Title: ‘Hybrid intrusion detection system using machine learning’

An intrusion detection system (IDS) watches network traffic for suspicious activity and sends out alerts when it is found. It is a software application that checks a system or network for malicious activities or policy violations.

20200585	20200217	20200856	20200807
Karam Issa	Maria Ishaq	Jude Hamdan	Jood Maaroufi

Table Of Contents:

1.1 Introduction:

Cyber attacks are one of the top five hazards for the year 2020 in the world we live in today. This issue affects both public and private spheres and is not particular to one industry. IoT cyberattacks are expected to treble by 2025, which is unfortunate because the problem is just getting worse..[1]

Cybersecurity Ventures estimates that by 2025, cybercrime would cost organisations throughout the world $10.5 trillion annually, a huge increase from the $3 trillion it cost them in 2015. Its cost is expected to increase at a pace of 15% every year. It has also been noted as the largest recorded transfer of economic wealth in history.[2]

Furthermore, the rate of cybercriminals being found and prosecuted in the United States is as low as 0.05 percent, according to the World Economic Forum's 2020 Global Risk Report. [3]. So, it is obvious why network security has grown to be a top concern for both individuals and organisations. It is now crucial to have strong security measures in place to guard against breaches and identify them given the frequency and sophistication of cyber attacks. A network-based intrusion detection system is one of these measures (NIDS).

A security technology called a Network Intrusion Detection System (NIDS) is made to find different kinds of network-based threats or attacks. NIDS functions by monitoring network traffic in real-time for any unusual activity that might be an attack sign. The system is able to identify a variety of security risks, such as malware infections, DoS assaults, port scanning, SQL injections, and other forms of attacks that could jeopardise network security.[4]

The system scans each packet for harmful activity by installing sensors at key network checkpoints. When aberrant traffic is discovered, the NIDS notifies the administrator, who can then look into it further. In order to give sensors the best visibility, it is critical to think about where to put them. Depending on the volume of traffic to and from network devices, numerous NIDS may be necessary.

The benefits of Network-Based Intrusion Detection Systems (NIDS) go beyond the features stated above. First off, they can be swiftly and painlessly installed into an existing network with minimum disturbance, enabling businesses to strengthen their security posture. Second, NIDS can record evidence of an attack that the malicious actor may have attempted to erase because they are capable of detecting real-time occurrences.[5]

A NIDS can also analyse various attack types and quantities, offering insightful information that may be utilised to implement more effective security measures and spot problems with network device settings. It is also simpler to comply with certain IT security compliance standards due to the improved network visibility.[5]

In summary, an NIDS is a potent tool that can assist organisations in anticipating potential cyber threats and taking appropriate action, enabling them to better defend their sensitive data and systems from hostile intrusions.

1.2 System Description:

A network intrusion detection system's (NIDS) primary job is to spot and warn of any unauthorised or suspicious activity on a network. The NIDS constantly scans network traffic for patterns or anomalies that show a possible security breach, like attempts at unauthorised access, malware infestations, or odd network behaviour. The NIDS alerts or sends a message to security staff as soon as it notices something unusual so that they can further analyse the occurrence and take the necessary precautions to reduce any potential security threats.

.

There are two main methods that the NIDS use for detecting suspicious activities:

1. **Signature-based detection** (rule-based detection) utilises a pre-existing database or pre-programmed list of signatures that identify particular patterns, behaviours, or traits that are frequently associated with recognized threats or attacks. These descriptions are called indications of compromise (IOC), and they could include particular behaviours that frequently precede hostile network attacks, file hashes, malicious sites, well-known byte sequences, or even the subject lines of emails. The IDS sends an alert to inform security staff of the potential security concern whenever network behaviour matches one of these signatures.

2. **Anomaly-based detection**,uses machine learning, to identify a normalised baseline of network behaviour and contrasts all activity with that baseline. Every strange behaviour is seen as a threat. Every network activity is compared to this baseline behaviour as the standard for typical system operations. Anomaly-based IDS recognizes any odd activity that could be a sign of a threat rather than looking for well-known indicators of compromise (IOCs). Also, any behaviour that deviates from the norm, such as a person signing in outside of regular business hours, an unapproved device being added to a network, or a large number of new IP addresses trying to connect, may raise a red signal.

The benefit of signature-based detection is that it is more accurate for well-known assaults and processes data quickly. Nevertheless, it only identifies known threats and is unable to find zero-day vulnerabilities. Anomaly-based detection, on the other hand, has the ability to identify undiscovered threats and zero-day vulnerabilities. Anomaly-based detection has the drawback of increasing the possibility of false positives because many benign behaviours may be labelled suspicious. Due to this, it can take more time and will cost more to look into every warning. [6]

In order to complement one another, the proposed NIDS system will be a hybrid that includes both signature-based and anomaly-based detection methods. While anomaly-based detection is good at finding unknown threats and zero-day exploits, signature-based detection excels at recognizing known threats.

Semi-Supervised machine learning will also be used in our suggested NIDS. A combination between supervised and unsupervised learning is called semi-supervised machine learning. To combine the advantages of both methods and get around the difficulty of obtaining a huge quantity of labelled data, it includes employing a small batch of labelled data and a large number of unlabeled data. As a result, the model can acquire labelling skills with less labelled training data than is necessary for standard supervised learning..[7]

In this method, the system employs unsupervised learning to detect unknown or anomalous behaviour, such as zero-day exploits, and establishes a baseline for "relative" normal network activity for various networks. Supervised learning is used to train the machine learning algorithm on known threats and behaviours.

Using several data sets, including network traffic statistics, logs, and other pertinent data sources, our machine learning algorithm will be trained. According to the literature, the system will be fed benchmark data sets such the KDD'99, NSL-KDD, UNSW-Nb15, and CICIDS2017. The application will produce a binary value that indicates whether or not an intrusion has happened. This value can be used to start alerts or perform other necessary actions.[8]

1.3 System Purpose:

In the modern world, technology use is only going to increase, which increases the risk of security breaches and cyberattacks. A trustworthy and efficient intrusion detection system must be in place to safeguard networks and systems from these dangers.

A hybrid intrusion detection system combines different detection techniques to offer a more thorough means of locating and reducing security threats. To better identify and stop cyberattacks, this kind of system may analyse network data, spot behavioural irregularities, and even pinpoint well-known attack patterns.

A hybrid intrusion detection system can also assist businesses in adhering to rules and regulations. To protect sensitive data, many businesses, including healthcare and finance, must adhere to tight security regulations. By offering a strong security framework that can identify and address possible threats, a hybrid IDS can assist organisations in meeting these standards.

A hybrid intrusion detection system's main goal is to offer a multi-layered defence against online attacks. The solution can help organisations comply with legal standards, promptly identify and address any security breaches, and ultimately safeguard crucial systems and data by combining several detection techniques.

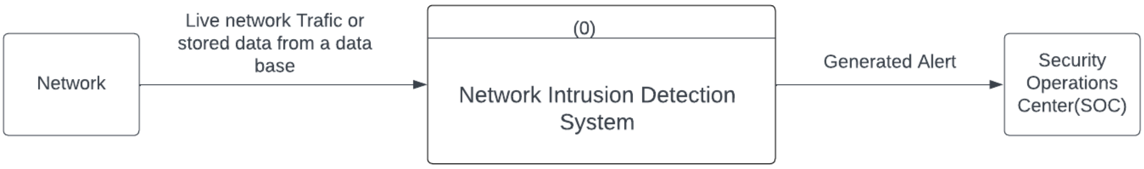
1.4 Problem Statement:

The issue stems from the requirement for an accurate and powerful intrusion detection system that can recognize and react to both known and unidentified threats. Although intrusion detection systems based on signatures are effective at detecting known threats. They are only able to detect attacks for which they have a preset signature, which limits their ability. They are unable to recognize unidentified attacks that do not correspond to any recognized signatures. Anomaly-based intrusion detection systems, on the other hand, can identify previously unidentified attacks by examining deviations from expected behaviour in network data. High false-positive rates present an additional issue for anomaly-based IDS, since they might cause alarms to be generated for legitimate system activity or changes to the network environment.

The objective is to design a hybrid intrusion detection system that combines the strengths of both signature-based and anomaly-based detection methods, enabling detection of both known and new threats while lowering false positive and false negative rates. This necessitates carefully integrating and optimising the numerous detection algorithms in order to ensure high accuracy and dependability.

Designing a robust hybrid intrusion detection system is essential given the growing sophistication of cyberattacks. As hybrid computing environments grow more prevalent and the amount and complexity of cyber threats continue to increase, it is essential to have an effective intrusion detection system that can quickly and efficiently identify potential assaults and respond to them before any damage can occur..

1.5 The system Context View:


Network
Live network Trafic or stored data from a data base
(0)
Network Intrusion Detection System
Generated Alert
Security Operations Center(SOC)

1.6 Literature review

Network-based Intrusion Detection Systems (NIDS) play a crucial role in safeguarding against network-based attacks. These systems scrutinise network traffic to detect any anomalies that might signal a cyberattack or intrusion. To enhance the precision of NIDS, machine learning algorithms have gained popularity as they can recognize patterns in network traffic that may go unnoticed by conventional rule-based methods.

Over the past few years, extensive research has been conducted on machine learning algorithms in NIDS, with scholars investigating various methods to enhance detection accuracy and lower false positives. This literature review presents a summary of the most recent research on NIDS employing machine learning, encompassing different machine learning algorithms implemented in NIDS, obstacles related to these systems, and various approaches suggested to boost detection accuracy.

The review emphasises the significance of carefully selecting suitable machine learning algorithms, the necessity of feature selection and engineering, and the importance of obtaining labelled data to train NIDS effectively. Additionally, the review examines the potential of deep learning methods, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), in enhancing detection accuracy in NIDS.

All in all, NIDS utilising machine learning algorithms have demonstrated their efficacy in detecting and curtailing network-based threats in contemporary cybersecurity systems. As the threat landscape evolves continually, scholars are exploring novel approaches to enhance the accuracy and efficacy of these systems, rendering them a critical tool for preserving the security of modern networks.

A 2018 research paper by Mrutyunjaya Panda, Ajith Abraham, Swagatam Das, and Manas Ranjan Patra, titled [8], presents a machine learning-based approach for network intrusion detection systems (NIDS) using diverse fundamental machine learning techniques.

The authors begin by presenting a survey of various types of network attacks and the significance of NIDS in detecting and preventing such attacks. They subsequently introduce the concept of machine learning and detail how it can enhance the accuracy and efficiency of NIDS.

The article [8] concentrates on several fundamental machine learning algorithms, including Naive Bayes, Decision Trees, k-Nearest Neighbors, Random Forest, and Support Vector Machines. The authors evaluate the performance of these algorithms using a publicly available dataset and compare their accuracy and efficiency. Additionally, they provide valuable insights into the advantages and drawbacks of each approach.

The study's findings indicate that the Random Forest algorithm outperformed the other algorithms in terms of detection rate while maintaining a low false alarm rate. The authors also emphasise the significance of feature selection and feature engineering in enhancing the precision of machine learning-based NIDS.

