intelligible format.

One of the advantages NSL-KDD dataset is that it has no severe data quality issues like missing values in both training and the test sets, as shown in the images below. There will, however, be some processing required.

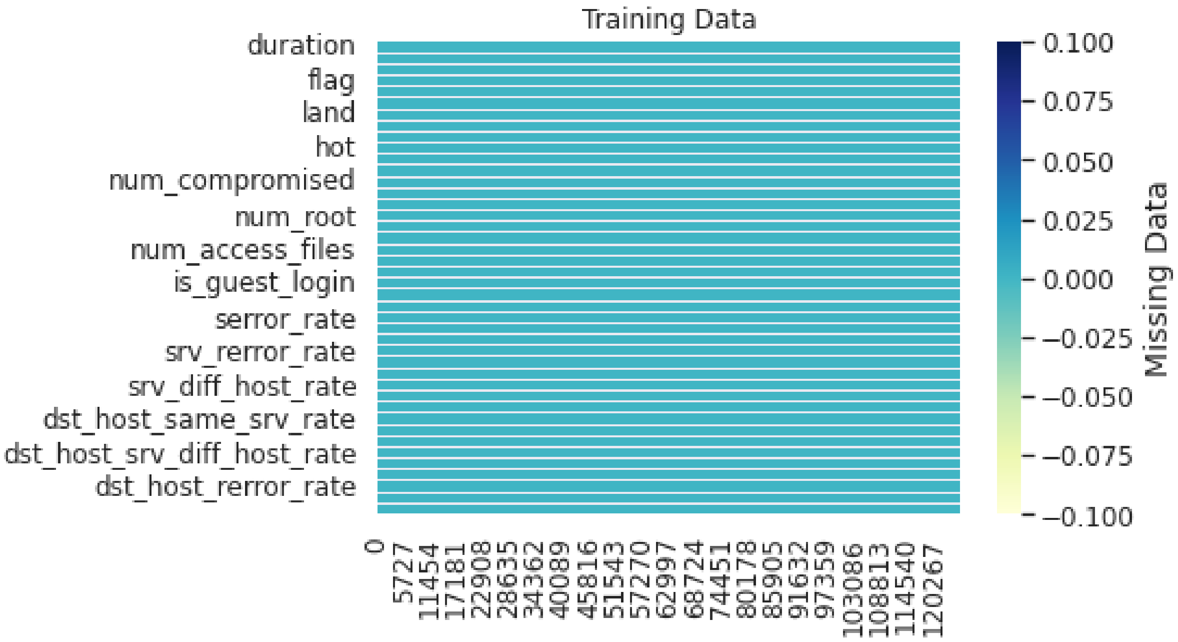


FIGURE 7: CHECK FOR MISSING VALUES IN TRAINING DATA.

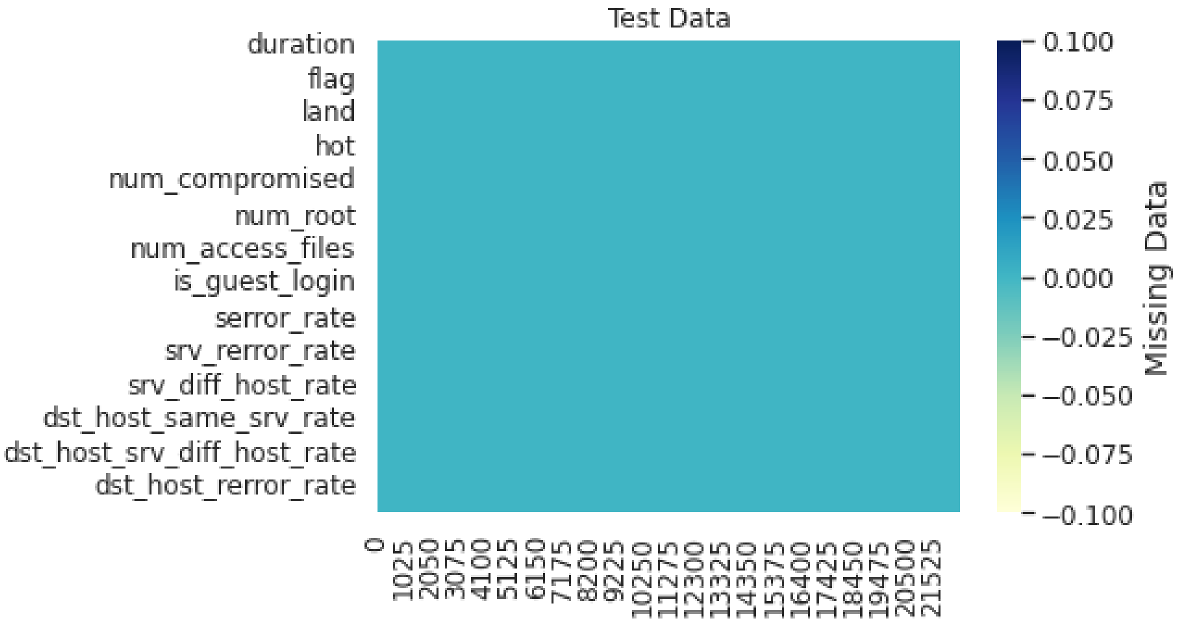


FIGURE 8: CHECK FOR MISSING VALUES IN TEST DATA.

The NSL KDD dataset still required some preprocessing even in its pristine state. The preprocessing steps applied in this project are as follows:

3.4.1. Feature Extraction

This step involves separating the dependent variable from the independent variables. The dependent variable is the "Class" feature in the NSL KDD Training and Test data.

The step is necessary for Supervised Learning.

3.4.2. Feature Scaling

Feature scaling is a data preprocessing application that utilises a set range to standardise the independent attributes found in the data. The scaling technique selected for this research is *Standardisation*. It is a very effective technique that re-scales a feature value to have a distribution with 0 mean value and variance that equals 1. Feature scaling is critical since various machine learning algorithms require it to produce accurate results. The range of features significantly impacts distance algorithms like KNN and SVM, which use distances between data points to assess similarity behind the scenes.

3.4.3. Data Transformation

Data transformation is used chiefly to convert numerical data into categorical data to ensure interoperability with ML algorithms.

Categorical variables are variables within a dataset that have specified categories. The values of the characteristics "protocol type," "service", and "flag" are transformed into discrete values using the Label Encoder Function for accurate classifier performance.

The class feature in the NSL-KDD training and test dataset contains 24 different attacks, as shown in the images below.