In 2019, Emad E. Abdallah, Wafa' Eleisah, and Ahmed Fawzi Otoom published a comprehensive survey article, [9], which presents an in-depth review of several supervised machine learning methods that have been employed for intrusion detection in network security.

The authors of [9] begin by introducing the concept of intrusion detection systems (IDS) and explaining the significance of incorporating machine learning techniques in IDS to enable effective detection and prevention of network intrusions. The survey encompasses a broad range of supervised machine learning algorithms employed in intrusion detection, including decision trees, support vector machines, artificial neural networks, and deep learning methods.

The survey [9] also sheds light on the challenges confronting researchers and practitioners in the field of intrusion detection, such as the requirement for extensive labelled datasets, class imbalance, and the difficulty of selecting relevant features. The authors propose potential solutions to these challenges and identify prospects for future research in this field.

[9] provides an excellent resource for those involved in network security, presenting a detailed survey of supervised machine learning techniques and their applications in intrusion detection. The article highlights the advantages and limitations of each technique and identifies key challenges that researchers and practitioners face in this field.

[10], on the other hand, presents a new approach for network intrusion detection systems, which combines two clustering algorithms to improve detection accuracy. This novel method shows potential for improving the effectiveness of NIDS and contributing to the ongoing development of network security systems.

The proposed hybrid unsupervised clustering-based anomaly detection method presented in [10] is an innovative approach that aims to improve the accuracy of network intrusion detection systems (NIDS). The authors of the article, Guo Pu, Lijuan Wang, Jun Shen, and Fang Dong, combine a density-based clustering algorithm and a distance-based clustering algorithm to achieve this goal. The authors conducted experiments using the UNSW-NB15 dataset and compared the performance of their method with several state-of-the-art anomaly detection methods. The results showed that the proposed method outperformed other methods in terms of both detection rate and false positive rate. Furthermore, the authors evaluated the proposed method on a real-world dataset and achieved similar results, demonstrating its effectiveness in practical applications.

This article proposes a novel technique that combines the benefits of two clustering methods to increase the precision and effectiveness of anomaly identification, making a significant addition to the field of network intrusion detection. The suggested approach has the potential to be used with more datasets and can be improved for usage in actual NIDS.

Zein Ashi, Laila Aburashed, Mahmoud Al-Qudah, and Abdallah Qusef published an paper in 2019 titled [11]. It examines the application of dimensionality reduction and supervised machine learning methods for network intrusion detection systems (NIDS). Key research gaps and upcoming prospects for the subject are identified by the authors, who also compare and assess various algorithms and methodologies. The paper is a helpful tool for academics and professionals working in the area of network security.

In his 2018 article [12], Maxime Labonne examines how anomaly-based detection can be used to leverage machine learning for network intrusion detection. The essay examines the drawbacks of conventional signature-based intrusion detection systems and suggests remedies based on machine learning methods such as SVMs, decision trees, and neural networks. It also discusses the difficulties in developing machine learning models for intrusion detection and makes recommendations for solutions.

[13] An improved machine learning strategy based on the decision tree method is proposed in the paper "An Enhanced Machine Learning Strategy for Network Intrusion Detection System" for network intrusion detection. The suggested system beat previous machine learning methods in terms of detection rate and false alarm rate, achieving an accuracy rate of 99.84%.

[14] This study suggests a machine learning-based hybrid feature selection methodology for network intrusion detection systems. The authors select the most pertinent attributes from a big dataset in order to increase the precision and effectiveness of network intrusion detection. To assess the chosen features and assess the performance of the various machine learning techniques, including decision trees, random forests, and support vector machines. The tests are carried out on the KDD Cup 99 dataset, and the outcomes demonstrate that the suggested framework can achieve excellent accuracy with a small feature set. The overall goal of this research is to offer a machine learning-based method for feature selection in network intrusion detection systems.

[15] The evaluation of several machine learning methods used in network intrusion detection systems is the main goal of the study carried out by Patrick Vanin and Thomas Newe. The study examines the effectiveness of these algorithms using a number of criteria, including accuracy, false alarm rate, and detection rate. The findings demonstrate that machine learning methods perform better than conventional signature-based intrusion detection systems and have high rates of accuracy in identifying network intrusions. The study emphasises how machine learning algorithms may be used to improve the effectiveness and efficiency of network security systems.

[16] This article does a thorough analysis of intrusion detection systems based on deep learning (IDS). The authors evaluate the efficacy of various deep learning techniques for intrusion detection and provide a summary of their benefits and drawbacks. They also shed light on potential future lines of investigation in this area.

1.7 Challenges:

1. As machine learning techniques used in IDS need a lot of data for training and testing, obtaining high-quality labelled data might be difficult. Finding high-quality data may be challenging due to variables such the scarcity of labelled data, data noise, and data imbalance.

2. Overfitting, which can happen when a model is trained on a dataset that is not representative of real-world data or when a model is overly sophisticated and fits noise in the training data rather than underlying patterns, can be a serious problem when creating an IDS using ML. This might lead to the model being overly specialised to the training data and making it incapable of detecting novel attack types or anomalies in the real-world data, which eventually reduces the IDS's effectiveness.

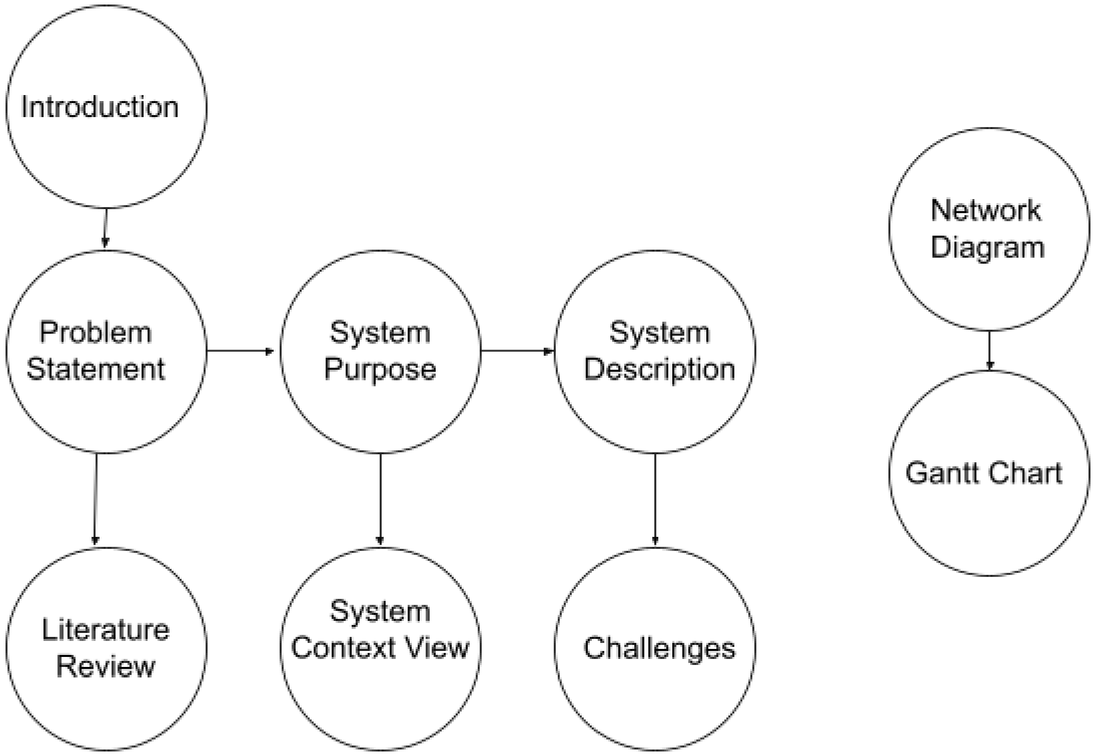
3.Training time might be a problem when using ML to create an IDS because the algorithms need a lot of data to be trained on, which can take a long time. Therefore, the length of training will increase with the complexity of the ML model. Because of this, training might take a while, especially when working with huge datasets.

4.When creating a hybrid intrusion detection system that makes use of machine learning techniques, selecting the appropriate models is a vital component. The models themselves, as well as the data, must be carefully examined in this process. Also, it's crucial to mix the several models in a way that maximises their individual benefits while minimising any disadvantages.

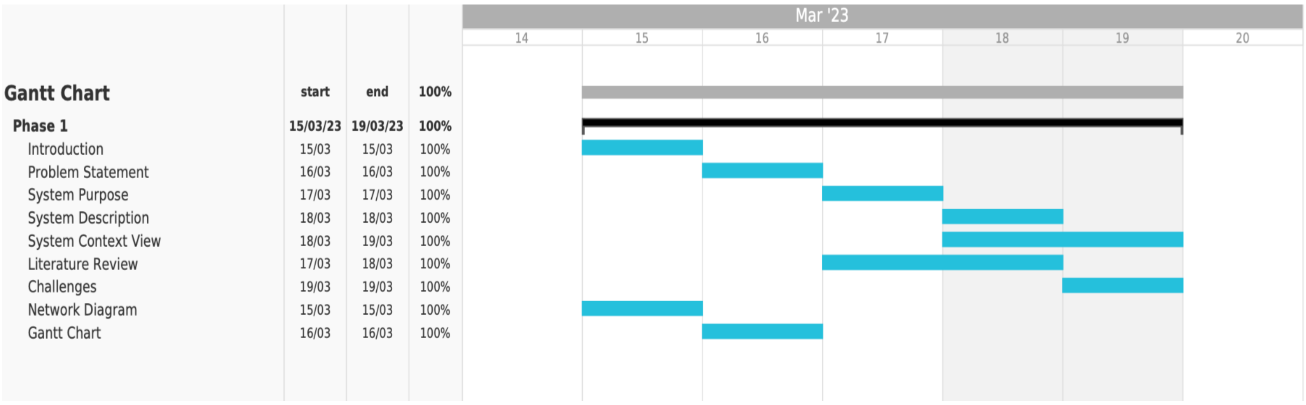
1.8 Projection:

Activity	Predecessors	Duration (days)
Introduction	—	1
Problem Statement	Introduction	1
System Purpose	Problem Statement	1
System Description	System Purpose	1
System Context View	System Purpose	2
Literature Review	Problem Statement	2
Challenges	System Description	1
Network Diagram	—	1
Gantt Chart	Network Diagram	1

Network diagram:



Gantt chart:



Gantt Chart	start	end	100%
Phase 1	**15/03/23**	**19/03/23**	**100%**
Introduction	15/03	15/03	100%
Problem Statement	16/03	16/03	100%
System Purpose	17/03	17/03	100%
System Description	18/03	18/03	100%
System Context View	18/03	19/03	100%
Literature Review	17/03	18/03	100%
Challenges	19/03	19/03	100%
Network Diagram	15/03	15/03	100%
Gantt Chart	16/03	16/03	100%

2.1 Functional requirements of the system

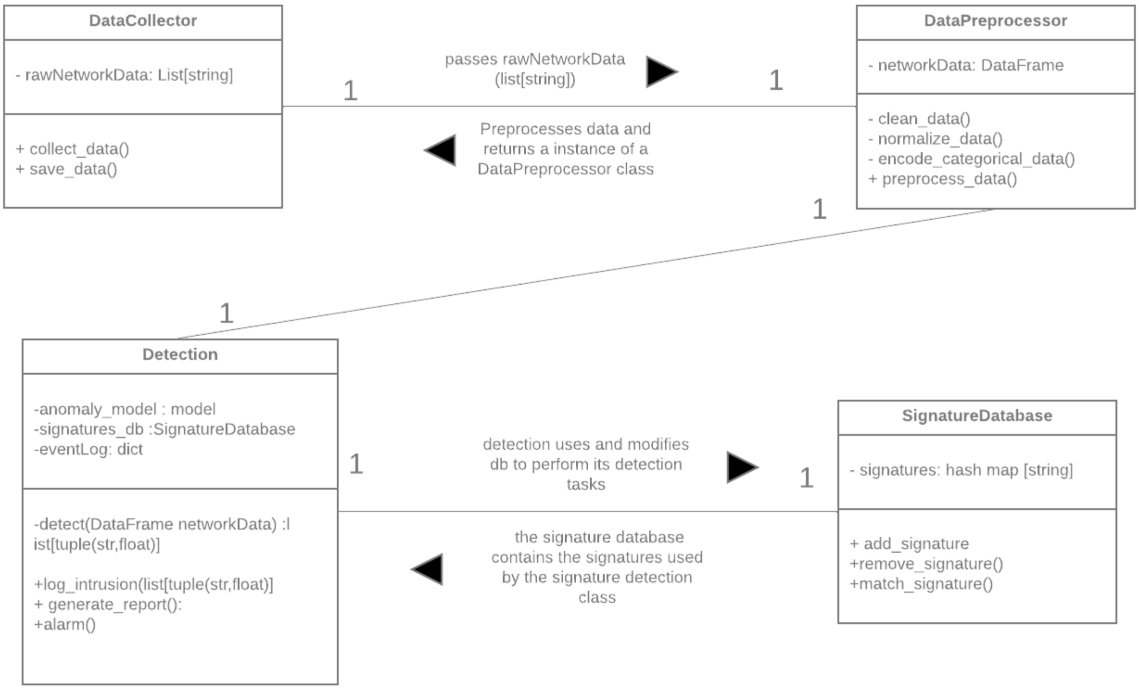
Requirement ID	Requirement Name	Requirement Details	Priority
FR-1	Collecting Network Traffic	Gather data on any network activity that appears to be suspicious, with information that defines the activity.	High
FR-2	Processing	Performing data reduction and cleansing to reduce redundancy and data load	High
FR-3	Signature-based detection	using pre-existing databases to detect patterns or behaviours of attacks.	High
FR-4	Anomaly-based detection	Using machine learning to create a baseline and compare network activities with it to recognize odd activity without relying on known indicators.	High
FR-5	Real-time monitoring	Continuously monitoring network traffic in real-time to detect any suspicious activity.	High
FR-6	Alerting mechanism	Having an alerting mechanism that notifies security personnel or administrators when suspicious activity is detected.	High
sFR-7	Event logging	Maintain a log of all events, including alerts, and system activities.	High
FR-8	Reporting	Provide reporting capabilities, including the ability to generate reports on detected events, system activity, and performance.	High
FR-9	Configuration	must allow the system administrator or anomaly detection module to configure the detection rules..The detection rules must be based on the organisation's security policies and the latest security threats and	High

		vulnerabilities, and must be stored in a centralised database that is updated regularly to ensure the accuracy and effectiveness of the system.	

2.2 Non-Functional requirements

Requirement ID	**Requirement Name**	**Requirement Details**	**Priority**
NF-1	Performance	The IDS must have the ability to quickly and effectively process both system events and network traffic in order to provide quick detection and reaction to potential security threats.	High
NF-3	Scalability	It is required for the IDS to be able to handle increasing workloads and data volumes to achieve effective security breach detection and prevention	Medium
NF-4	Maintainability	IDS must be maintainable to be able to handle the emerging new threats and keep up with the technological improvements.	Medium
NF-5	Usability	Usable IDS makes It simpler and more effective for security analysts to utilise and administer the system.	Medium
NF-6	Legal and ethical	The system has to abide by all applicable laws, rules, and moral principles.	High

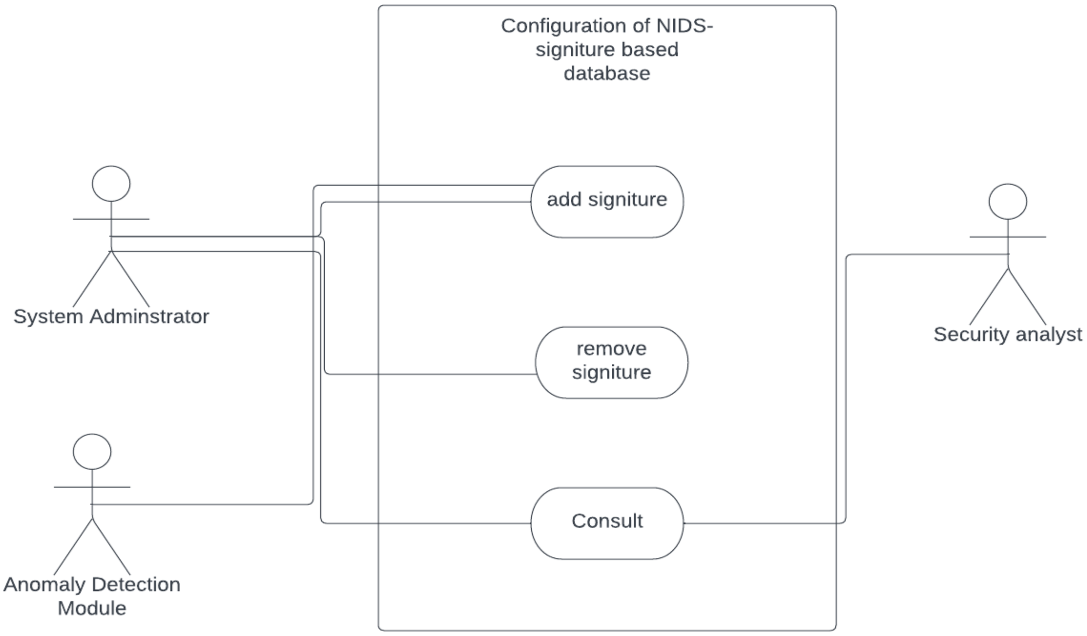
2.3 Class Diagram:


DataCollector
- rawNetworkData: List[string]
+ collect_data()
+ save_data()
1
passes rawNetworkData (list[string])
1
Preprocesses data and returns a instance of a DataPreprocessor class
DataPreprocessor
- networkData: DataFrame
- clean_data()
- normalize_data()
- encode_categorical_data()
+ preprocess_data()
1
1
Detection
-anomaly_model : model
-signatures_db :SignatureDatabase
-eventLog: dict
-detect(DataFrame networkData) :list[tuple(str,float)]
+log_intrusion(list[tuple(str,float)]
+ generate_report():
+alarm()
1
detection uses and modifies db to perform its detection tasks
1
the signature database contains the signatures used by the signature detection class
SignatureDatabase
- signatures: hash map [string]
+ add_signature
+remove_signature()
+match_signature()

2.4 Use Case Diagram:

Use case(1): Configure detection rules

Use Case (1) Diagram:



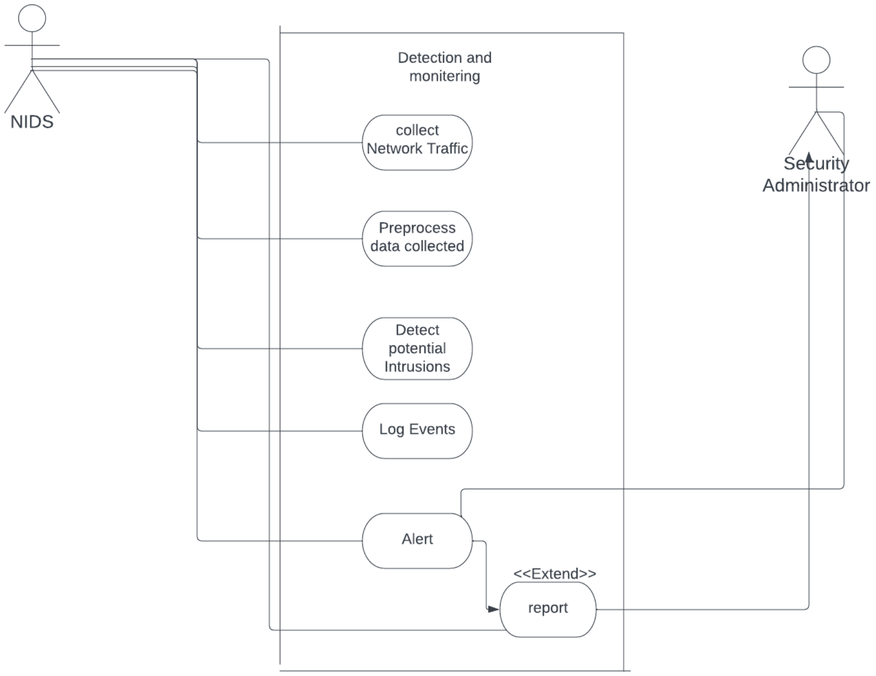
Actors: System Administrator, Security Analyst, Anomaly Detection Module
purpose:Allows configures the intrusion detection system's detection criteria, including the signatures, protocols, and ports to monitor, for the system administrator or anomaly detection module.
Overview(Success Scenario): having the signature databases successfully updated after adding or removing a signature.
Type: Primary
Cross Reference: FR9
Typical course of action: After getting consultation from the security analyst the system administrator can add the latest known security threats and vulnerabilities or remove any signatures that may have resulted in false positives from the anomaly detection updates.Also allowing the anomaly based detection system to automatically add signatures to the

database that have a highly likelihood of being a intrusion, so next time a similar signature is collected it will be quickly detected by rule-based detection.

Actor Action	System Response
1- this case begins when either the security administrator wants to manually configure the detection rule set or the anomaly detection module configures the rules set by adding a new rule	
2-the security analyst gives consultation to the security administrator	3- Makes sure the new rule doesn't already exist in the rule set
	4- generates the correct signature for the intrusion
	5- adds the rule to the signature based model
	6-updates the signature based rule set so all instances of detection.