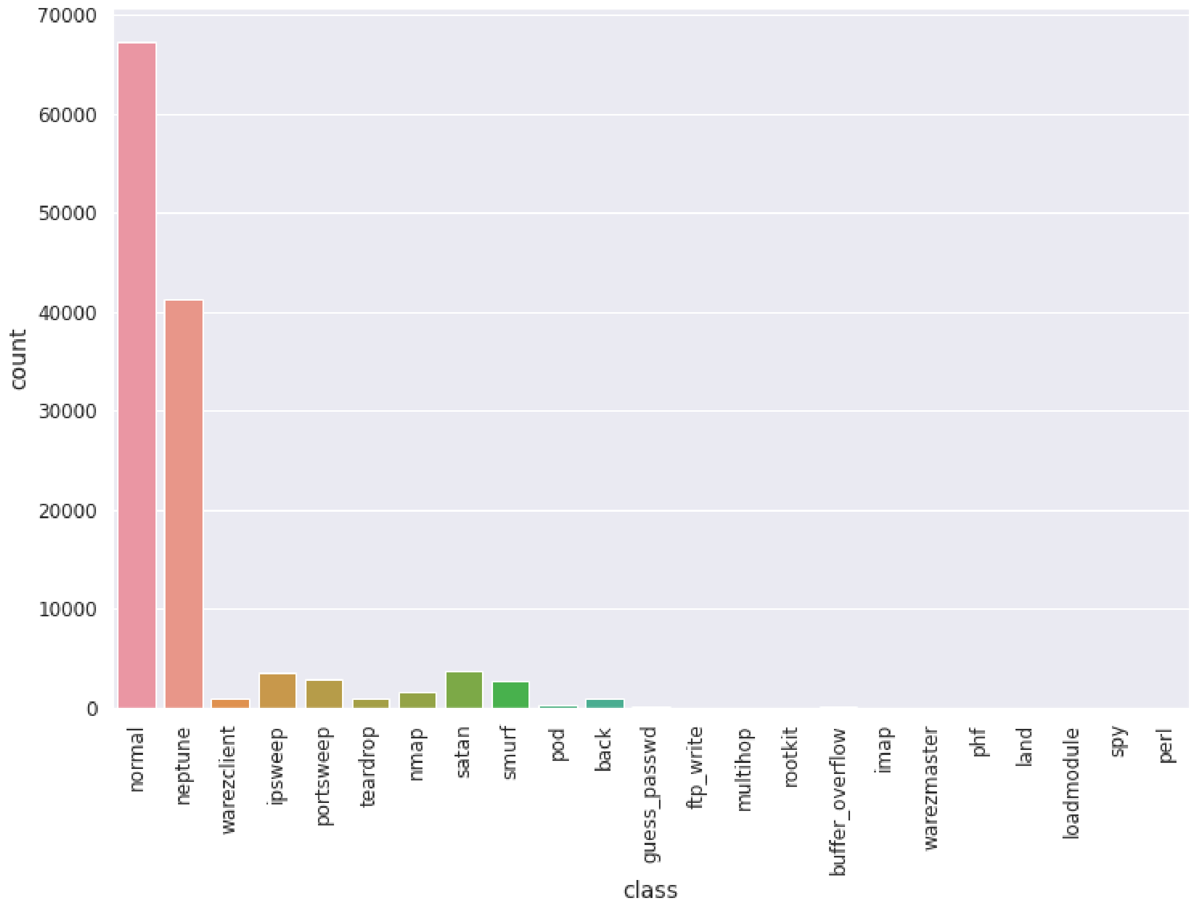


FIGURE 9: SUBCLASSES OF INTRUSION.

There are four types of intrusions in the data set: denial of service (DoS), probe, User to Root (U2R), and Remote to local (R2L). The following is a brief description of each attack:

- Denial of Service (DoS) is an intrusion that attempts to stop traffic from and to the target system. The IDS receives an unusual volume of traffic that it cannot handle, and it shuts down to protect itself. This makes it impossible for typical traffic to access a network. An online business might experience a surge in online orders on a significant sale day. Because the network cannot handle all requests, it would shut down, preventing paying consumers from making purchases. In the data set, this is the most common attack.
- A probe or surveillance attack attempts to obtain data from a network. The purpose is to impersonate a hacker and steal vital information, such as client personal information or banking information.
- User to Root is an intrusion that starts with a regular user account and attempts to achieve super-user access to the system or network (Root). The attacker attempts to get root privileges/access by exploiting system vulnerabilities.
- Remote to Local (R2L) is a method of gaining local access to a remote machine. An attacker who does not have local access to the system/network attempts to "hack" their way through.

The intrusion and their subclasses are listed in the table below:

Class	**Subclass**
Denial of Service (DoS	Apache2, Back, Land, Neptune, Mailbomb, Pod, Processtable, Smurf, Teardrop, Udpstorm, And Worm.
Probe	Ipsweep, Mscan, Nmap, Portsweep, Saint, And Satan.
User to Root	Buffer Overflow', Loadmdoule, Perl, Ps, Rootkit, Sqlattac, And Xterm.
Remote to Local (R2L)	Ftp_Write, Guess_Passwd, Http_Tunnel, Imap, Multihop, Named, Phf, Sendmail, Snmpgetattack, Snmpguess, Spy, Warezclient, Warezmaster, Xclock, and Xsnoop.

TABLE 4: CLASSES OF INTRUSION.

As seen in the graphic below for both the training and test datasets, the attack categories have now been transformed into two features, with "0" denoting "normal" and "1" indicating an intrusion. The study aims to identify cyberspace intrusions, not types of intrusions.

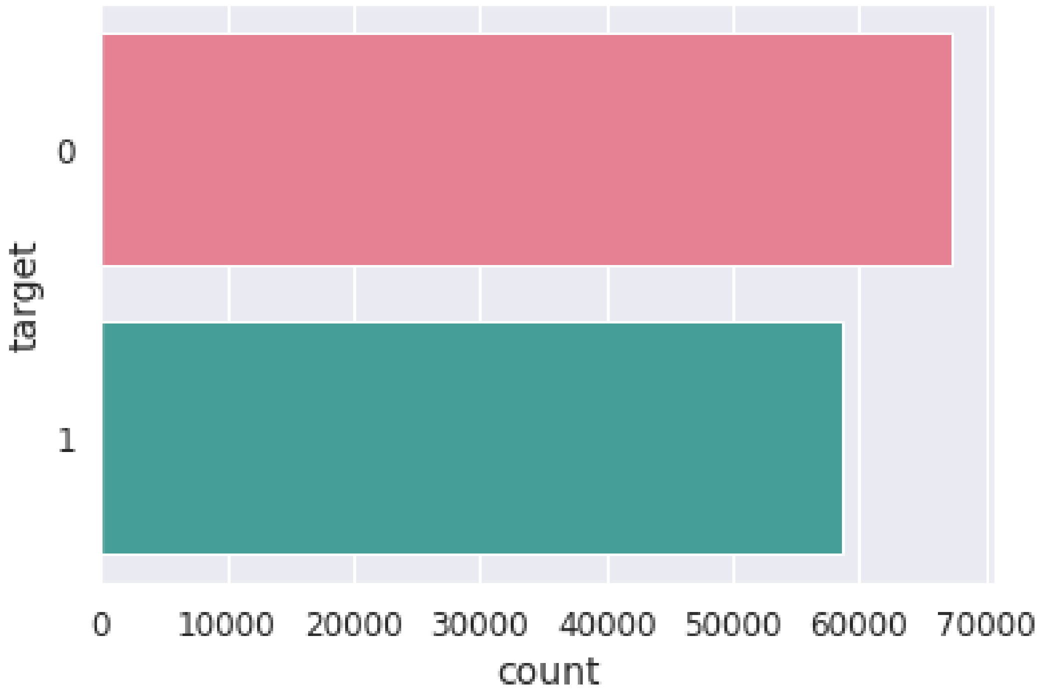


FIGURE 10: COUNT OF CLASSES IN THE TRAINING TARGET FEATURE.

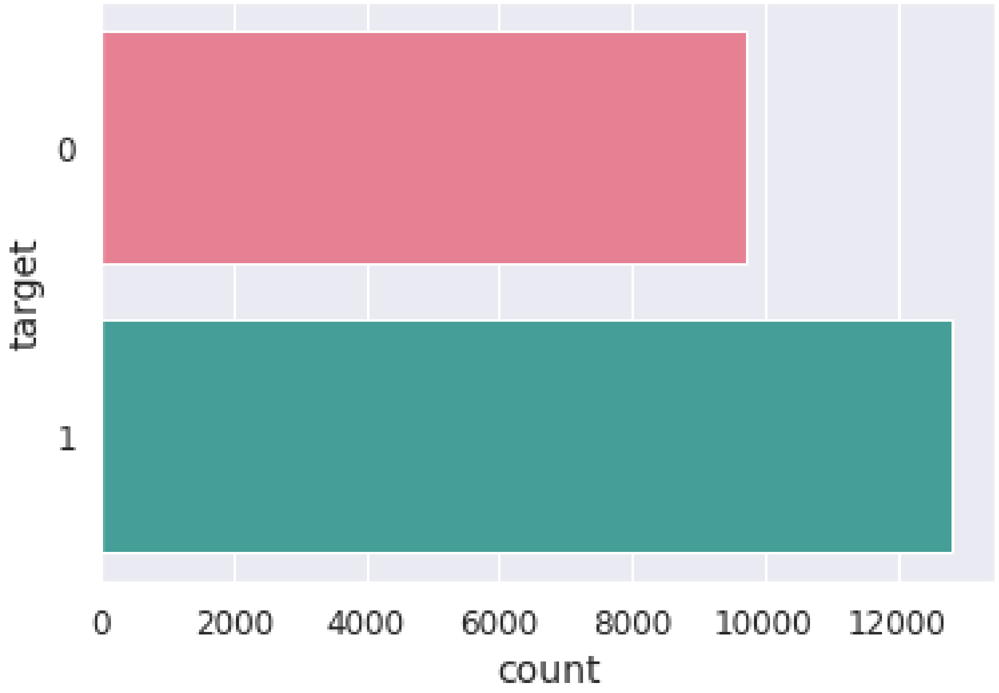


FIGURE 11: COUNT OF CLASSES IN THE TEST TARGET FEATURE.

Judging by the plot above, it can be seen that an event intrusion happened 47% of the time in all of the data in the training data set.

3.5. Modelling

In this phase, the Machine learning models are trained with training data. The training phase is one of the most crucial machine learning processes. The prepared data is fed to machine learning models to detect patterns and create predictions. Consequently, the models learn from the data and can complete the assigned task. The models improve in predicting over time as they are trained.

3.6. Evaluation of Machine Learning Models

The models are evaluated on previously unseen data once they have been trained. The test data set utilised here is the testing set that has been pre-split from the NSL KDD data before. If testing is done on the same data that is used for training, the results of the models would not be reliable since the models are already familiar with the data and would see the same patterns that they identified previously. These models make predictions with the test data and are evaluated using the following metrics.

3.6.1. Performance Metrics

3.6.2. Precision

The precision score indicates what percentage of positively projected labels are correct. The positive predictive value is another term for precision. Precision decreases if there are more samples in the minority class. Precision can be considered a metric for how reliable a model is. A model with high precision should be used if the goal is to reduce false negatives.

In contrast, a model with a high recall should be used if the aim is to reduce false positives. Precision is employed when there is a need to predict the positive class. The cost of false positives is higher than false negatives, such as in medical diagnoses or spam filtering. For example, if a model is 99 % accurate but only 50 % precise, it means that half of the time, it correctly predicts that an email is spam, but it is not. Precision is utilised in conjunction with recall when trading false positives (F.P.) and false negatives(F.N.). The class distribution always has an impact on precision.

The precision score is the division of all True Positives (T.P.) by adding all True Positives and All False Positives (F.P.). T.P. stands for True Positives when a model successfully classifies a data item into the correct class.

F.P. stands for False Positives, which occur when the model incorrectly classifies a data item.

$$P = TP / TP + FP$$

3.6.3. Recall

The recall score represents the model's ability to predict positives out of real positives correctly. Recall differs from precision, which evaluates how many positive predictions a model makes out of all positive predictions. If the goal is to find favourable reviews, the recall score is the percentage of positive reviews that an algorithm accurately identifies as positive. It assesses a machine learning model's ability to identify all true positives among all possible positives in a dataset. The better the machine learning model identifies both positive and negative samples, the higher the recall score. Recall is frequently combined with additional performance metrics like precision and accuracy to provide a complete picture of the model's performance.

The Recall Score is the division of All True Positives by adding all True Positive and all False Negatives.

R = TP / TP + FN

3.6.4. F1 – Score

The F1 score is a precision and recall function representing a model's score. It's a performance metric that weighs Precision and Recall equally when assessing accuracy, making it a viable alternative to Accuracy metrics. However, it does not require the user to know the total number of observations. It is frequently used as a single value that offers high-level information regarding the quality of the model's output. This is a valuable model measure in situations where one tries to maximise either precision or recall score, and the model performance suffers as a result.

The F1 score is the mean of an algorithm's performance based on precision and recall.

F1 = 2((precision*recall) / (precision+recall))*

3.6.5. Accuracy

The accuracy of a machine learning model is a metric for determining which model is the best at recognising patterns and correlations between variables in a dataset given input or training data. The stronger a model's generalisation to 'unseen' data is the more reliable predictions and insights it can provide and the more value it can give. Errors have a high cost, but improving model accuracy lowers that cost. Although there is a point at which the benefit of constructing a more accurate model does not result in a matching rise in making correct predictions, it is frequently advantageous across the board.

Accuracy is the addition of all True Positives and True Negatives over the addition of True Positives, All True negatives, False Positives and False Negatives.

A = TP+ TN / TP + TN + FT+FN

3.6.6. Area Under the Receiver Operating Characteristics

AUROC is a "discrimination" performance statistic that measures a model's ability to distinguish between cases (positive instances) and non-cases (negative examples.) An AUROC of 0.8 indicates that the model can discriminate adequately. A model would correctly assign a more significant absolute risk to a randomly picked class with an event 80% of the time than to a randomly selected class without an event.

The ROC curve is plotted with Sensitivity against the Specificity, where Sensitivity is on the y-axis and Specificity is on the x-axis.

- **Sensitivity**

  Sensitivity refers to the chance of a positive test, conditioned on being positive, also known as the True Positive Rate.

Sensitivity (TPR) = TP/(TP +FN)

- **Specificity**

Specificity refers to the chance of a negative test, provided it is negative, also known as the True Negative Rate.

Specificity (FPR) = TN/(TN+FP)

The area under the ROC curve is used to determine the AUC. A ROC curve depicts the trade-off between SensitivitY and Specificity across multiple judgment thresholds.

Discussion and Evaluation of Findings

This section details the results of evaluating the performance of five supervised machine learning algorithms on the Nsl KDD intrusion data set. The algorithms' performance is assessed on their ability to classify an event as routine traffic or an Intrusion into the network with five different metrics: accuracy, precision, recall, and the area under the ROC curve.

Models	**Train Accuracy**	**Accuracy Score**	**F1 Score**	**Precision**	**Recall**	**AUC**
Random Forest	0.99	0.78	0.78	0.90	0.68	0.79
Decision Trees	0.99	0.79	0.81	0.86	0.76	0.80
Naïve Bayes	0.89	0.57	0.73	0.57	0.99	0.50
K-nearest Neighbours	0.99	0.77	0.76	0.97	0.62	0.86
Support Vector Machines	0.99	0.82	0.81	0.98	0.70	0.84

TABLE 5: RESULTS.

In terms of accuracy, the selected classification models recorded near-perfect scores, averaging 99% when trained on the training data and underperformed when pitted against the test data (averaging above 70%). The Naïve Bayes model was the only exception in this category, as it poorly performed when tested on the unseen data, with a training score of 89 % and a test score of 56%. The scores suggest that the models overfitted the training data. However, the accuracy score is not the only metric used in judging the performance of a classifier, as it never paints a complete picture. Accuracy tells how often a machine learning model will correctly predict an outcome out of the number of times it makes predictions. Still, it provides some measure of telling what can be expected from other metrics. The Accuracy Scores of the models are visually represented in the figure below.