Use case(2): Monitor Network Traffic and Detect security threats

Use Case(2) Diagram:



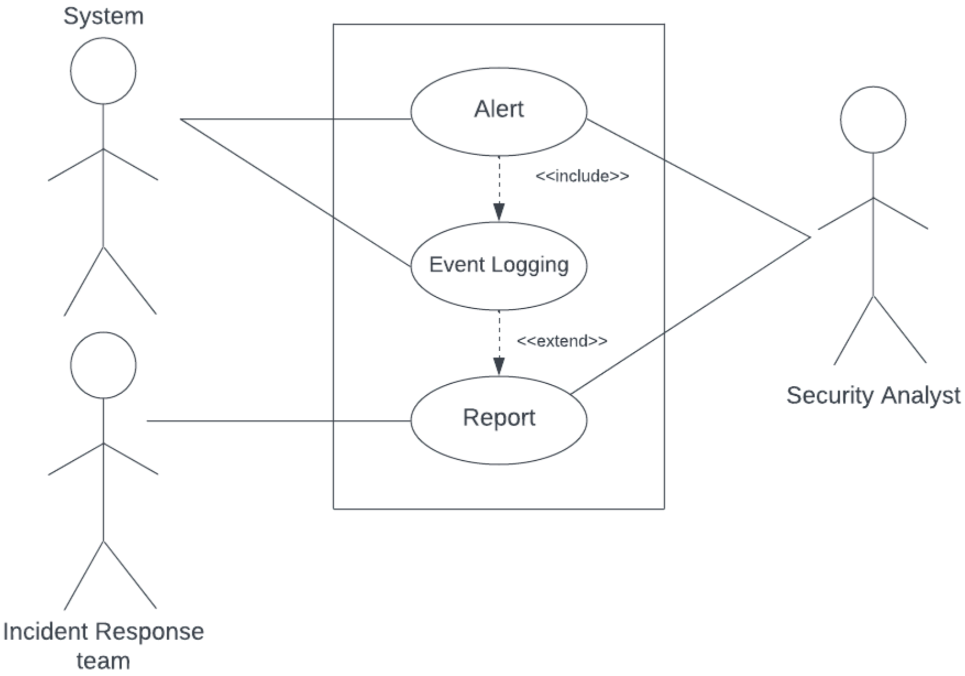
Actors: NIDS , Security Administrator.
Purpose: core functionality to detect various types of security threats
Overview(Success scenario): the system will start listening to network traffic, preprocess the data collected, passes this data to the detection module that is made from both anomaly and signature based detection modules, if an intrusion is found or suspected it will log this network event and alarm the security administrator with an alert, with additional optional feature of generating a detailed report about a specific incident for further investigation.
Type: Primary.
Cross references: FR-1 , FR-2, FR-3, FR-4, FR-6,FR-7.
Typical course of action: The admin starts the IDS in order for the system to continuously collect network traffic and detect threats and then alert the security analyst, by generating a report.

Actor Action	System Response
1- This use case begins when the security administrator starts the NIDS system.	2- The NIDS starts collecting data, then processing it.

	3- The collected data is then passed to a detection module, to detect threats.
	4- If a threat is detected, the system logs the event and generates an alert to the security administrator.
5-Security analyst further investigates the threat by requesting a report	6- The system generates a report to the security Administrator.

<u>**Use case(3):**</u> Investigate Security Incidents and receive alerts

Use case(3) Diagram:



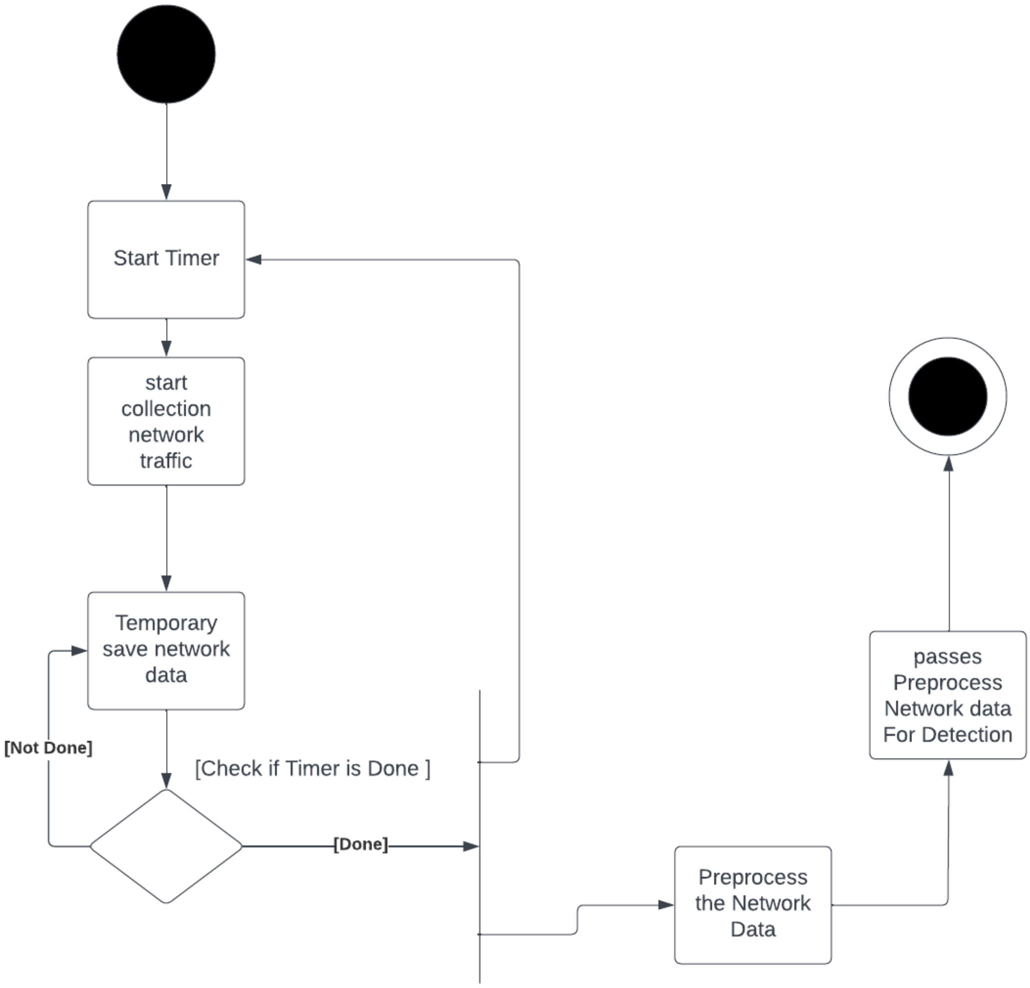
Actor: Incident Response Team,system, security analyst
Purpose: Respond to security incidents by reviewing alerts generated by the NIDS and investigating the nature and scope of the incident to prevent further damage to the organisation.
Overview(Success Scenario): Aiding in the investigation of the incident that will help in planning the correct response to the intrusion, so the organisation can resume its normal activities
Type: Primary
Cross Reference: FR5, FR6, FR8

Typical course of actions: While the NIDS is continuously monitoring network traffic, a security alert will be generated to determine the nature and scope of the incident, after that the security analyst may request a detailed report from the incident response team to gather additional information and choose the correct course of action in order to prevent further damage.

Actor Action	System Response
1- Incident will be detected from the anomaly or signature based detection modules front NIDS system	2- system will generate an alert to the security analyst to flag any abnormal activity
	3- system will also log this intrusion for backup and lookup purposes
4-Security analyst will review the alert and based on the scope and impact of the incident will request a report from the system	5- the system will use the logged events to generate a detailed report for further investigation that will contain additional information .
6-Security analyst and an incident response team will devise a plan to isolate or block the malicious traffic so the organisation is able to return to normal operations.	

2.5 Activity Diagram:

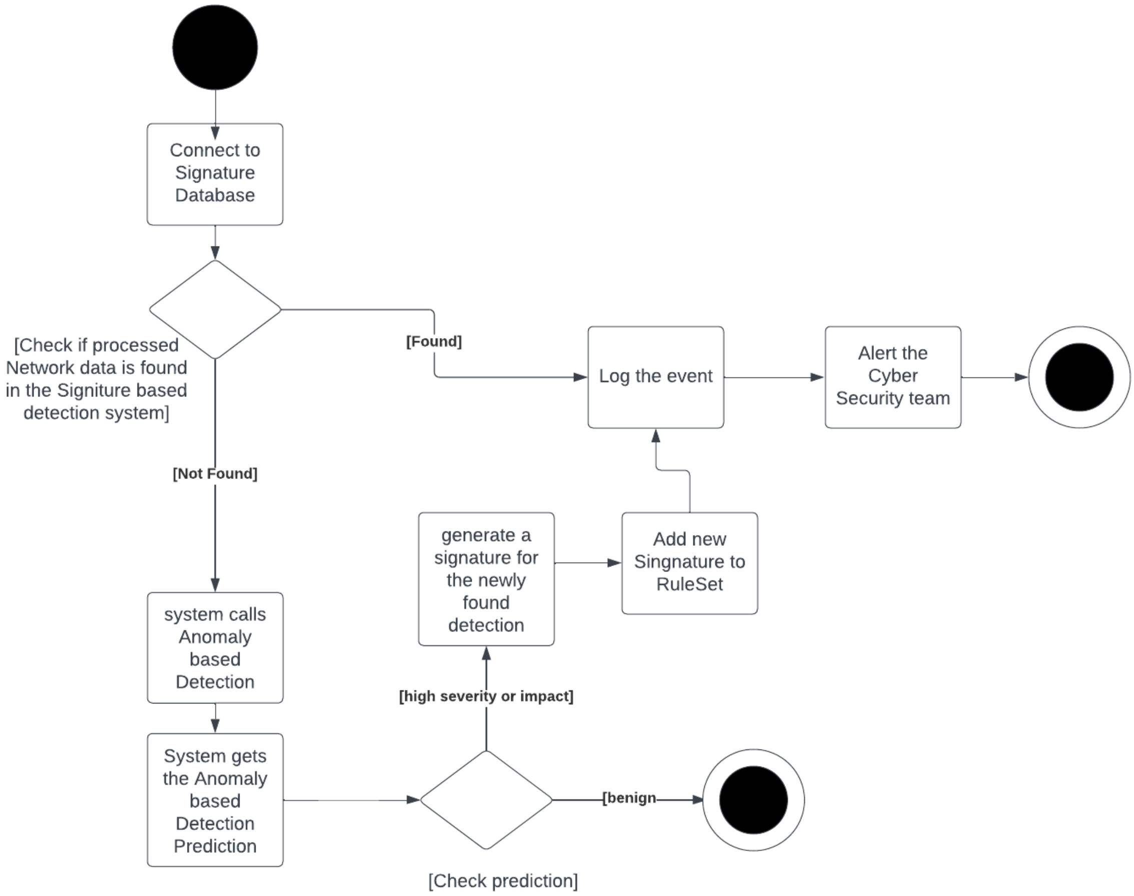
Activity Diagram (1) :
The figure below is the activity diagram when it first starts to collect network data and how it keeps collecting and passing data for real-time monitoring.


Start Timer
start collection network traffic
Temporary save network data
[Not Done]
[Check if Timer is Done]
[Done]
Preprocess the Network Data
passes Preprocess Network data For Detection

Description:
The NIDS will start a timer before collecting network traffic, then it will collect and temporarily store this raw network traffic in a buffer, when the timer finishes, the buffer containing raw network data will be passed to be preprocessed, and a new counter will start to continually collect traffic data and again temporary store the continuous network data to ensure real time monitoring.if the timer did not end, the system will keep saving the collected network

traffic in the temporary buffer. After Preprocessing the data it will be passed to the detection module.

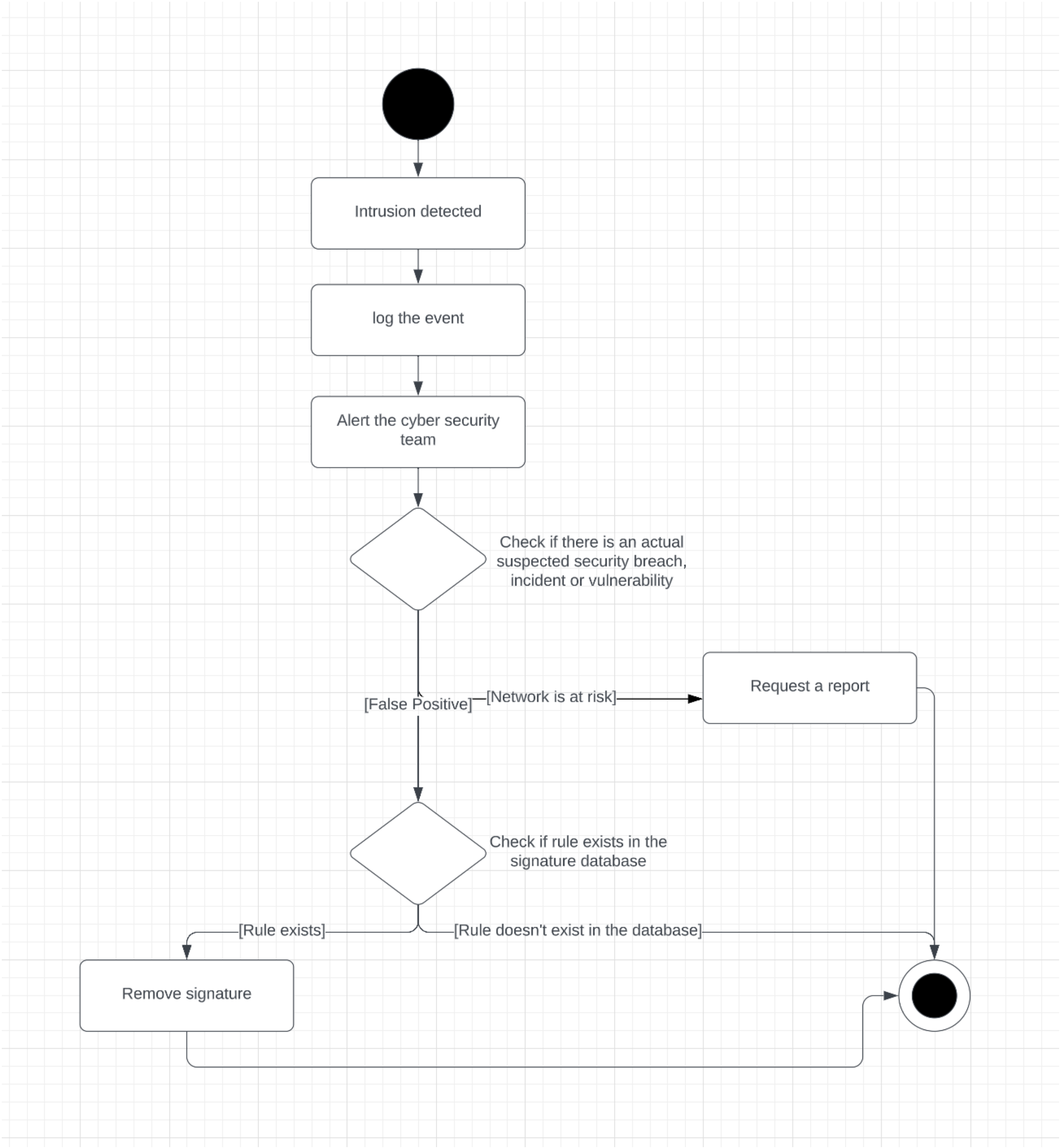
Activity Diagram (2):
This diagram below is the activity diagram for the main detection system for intrusions


Connect to Signature Database
[Check if processed Network data is found in the Signiture based detection system]
[Found]
Log the event
Alert the Cyber Security team
[Not Found]
system calls Anomaly based Detection
System gets the Anomaly based Detection Prediction
generate a signature for the newly found detection
Add new Singnature to RuleSet
[high severity or impact]
[benign
[Check prediction]

Description: After passing the processed data to the detection module, the module connects to the signature database as it will begin by searching for a match in the signature database, if match is found then the intrusion will be logged and an alert will be sent.if not initially found in signature based detection, system will call the anomaly based detection to try to predict the scope and likelihood of the event being an intrusion, after getting the prediction , if prediction is high and accurate and confirmed by security analyst it will create a new signature with the newly found network data to be added to the rule set, so it will be detected the next time instantly, then it will be logged and an alert will be sent to the cyber security team.

Activity Diagram (3):

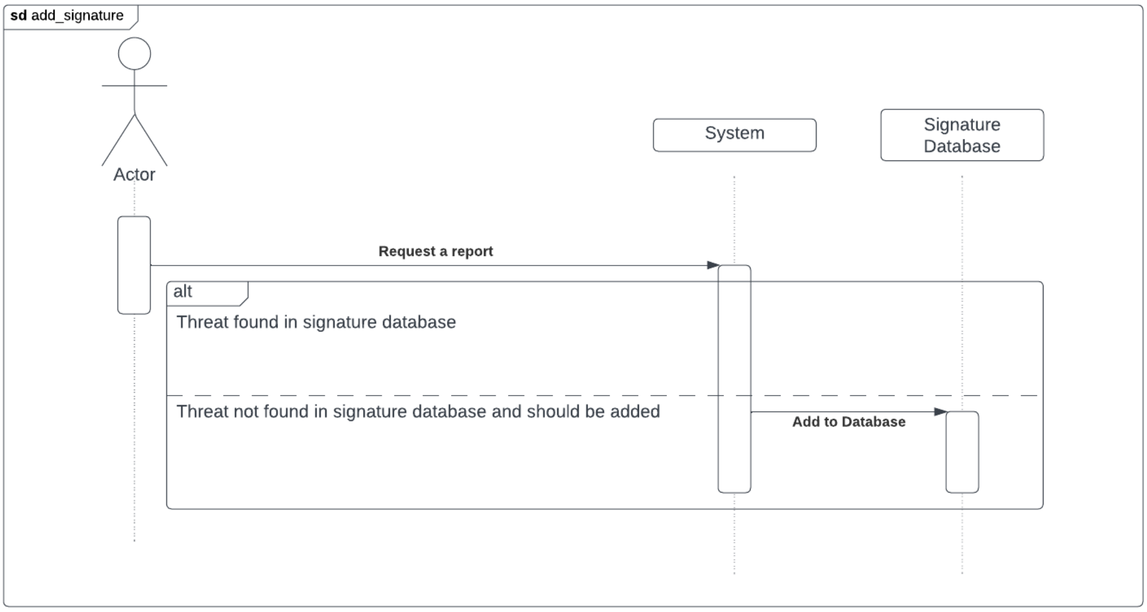
The diagram below is the activity diagram for investigating actual threats and false positive alerts


Intrusion detected
log the event
Alert the cyber security team
Check if there is an actual suspected security breach, incident or vulnerability
[False Positive]
[Network is at risk]
Request a report
Check if rule exists in the signature database
[Rule exists]
[Rule doesn't exist in the database]
Remove signature

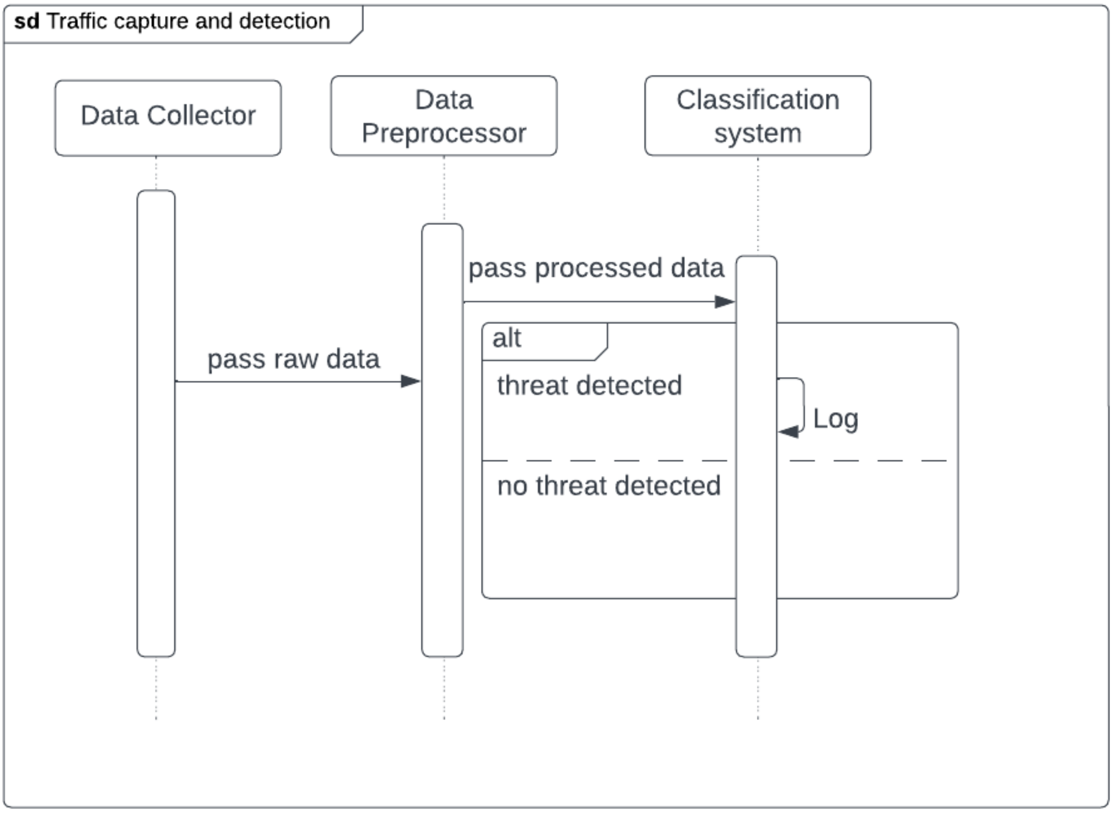
Description:
When the system alerts the cyber security team and logs the event. The team checks if there is an attempted intrusion, a successful breach or a security vulnerability is identified. If yes, they request a report from the system. If not and it's a false positive and it exists in the signature database, it should be removed.