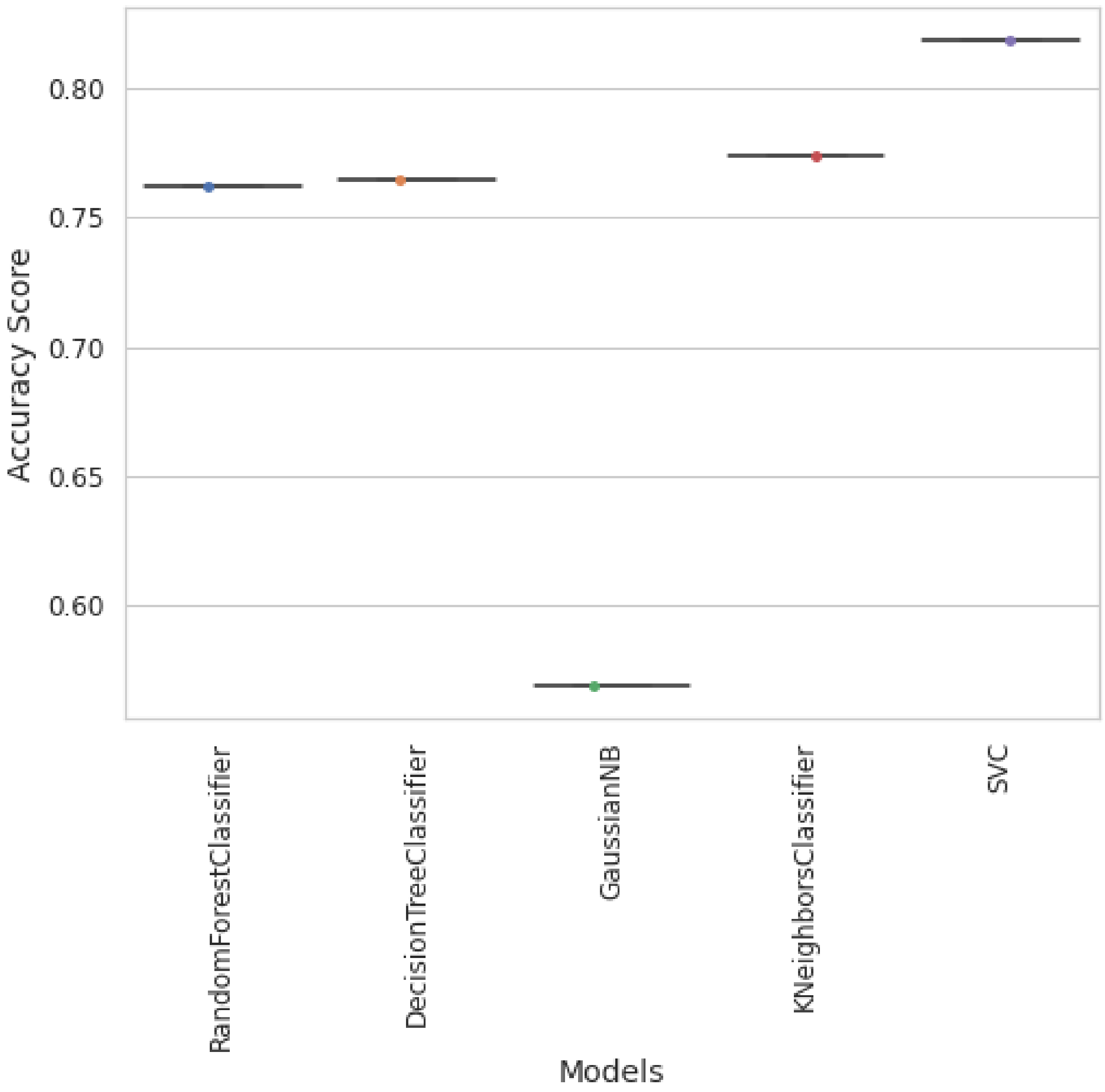


FIGURE 12: ACCURACY OF CLASSIFIERS.

The Naïve Bayes model was the only exception in this category, as evident in the boxplot above. It produced a below-par performance when tested on the unseen data, with a score of just under 60%.

Regarding the F1, the Random Forest and Support Vector Machines obtained the highest average F1 value of 81%. In contrast, the model with the lowest was the Naive Bayes Classifier with 78%, as shown in the figure below.

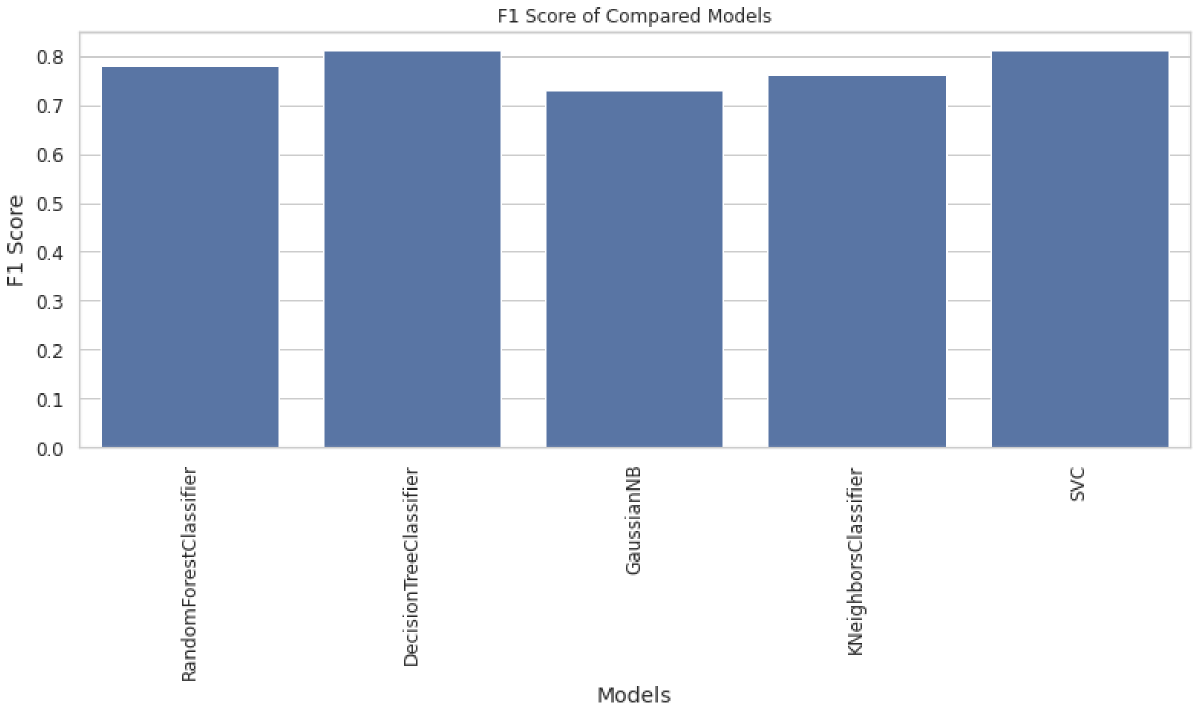


FIGURE 13: F1- SCORE OF CLASSIFIERS.

The F1 score computes the weighted mean of recall and precision by adding them to get a single figure. It determines how exactly a model is at deciding how many cases it correctly classifies and how robust it is by ensuring that it does not miss a large number of instances. This suggests that the Naïve Bayes was the least exact in its predictions.