3.1 Sequence Diagram:

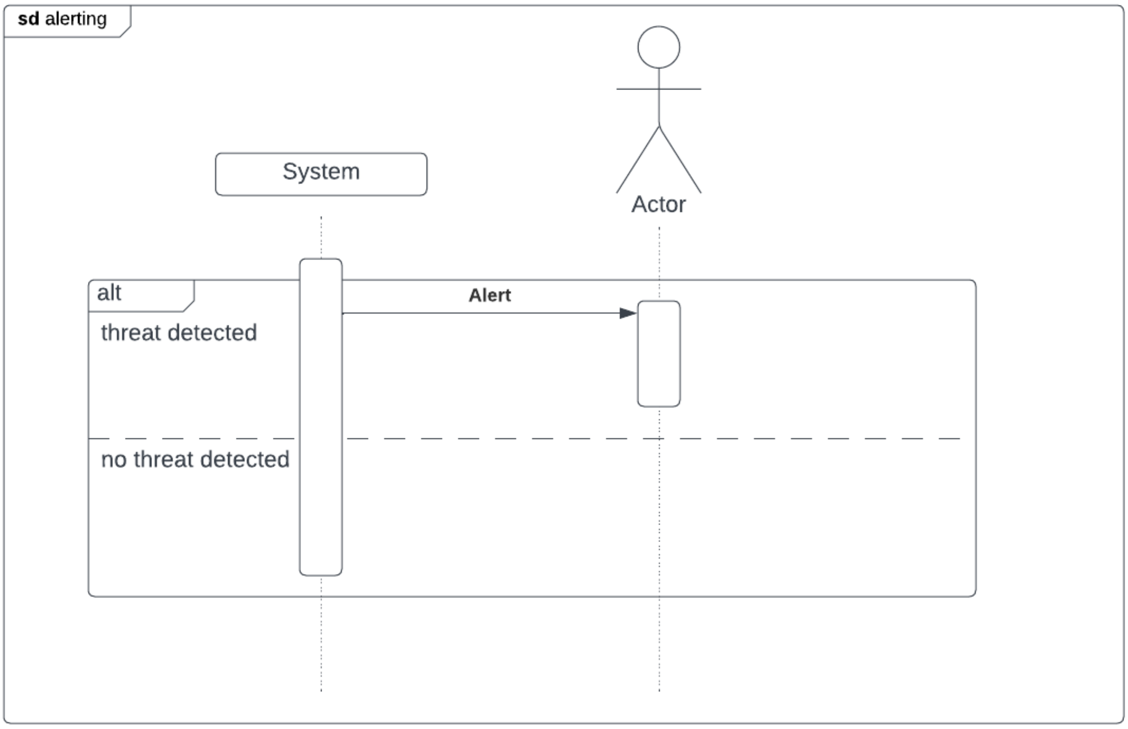
Sequence Diagram(1):


sd add_signature
Actor
System
Signature Database
Request a report
alt
Threat found in signature database
Threat not found in signature database and should be added
Add to Database

Sequence Diagram(2):



Sequence Diagram(3):

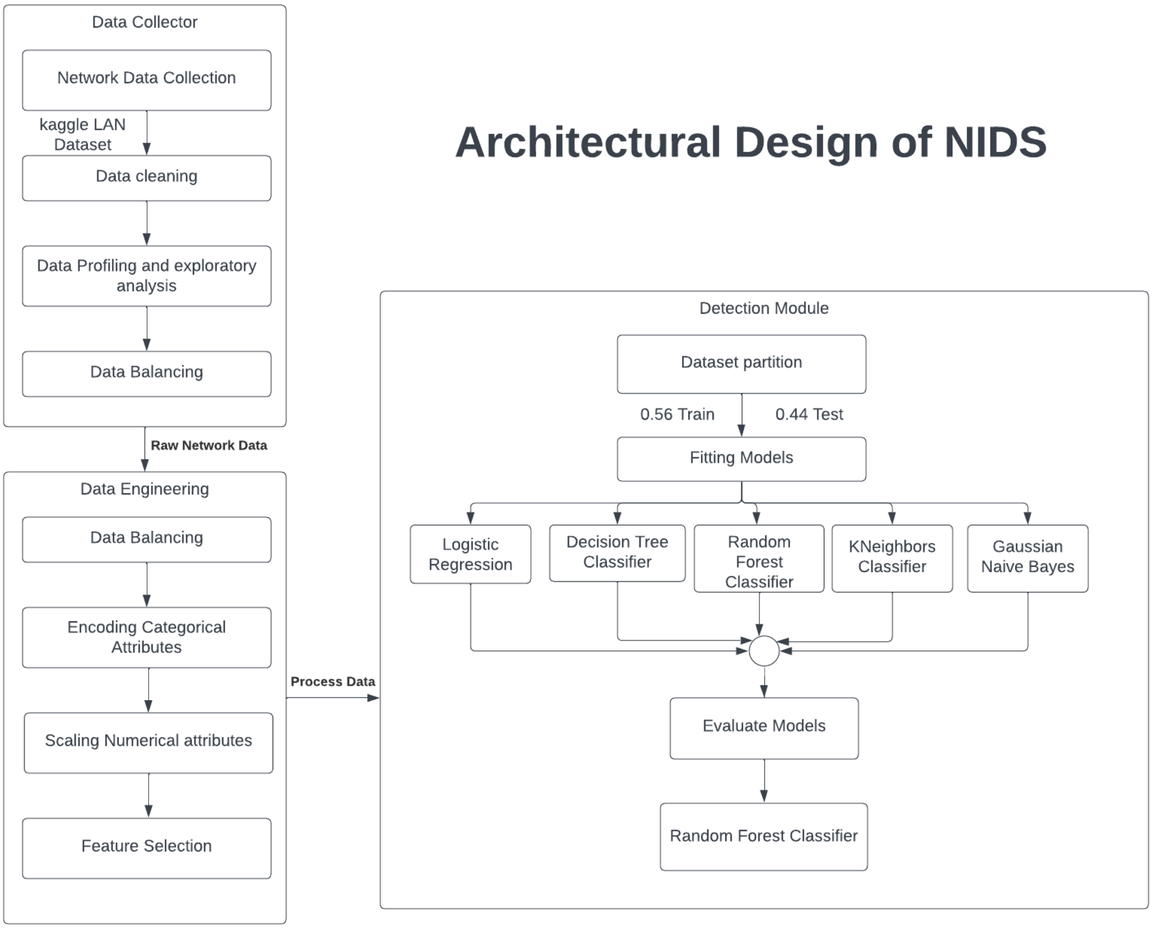


3.2 Proposal algorithm

The suggested method for network intrusion detection system (NIDS) seeks to locate and recognize probable intrusions or malicious activity within a network. The proposed algorithm makes use of a number of methods, such as network data collection, data preprocessing, anomaly detection, and signature-based analysis. In order to find well-known intrusion patterns or harmful signatures, the system checks the packet content against a large signature database.If the packet content was not found in the signature database, it will be passed to the anomaly based detection module. Overall, the NIDS is based on the use of statistical analysis, machine learning algorithms, or rule-based systems to identify anomalous patterns or behaviours that could point to an intrusion.

The system generates warnings and logs pertinent data upon the discovery of questionable activity or matches with recognized signatures. As new threats and changing network patterns emerge, the system is built to continually update its signature database under the supervision of the security operation centre (SOC). The system's capacity to efficiently identify and respond to various sorts of network intrusions is improved by this dynamic method.

The architectural design of our system consists of 3 modules:



Summary of Architectural Design:

Data collection module that collects live real time network data, then will be passed to the data preprocessing that will scale, encode and feature select the data to be ready to be processes, after that the processed data will be passed to the signature -based detection module that will check if the signature is found in the signature database. If the signature was not found in the database, the processed data will be passed to the anomaly based detection module that will apply the random forest classification model already trained and fitted to prior data. Based on the machine learning model, it will classify each network connection as anomaly or normal behaviour and will store them in an event log and alert the soc for anomaly classified connections. If the soc wanted to further investigate the suspicious event, the soc will request a system generated report in human readable format for a final confirmation from the soc. If the soc has confirmed the impact of the event , they can add this newly found malicious activity signature in the signature database, making the system dynamic, and will be detected faster and more efficiently through the signature based detection module in the NIDS.

Brief explanation of suggested classification models:

(1) Logistic regression:

A statistical model called logistic regression is used to forecast binary outcomes based on a number of independent factors. By fitting data to a logistic curve, which can be used to generate binary predictions, it represents the likelihood that an event will occur. The method functions by reducing the difference between actual results and projected probability.

(2) Decision Tree:

A decision tree is a flowchart-like paradigm for making choices based on a variety of factors. Decision trees are used as classification models in the context of machine learning to forecast the class of an input. Up until a halting requirement is satisfied, the input space is recursively divided into areas depending on the feature values. The final output is the class given to the leaf node that corresponds to the input, and each split is represented by a node in the tree.

(3) Random Forest Classifier:

An ensemble learning system known as a random forest classifier employs many decision trees to create predictions. It operates by building a collection of decision trees, each of which is trained using a random subset of both the data and the features. The outcome is then calculated by combining all of the trees' projections.

(4) K-Nearest Neighbors (KNN):

The KNN classifier is a straightforward technique that determines which class is more prevalent among a collection of input's K nearest neighbours in the training set. It is a non-parametric method, which means it makes no assumptions about how the data are distributed. K can either be explicitly specified or determined via cross-validation.

(5) Gaussian Naive Bayes Model:

Based on Bayes' theorem, the Gaussian Naive Bayes model is a probabilistic classification technique. In order to determine the likelihood that a given input belongs to a specific class, it makes the assumption that the characteristics are independent and normally distributed.

The likelihood of each feature value given the class is calculated by the algorithm, and the likelihood of the total input is obtained by multiplying these probabilities together. The class with the highest likelihood is the final result.

Data Profiling and Data engineering

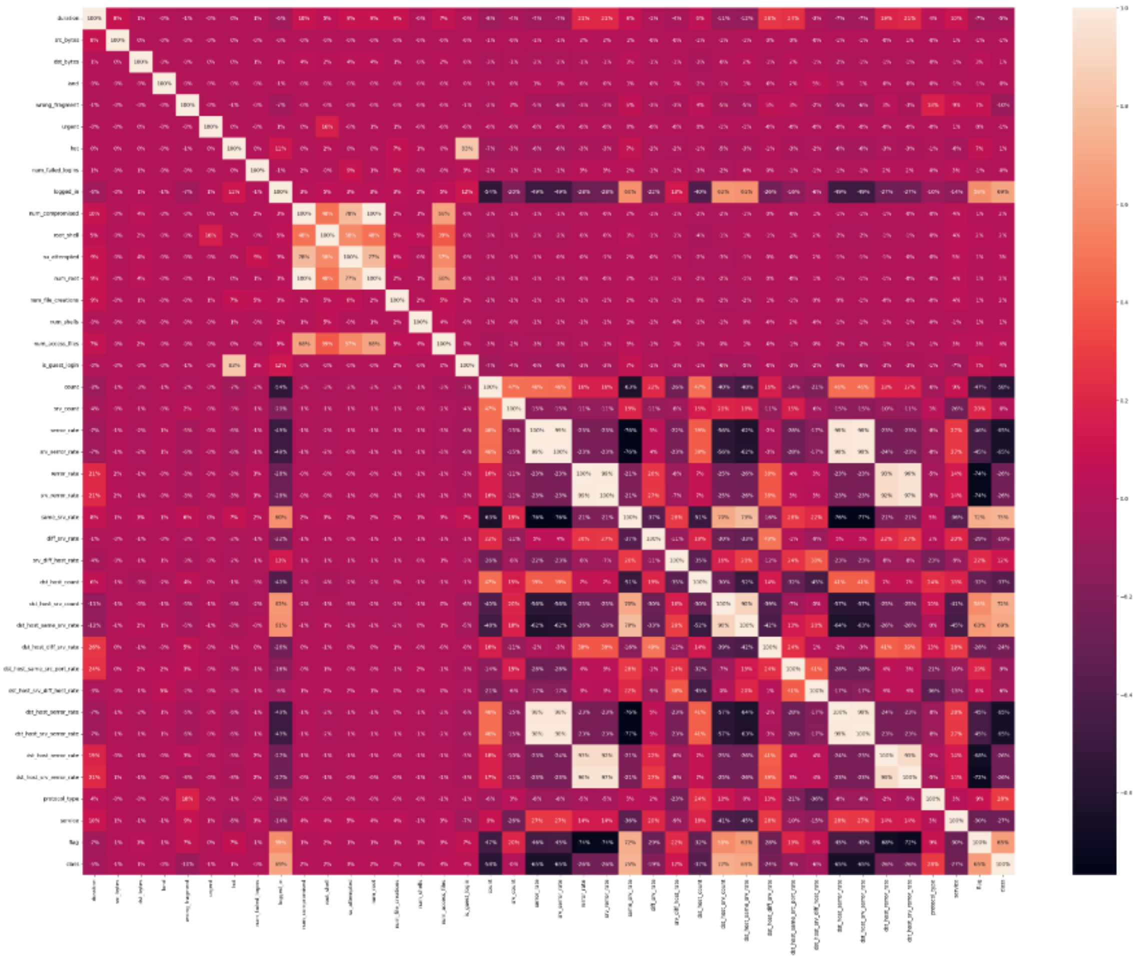
The audited dataset offered for study includes a wide range of simulated intrusions in a military network environment. It entails simulating a typical US Air Force LAN, created to replicate a real-world environment and subject to several attacks, then using that simulation to generate raw TCP/IP dump data. According to established protocols, each connection in the dataset reflects the data flow between a source IP address and a target IP address. These connections are either classified as normal or assigned a particular assault type. A connection record typically has 100 bytes of data in it. The dataset contains 41 characteristics that were created from normal and attack data and consist of both quantitative and qualitative aspects. There are two subcategories in the class variable: "Normal" and "Anomalous."

The dataset we used from Kaggle is the NIDS dataset [18], which contains 25192 connections records shown in the table below:

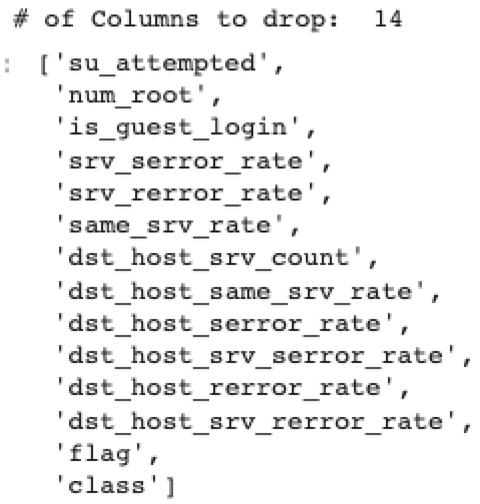
	Normal	Abnormal
Total	13449	11743

Which is split into 14107 training set and 11085 testing setl.

Feature selection is the process of selecting a subset of relevant features (variables, predictors) for use in model construction. The purpose of feature selection is to reduce the number of input variables to those that are most useful in predicting the target variable.
After creating the heatmap and visualising the correlations between the features, you can choose to remove highly correlated features. A common threshold is to remove features with a correlation coefficient above 0.7 or below -0.7.



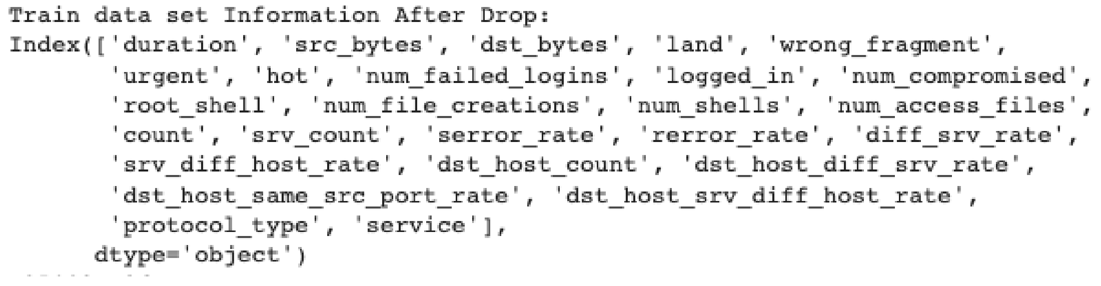
Based on the heat map we found that these features need to be dropped:



```
 # of Columns to drop:  14
```

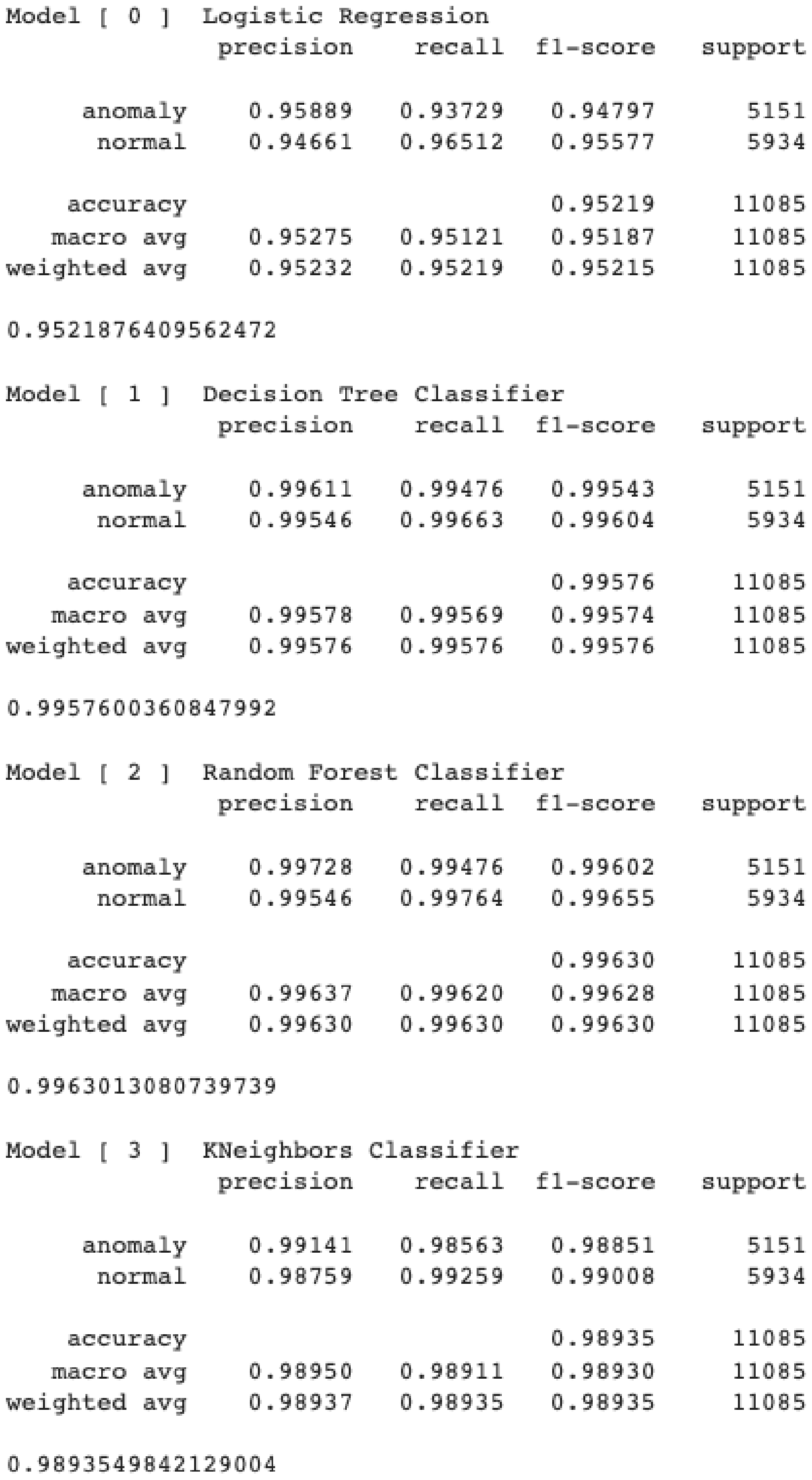
```
: ['su_attempted',
   'num_root',
   'is_guest_login',
   'srv_serror_rate',
   'srv_rerror_rate',
   'same_srv_rate',
   'dst_host_srv_count',
   'dst_host_same_srv_rate',
   'dst_host_serror_rate',
   'dst_host_srv_serror_rate',
   'dst_host_rerror_rate',
   'dst_host_srv_rerror_rate',
   'flag',
   'class']
```

Train data set Attributes After Drop:

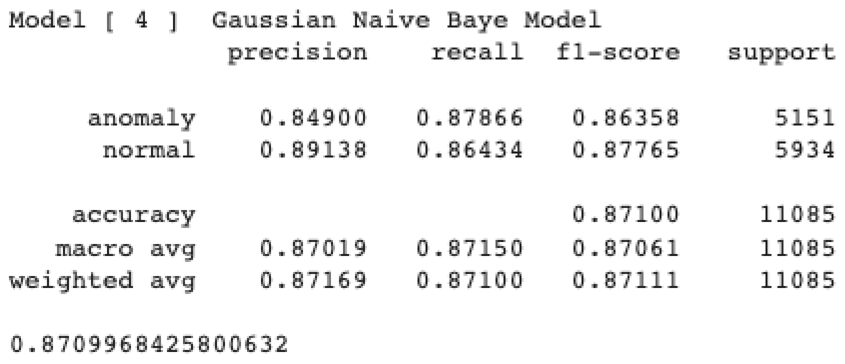


```
Train data set Information After Drop:
Index(['duration', 'src_bytes', 'dst_bytes', 'land', 'wrong_fragment',
       'urgent', 'hot', 'num_failed_logins', 'logged_in', 'num_compromised',
       'root_shell', 'num_file_creations', 'num_shells', 'num_access_files',
       'count', 'srv_count', 'serror_rate', 'rerror_rate', 'diff_srv_rate',
       'srv_diff_host_rate', 'dst_host_count', 'dst_host_diff_srv_rate',
       'dst_host_same_src_port_rate', 'dst_host_srv_diff_host_rate',
       'protocol_type', 'service'],
      dtype='object')
```

3.3 Simulation result



```
Model [ 0 ]  Logistic Regression
               precision    recall  f1-score   support

     anomaly     0.95889   0.93729   0.94797      5151
      normal     0.94661   0.96512   0.95577      5934

    accuracy                         0.95219     11085
   macro avg     0.95275   0.95121   0.95187     11085
weighted avg     0.95232   0.95219   0.95215     11085

0.9521876409562472

Model [ 1 ]  Decision Tree Classifier
               precision    recall  f1-score   support

     anomaly     0.99611   0.99476   0.99543      5151
      normal     0.99546   0.99663   0.99604      5934

    accuracy                         0.99576     11085
   macro avg     0.99578   0.99569   0.99574     11085
weighted avg     0.99576   0.99576   0.99576     11085

0.9957600360847992

Model [ 2 ]  Random Forest Classifier
               precision    recall  f1-score   support

     anomaly     0.99728   0.99476   0.99602      5151
      normal     0.99546   0.99764   0.99655      5934

    accuracy                         0.99630     11085
   macro avg     0.99637   0.99620   0.99628     11085
weighted avg     0.99630   0.99630   0.99630     11085

0.9963013080739739

Model [ 3 ]  KNeighbors Classifier
               precision    recall  f1-score   support

     anomaly     0.99141   0.98563   0.98851      5151
      normal     0.98759   0.99259   0.99008      5934

    accuracy                         0.98935     11085
   macro avg     0.98950   0.98911   0.98930     11085
weighted avg     0.98937   0.98935   0.98935     11085

0.9893549842129004
```