In terms of Precision, Four of the selected classifiers recorded average precision scores with Support Vector Machines, K-nearest Neighbour, Random Forrest scoring 0.98, 0.97 and 0.90, respectively, and Decision Forrests fell slightly behind at 0.86. Nayes Bayes recorded the lowest score at 0.57.

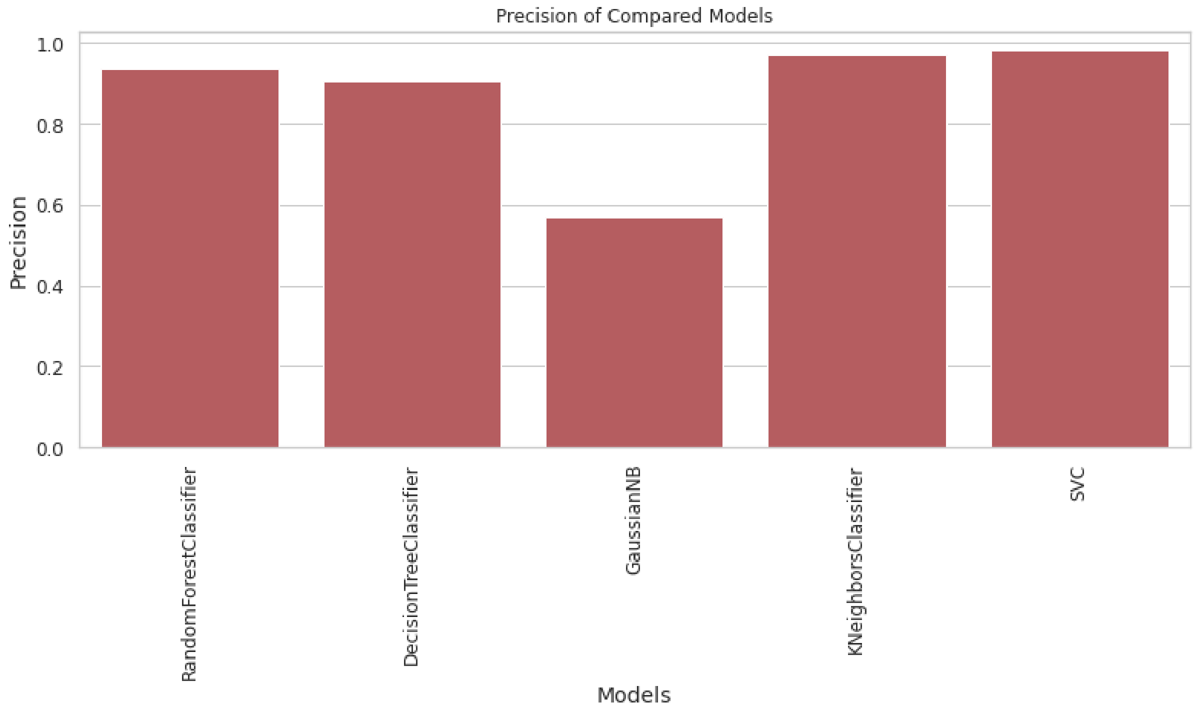


FIGURE 14: PRECISION OF CLASSIFIERS.

Recall is a metric for how well models detect True Positives. As a result, recall reveals how many intrusions have been accurately recognised for all of the events in the network. The figure below shows that the Naïve Bayes recorded the highest average recall with a 0.99 score, while the rest of the classifiers went above 0.6.

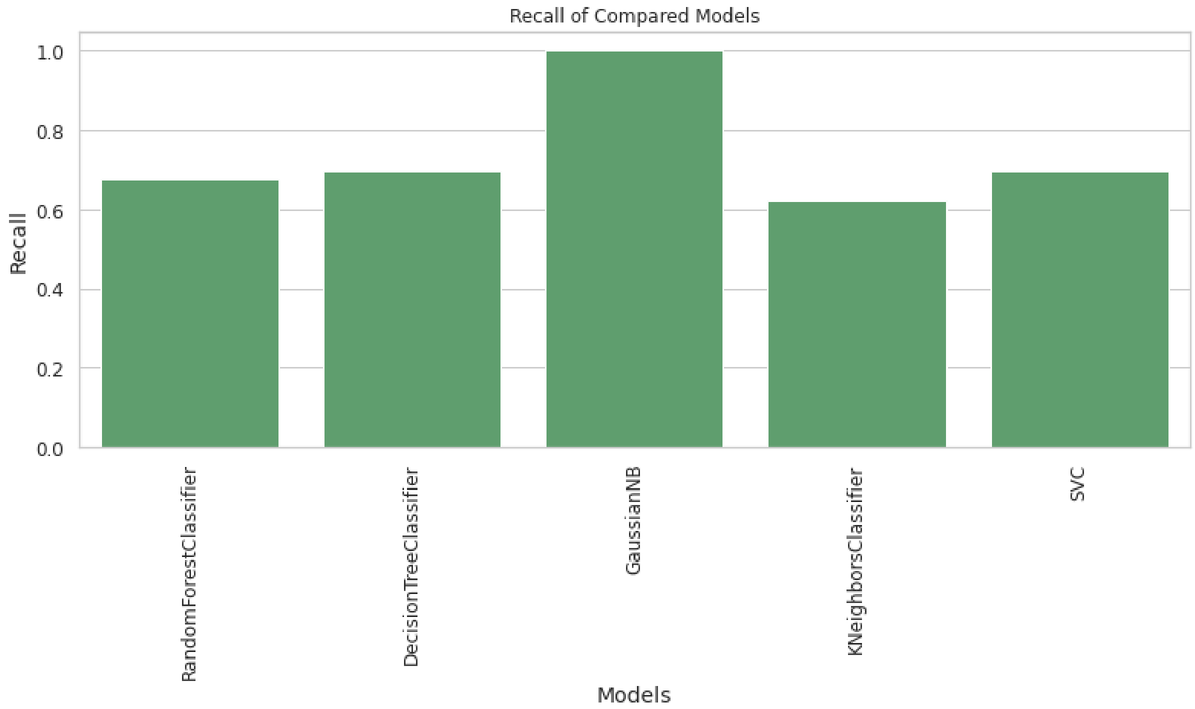


FIGURE 15: RECALL OF CLASSIFIERS

The ideal situation is one in which both precision and recall are high. Intuitively, this indicates that when the models predict a "normal" occurrence, which is the most likely, the models will almost always be correct. However, when a model has high precision but a poor recall, it suggests that it is very selective in its predictions. The algorithms elected not to take the risk of predicting an inaccurate event when an event was challenging to classify. This suggests that the models are, more frequently than not, right (high precision) when they predict an occurrence, but not the other way around with low recall. Precision and recall are constantly at odds. If a model is made very selective, it will have higher precision but lower recall, and vice versa.

Another performance metric considered for this research is the AUC - ROC Curve. The AUC - ROC curve is a performance metric for classifying events at various thresholds. AUC indicates the degree or measure of distinction, whereas ROC is a probability curve. It demonstrates how well the model can discriminate between classes.

The area under the Receiver Operating Characteristics (ROC) curve is used to determine the quality of classification models.

The False Positive Rate is shown against the True Positive Rate on the graph. The AUC for the classifiers stated above is represented in the diagram below.

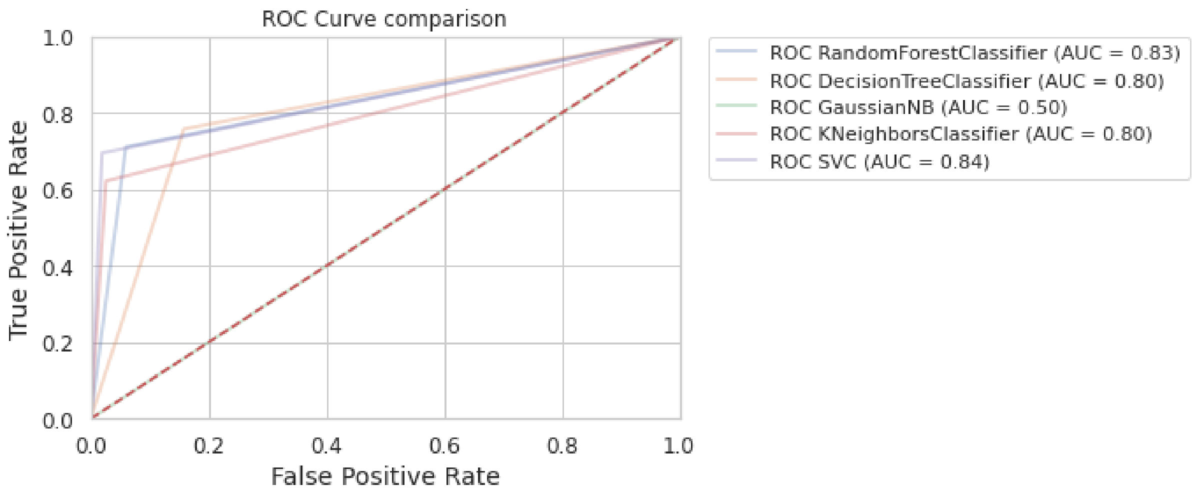


FIGURE 16: ROC- AUC CURVE OF CLASSIFIER

Support Vector Machines had the best AUC score of 0.84, followed by Random Forest at 0.83. The Naive Bayes classifier came in last with 0.5, while the K-Nearest Neighbour and Decision trees classifiers both returned identical figures of 0.80.

The more significant the AUC, the better a model's ability to predict classes present in a dataset. The model can distinguish between a regular event in a network's traffic and an intrusion, suggesting four selected algorithms produced acceptable results.

The Support Vector Machine classifier outperformed the other classifiers in every evaluation category as it recorded more success in detecting intrusion, making it the best of the lot.

Comparing this research's evaluation to Other authors have recorded similar success in detecting intrusions by applying Supervised Machine learning Algorithms to the NSL KDD dataset.

Machine learning has become an essential technique to secure Cyberspace as a whole. They can swiftly analyse millions of activities and detect a wide range of threats, from malware that exploits zero-day vulnerabilities to dangerous behaviour that might lead to Phishing or the download of malicious code. These systems improve with time, relying on previous experiences to recognise new events in the present. User, asset, and network profiles are created using behaviour histories, allowing A.I. to identify and respond to departures from established norms.

Intrusion Detection systems may now use machine learning to analyse patterns and learn from them to help prevent repeated assaults and adapt to changing behaviour. Security analysts have effectively employed machine learning to develop effective intrusion detection capabilities.

Machine Learning has given cybersecurity teams the ability to be more proactive in avoiding risks and responding to ongoing assaults in real-time.

IBM Security QRadar XDR, Vectra, and MicroAI are examples of ML tools for cybersecurity.

Conclusion and Future Work

The performance of IDS improvement is dependent on a variety of machine learning approaches. For Intrusion Detection Systems to discriminate between different types of assaults, classification algorithms play a critical role. The findings of an evaluation of the performance of five supervised algorithms for detecting intrusion in network traffic are discussed in this research: Nave Bayes, support vector machine, Random Forest, decision tree, and K-nearest neighbour. These algorithms were evaluated using Accuracy, Recall, precision, F1 and AUC using the NSL-KDD dataset.

The Nsl KDD dataset comprises 24 types of assaults. The study's goal is to distinguish between intrusions and regular network activity. The classes were binned into "Normal" for regular activity and "intrusion" for intrusions, making them binary classification jobs.

The results show that the Support Vector Machine Classifier obtained the best identification of intrusion and routine network traffic with an average accuracy of 76%, an AUC of 0.84, an F1 score of 81%, and a Recall average of 70% o over the rest of the classification algorithms. The Decision Tree Classifier came closest as it recorded an average accuracy score of 79%, 81% F1 score, 86% Precision, a recall average of 76% and an AUC of 0.83.

Though training and fit time were not considered as part of the chosen evaluation metrics, it was noted that K-Nearest Neighbour, Random Forest and Support Vector Machine took a lot of time producing results. This is due to the limited resources available during experimentation. On the free plan, Google Colab proved to have restricted resources. The free GPU instances provided by Colab, most often K80 GPUs, are typically underpowered. As instances disconnected often, connectivity was inconsistent. And instances frequently lack sufficient RAM, which impacts computation time when working with massive datasets like the NSL KDD.

One future focus should be to utilise dedicated resources for running experiments like this; it could help cut the computation time short. Cross-Validation using various N-fold settings may aid in improving the rate of detection and distinguishing between regular network activity and intrusions.

The properties of our dataset may not help detect new intrusion instances due to the quick and constant evolution of cyber assaults. As a result, deep learning methods can be valuable alternatives for detecting and identifying the traffic of new occurrences of intrusions because these approaches are constructed internally as part of the complicated and hierarchical process of Deep Learning, rather than relying on predetermined characteristics.

Ensemble learning approaches should also be considered since, when compared to individual models, they have a better prediction accuracy.

They are especially beneficial when the dataset contains linear and non-linear data; several models may be coupled to manage it.

Bibliography

El Naqa, Issam & Murphy, Martin. (2015). What Is Machine Learning? 10.1007978-3-319-18305-3_1.

A. R. Tapsoba and T. Frédéric OUEDRAOGO (2021) *Evaluation of supervised learning algorithms in binary and multi-class network anomalies detection.* pp. 1.

Alhawi, O.M., Baldwin, J. and Dehghantanha, A. (2018) 'Leveraging machine learning techniques for windows ransomware network traffic detection' *Cyber threat intelligence* Springer, pp. 93-106.

Anti-Phishing Working Group (2021) *PHISHING ACTIVITY TRENDS REPORTS.* Available at: https://apwg.org/trendsreports/ (Accessed: 12 December, 2021).

Azeroual, O. and Nikiforova, A. (2022) 'Apache Spark and MLlib-Based Intrusion Detection System or How the Big Data Technologies Can Secure the Data', *information,* 13(2), pp. 58.

Barker, E. and Kelsey, J. (2012) 'NIST DRAFT Special Publication 800-90b recommendation for the entropy sources used for random bit generation', .

Basnet, R., Mukkamala, S. and Sung, A.H. (2008) Detection of phishing attacks: A machine learning approach'*Soft computing applications in industry* Springer, pp. 373-383.

Chauhan, P. and Chandra, N. (2013) 'A Review on Hybrid Intrusion Detection System using Artificial Immune System Approaches', *International journal of computer applications,* 68(20), pp. 22-27. doi: 10.5120/11695-6499.

F. Z. Belgrana, N. Benamrane, M. A. Hamaida, A. Mohamed Chaabani and A. Taleb-Ahmed (2021) *Network Intrusion Detection System Using Neural Network and Condensed Nearest Neighbors with Selection of NSL-KDD Influencing Features.* pp. 23.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay (2011) *Scikit-learn: Machine Learning in Python.* Available at: https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html (Accessed: 27 April 2022).

Gartner, I. (2021) *Gartner Says Threat of New Ransomware Models is the Top Emerging Risk Facing Organisations.* Available at: https://www.gartner.com/en/newsroom/press-releases/2021-10-21-gartner-says-threat-of-new-ransomware-models-is-the-top-emerging-risk-facing-organizations (Accessed: 14 December 2021).

Granjal, J., Monteiro, E. and Silva, J.S. (2015) 'Security for the internet of things: a survey of existing protocols and open research issues', *IEEE Communications Surveys & Tutorials,* 17(3), pp. 1294-1312.

Harris, C.R., Millman, K.J., van der Walt, Stéfan J, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., Del Río, J.F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C. and Oliphant, T.E. (2020) 'Array programming with NumPy', *Nature (London),* 585(7825), pp. 357-362. doi: 10.1038/s41586-020-2649-2.

IBM *What is a cyber attack?* Available at: https://www.ibm.com/topics/cyber-attack (Accessed: 12 December, 2021).

IBM Cloud Education (2020a) *Supervised Learning.* Available at: https://www.ibm.com/cloud/learn/supervised-learning (Accessed: 14 April 2022).

IBM Cloud Education (2020b) *Unsupervised Learning.* Available at: https://www.ibm.com/cloud/learn/unsupervised-learning (Accessed: 14 April 2022).

INSIKT GROUP® *H1 2021: Malware and Vulnerability Trends Report.* Available at: https://www.recordedfuture.com/malware-vulnerability-trends-report/ (Accessed: 14 December 2021).

J. D. Hunter (2007) *Matplotlib: A 2D Graphics Environment*.

Jena, P.K., Ghosh, S. and Koley, E. (2022) 'Identification of strategic sensor locations for intrusion detection and classification in smart grid networks', *International Journal of Electrical Power & Energy Systems,* 139, pp. 107970.

K. K. Jha, R. Jha, A. K. Jha, M. A. M. Hassan, S. K. Yadav and T. Mahesh (2021) *A Brief Comparison On Machine Learning Algorithms Based On Various Applications: A Comprehensive Survey.* pp. 1.

Khalaf, B.A., Mostafa, S.A., Mustapha, A., Mohammed, M.A. and Abduallah, W.M. (2019) 'Comprehensive review of artificial intelligence and statistical approaches in distributed denial of service attack and defense methods', *IEEE Access,* 7, pp. 51691-51713.

M. Usama, J. Qadir, A. Raza, H. Arif, K. A. Yau, Y. Elkhatib, A. Hussain and A. Al-Fuqaha (2019) *Unsupervised Machine Learning for Networking: Techniques, Applications and Research Challenges*.

Mahesh, B. (2020) 'Machine learning algorithms-a review', *International Journal of Science and Research (IJSR).[Internet],* 9, pp. 381-386.

MarketsandMarkets Research (2020) *Intrusion Detection and Prevention Systems Market by Component (Solutions and Services), Type, Deployment Type (Cloud and On-Premises), Organization Size (SMEs and Large Enterprises), Vertical, and Region - Global Forecast to 2025.* Available at: https://www.marketsandmarkets.com/Market-Reports/intrusion-detection-prevention-system-market-199381457.html (Accessed: 1 May, 2022).

Matplotlib (2021) *Matplotlib: Visualisation with Python.* Available at: https://matplotlib.org/ (Accessed: 27 April 2022).

Morgan, D. and Jacobs, R. (2020) 'Opportunities and Challenges for Machine Learning in Materials Science', *Annual Review of Materials Research,* 50(1), pp. 71-103. doi: 10.1146/annurev-matsci-070218-010015.

Norton *Norton (2021) Emerging Threats.* . Available at: https://us.norton.com/internetsecurity-emerging-threats-how-do-zero-day-vulnerabilities-work.html (Accessed: : 15 December 2021).

Otmane Azeroual and Anastasija Nikiforova (2022) 'Apache Spark and MLlib-Based Intrusion Detection System or How the Big Data Technologies Can Secure the Data', *Information (Basel),* 13(2), pp. 58. doi: 10.3390/info13020058.

Ottis, R. and Lorents, P. (2010) *Cyberspace: Definition and implications.* Academic Conferences International Limited, pp. 267.

Pandas (2022) *Pandas Documentation.* Available at: https://pandas.pydata.org/docs/ (Accessed: 27 April 2022).

Panigrahi, R., Borah, S., Bhoi, A.K., Ijaz, M.F., Pramanik, M., Jhaveri, R.H. and Chowdhary, C.L. (2021) 'Performance assessment of supervised classifiers for designing intrusion detection systems: a comprehensive review and recommendations for future research', *Mathematics,* 9(6), pp. 690.

Rieck, K., Trinius, P., Willems, C. and Holz, T. (2011) 'Automatic analysis of malware behavior using machine learning', *Journal of Computer Security,* 19(4), pp. 639-668.

S. Ray (2019) *A Quick Review of Machine Learning Algorithms.* pp. 35.

Saranya, T., Sridevi, S., Deisy, C., Chung, T.D. and Khan, M. K. A. Ahamed (2020a) 'Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review', *Procedia Computer Science,* 171, pp. 1251-1260. doi: https://doi.org/10.1016/j.procs.2020.04.133.

Saranya, T., Sridevi, S., Deisy, C., Chung, T.D. and Khan, M.A. (2020b) 'Performance analysis of machine learning algorithms in intrusion detection system: A review', *Procedia Computer Science,* 171, pp. 1251-1260.

Teles, G., Rodrigues, J.J., Rabêlo, R.A. and Kozlov, S.A. (2021) 'Comparative study of support vector machines and random forests machine learning algorithms on credit operation', *Software: Practice and Experience,* 51(12), pp. 2492-2500.

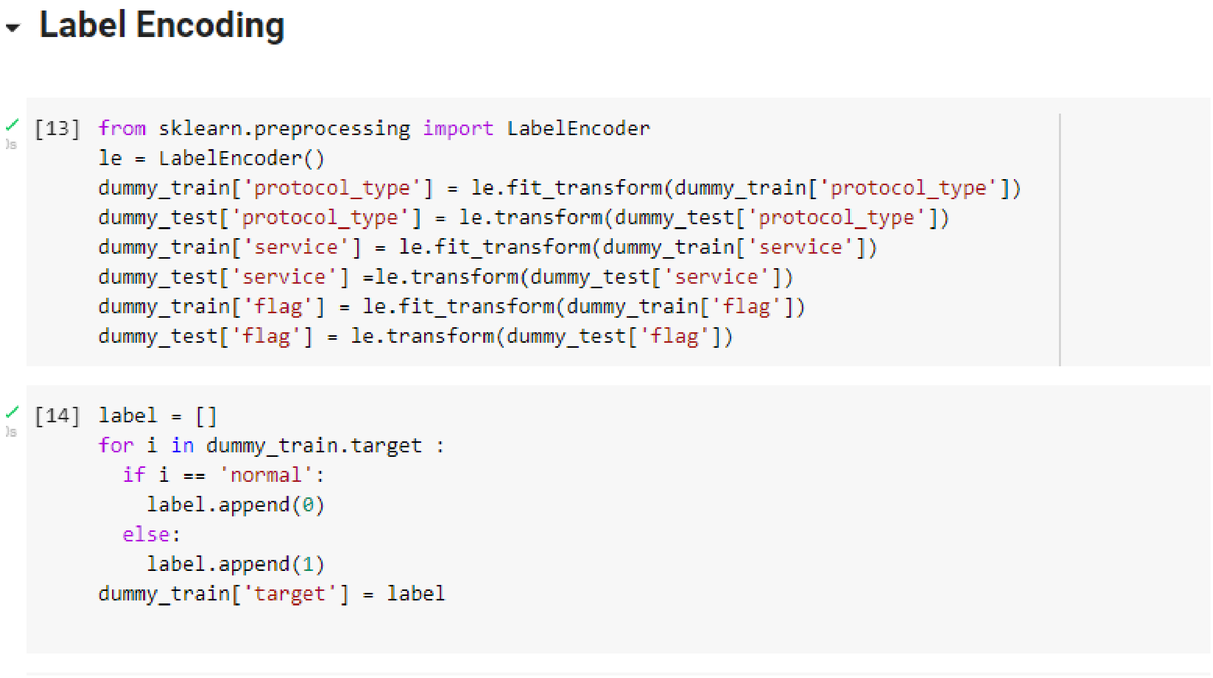
Wang, L., Jajodia, S., Singhal, A. and Noel, S. (2010) *k-zero day safety: Measuring the security risk of networks against unknown attacks.* Springer, pp. 573.

Wood, P. (2012) *Symantec Intelligence Report: October 2012.* Symantec Intelligence. Available at: https://docs.broadcom.com/doc/intelligence-report-oct-12-en (Accessed: .

Y. Rbah, M. Mahfoudi, Y. Balboul, M. Fattah, S. Mazer, M. Elbekkali and B. Bernoussi (2022) *Machine Learning and Deep Learning Methods for Intrusion Detection Systems in IoMT: A survey.* pp. 1.

Appendicies

Codes

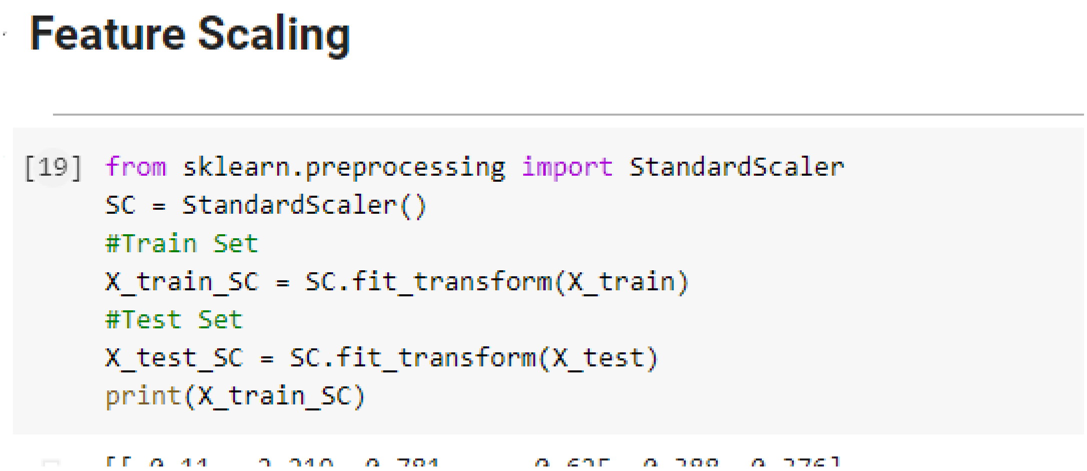


Label Encoding

```
[13] from sklearn.preprocessing import LabelEncoder
     le = LabelEncoder()
     dummy_train['protocol_type'] = le.fit_transform(dummy_train['protocol_type'])
     dummy_test['protocol_type'] = le.transform(dummy_test['protocol_type'])
     dummy_train['service'] = le.fit_transform(dummy_train['service'])
     dummy_test['service'] =le.transform(dummy_test['service'])
     dummy_train['flag'] = le.fit_transform(dummy_train['flag'])
     dummy_test['flag'] = le.transform(dummy_test['flag'])
```

```
[14] label = []
     for i in dummy_train.target :
       if i == 'normal':
         label.append(0)
       else:
         label.append(1)
     dummy_train['target'] = label
```

FIGURE 17: LABEL ENCODING CODE.

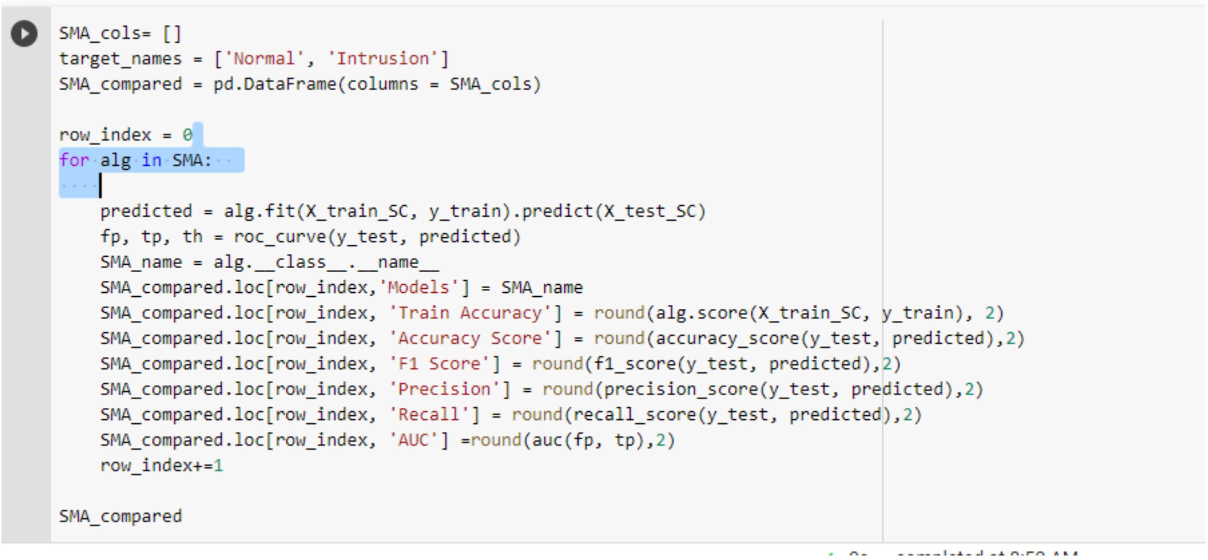


Feature Scaling

```
[19] from sklearn.preprocessing import StandardScaler
     SC = StandardScaler()
     #Train Set
     X_train_SC = SC.fit_transform(X_train)
     #Test Set
     X_test_SC = SC.fit_transform(X_test)
     print(X_train_SC)
```

FIGURE 18: FEATURE SCALING CODE.

Modelling



```
SMA_cols= []
target_names = ['Normal', 'Intrusion']
SMA_compared = pd.DataFrame(columns = SMA_cols)

row_index = 0
for alg in SMA:

    predicted = alg.fit(X_train_SC, y_train).predict(X_test_SC)
    fp, tp, th = roc_curve(y_test, predicted)
    SMA_name = alg.__class__.__name__
    SMA_compared.loc[row_index,'Models'] = SMA_name
    SMA_compared.loc[row_index, 'Train Accuracy'] = round(alg.score(X_train_SC, y_train), 2)
    SMA_compared.loc[row_index, 'Accuracy Score'] = round(accuracy_score(y_test, predicted),2)
    SMA_compared.loc[row_index, 'F1 Score'] = round(f1_score(y_test, predicted),2)
    SMA_compared.loc[row_index, 'Precision'] = round(precision_score(y_test, predicted),2)
    SMA_compared.loc[row_index, 'Recall'] = round(recall_score(y_test, predicted),2)
    SMA_compared.loc[row_index, 'AUC'] =round(auc(fp, tp),2)
    row_index+=1

SMA_compared
```

FIGURE 19: MODELLING CODE.