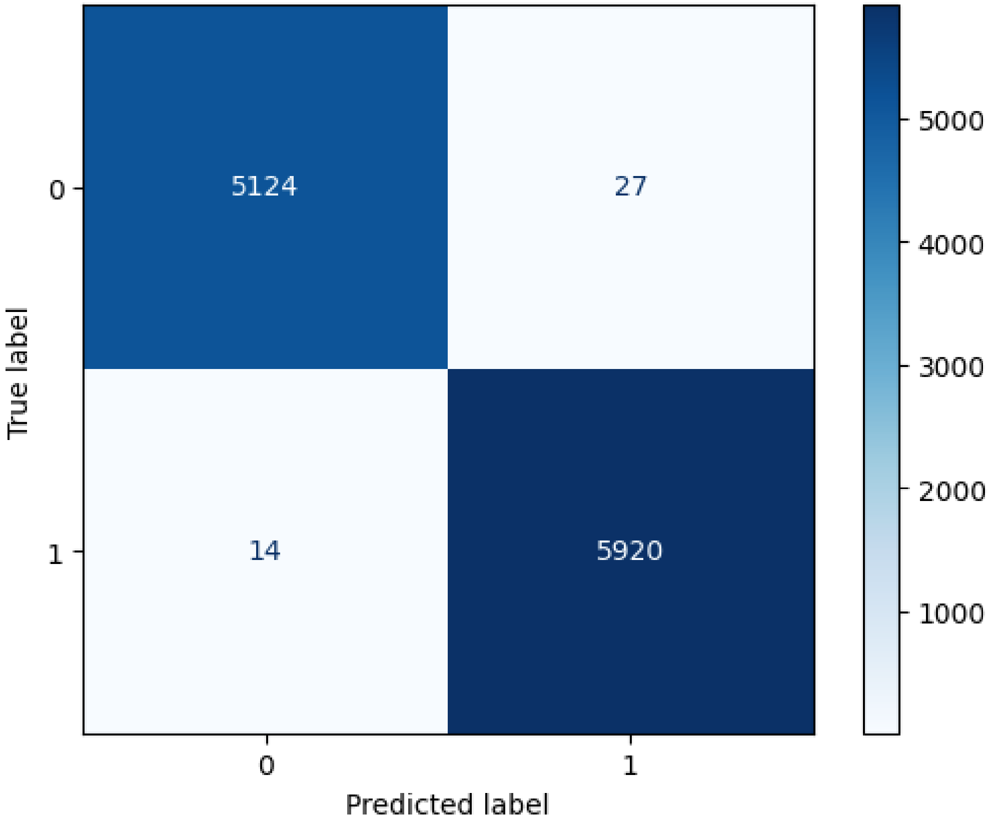
```
Model [ 4 ]  Gaussian Naive Baye Model
              precision    recall  f1-score   support

     anomaly    0.84900   0.87866   0.86358      5151
      normal    0.89138   0.86434   0.87765      5934

    accuracy                        0.87100     11085
   macro avg    0.87019   0.87150   0.87061     11085
weighted avg    0.87169   0.87100   0.87111     11085

0.8709968425800632
```

Based on our research we found that model 2 is the best fit because :

The Random Forest Classifier performs exceptionally well, with an accuracy of 0.99630 on the testing data.

- It achieves high precision, recall, and F1-scores for both classes, indicating accurate and reliable classification.
- The model shows a slightly higher recall for the "normal" class, implying its strength in correctly identifying normal samples.
- With its outstanding accuracy and balanced performance, the Random Forest Classifier is a robust model for this dataset.

Visualiization of Predicted Results



- True Positives (TP): The model correctly classified 5124 samples as positive (anomaly in this case).
- False Positives (FP): The model incorrectly classified 27 samples as positive (anomaly) when they were actually negative (normal).
- False Negatives (FN): The model incorrectly classified 14 samples as negative (normal) when they were actually positive (anomaly).
- True Negatives (TN): The model correctly classified 5920 samples as negative (normal).

These findings provide insights into the performance of the selected model on the test data. It correctly identified a substantial number of anomalies (TP) and normal samples (TN). However, it had a small number of false positives (FP) and false negatives (FN), indicating some misclassifications.

3.4 The user will use your application (Compare your Algorithm and Prior work)

Similar results were discovered for the performance of machine learning techniques for network intrusion detection systems (NIDS), building on the literature survey reported in [8]. In order to identify network intrusions, the study demonstrated the efficiency of numerous core algorithms, including Naive Bayes, Decision Trees, k-Nearest Neighbors, Random Forest, and Support Vector Machines. The Random Forest method was shown to perform better than the other algorithms in terms of detection rate while retaining a low false alarm rate, which is in line with the research. This is consistent with the goal of limiting the amount of false positives while attaining high accuracy in recognizing network intrusions.

The study also stressed the significance of feature engineering and selection, which are essential for improving the accuracy and effectiveness of machine learning-based NIDS. Overall, our results support the literature and emphasise the need of using effective algorithms, such Random Forest, to build reliable and precise network intrusion detection systems.

In the literature review[14], it is emphasised how well supervised approaches perform for detecting network intrusions with high accuracy and dependability. Researchers showed the higher performance of supervised approaches on the KDD Cup 99 dataset by choosing pertinent features and applying several machine learning algorithms, such as decision trees, random forests, and support vector machines. By identifying underlying patterns and adapting to various settings, these approaches, which rely on labelled training data, provide reliable classification and detection of network intrusions. The improvement of intrusion detection systems' accuracy and efficacy through the application of supervised approaches in feature selection aids in preventive defence against cybersecurity threats.

For full detailed explanation of the whole process, please go to the following github repository:

https://github.com/Karam-Issa/Network-Intrusion-Sytem/blob/main/nidsMLModel.ipynb

3.5 Conclusion and future work

In summary, this paper offers a powerful approach to detect malicious threats in a network environment. Five models were developed and tested on a Kaggle dataset which consists of a wide variety of intrusions simulated in a military network environment. The proposed models (Logistic Regression, Decision Tree Classifier, Random Forest Classifier, KNeighborsClassifier, Gaussian Naive Bayes) resulted in the accuracies 0.95219, 0.99576, 0.99630, 0.98935, 0.87100 respectively. Overall, among the proposed models in this research, the Random Forest Classifier produced the best results. It successfully detected a significant amount of anomalies (true positives, TP) and normal samples (true negatives, TN). However, it exhibited a minor number of false positives (FP) and false negatives (FN), suggesting a few instances of misclassifications.

Our main goal in the future is to improve our system to collect real-time network traffic, perform real-time analysis, respond to threats and continuously monitor and update the IDS to stay effective against evolving threats.
In addition, we are considering delegating our signature-based detection system to Snort, an open-source network intrusion detection and prevention system. With the help of a committed community of security experts, Snort offers a trustworthy and current signature database. By utilising Snort's experience, we can improve the efficiency of our security architecture while concentrating on other vital areas and having access to a wide range of signatures for identifying and reducing network risks.

Furthermore, the use of clustering algorithms in the machine learning model for thorough network intrusion detection is a fascinating direction for future research. While clustering algorithms can provide a more detailed picture by classifying network traffic into discrete groups based on their similarities, classification models are excellent at differentiating between typical and well-known attack patterns. This may reveal novel attack methods or variants, as well as insights into the underlying structures and patterns of network invasions.

Clustering methods enable the detection system to recognize unusual network events that might not match predetermined attack fingerprints. Clustering can pick up on minute changes in network traffic, enabling the discovery of previously unknown attack patterns or zero-day vulnerabilities. Furthermore, clustering can give a more detailed picture of attack campaigns by combining comparable actions and facilitating improved attribution and mitigation techniques.

However, it is important to keep in mind that clustering-based intrusion detection may present difficulties, including choosing suitable clustering methods, dealing with high-dimensional data, and figuring out the best cluster thresholds. To overcome these obstacles, further study and testing are needed to create reliable and effective clustering methods designed particularly for network intrusion detection. However, employing clustering techniques may increase the sophistication and potency of network intrusion detection systems, providing greater security against new and growing cyber threats.

References:

1. “2023 must-know cyber attack statistics and Trends,” Embroker, 06-Mar-2023. [Online]. Available: https://www.embroker.com/blog/cyber-attack-statistics/. [Accessed: 18-Mar-2023].
2. INTRUSION Inc., “Cybercrime to cost the world $10.5 trillion annually by 2025,” GlobeNewswire News Room, 18-Nov-2020. [Online]. Available: https://www.globenewswire.com/news-release/2020/11/18/2129432/0/en/Cybercrime-To-Cost-The-World-10-5-Trillion-Annually-By-2025.html. [Accessed: 18-Mar-2023].
3. “World economic forum.” [Online]. Available: https://www3.weforum.org/docs/WEF_Global_Risk_Report_2020.pdf. [Accessed: 18-Mar-2023].
4. B. Lutkevich, “What is an intrusion detection system (IDS)? definition from searchsecurity,” Security, 07-Oct-2021. [Online]. Available: https://www.techtarget.com/searchsecurity/definition/intrusion-detection-system. [Accessed: 18-Mar-2023].
5. “Nids – A Guide to Network Intrusion Detection Systems,” Bulletproof.co.uk. [Online]. Available: https://www.bulletproof.co.uk/blog/network-intrusion-detection-systems. [Accessed: 18-Mar-2023].
6. N-able, “Intrusion detection system (IDS): Signature vs. anomaly-based - N-able,” N, 05-May-2021. [Online]. Available: https://www.n-able.com/blog/intrusion-detection-system. [Accessed: 18-Mar-2023].
7. “Semi-supervised learning ,” DataRobot AI Platform, 01-Jul-2022. [Online]. Available: https://www.datarobot.com/blog/semi-supervised-learning/#:~:text=Semi%2Dsupervis
8. d%20machine%20learning%20is,large%20amount%20of%20labeled%20data. [Accessed: 18-Mar-2023]
9. “Network intrusion detection system: A machine learning approach.” [Online]. Available: https://www.researchgate.net/publication/220468036_Network_intrusion_detection_system_A_machine_learning_approach. [Accessed: 19-Mar-2023].
10. Author links open overlay panelEmad E. Abdallah and A. I. this paper, “Intrusion detection systems using supervised Machine Learning Techniques: A survey,” Procedia Computer Science, 27-Apr-2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050922004422?ref=pdf_download. [Accessed: 19-Mar-2023].
11. G. Pu, F. Dong, and L. Wang, “Research online - university of wollongong,” Site, 2021. [Online]. Available: https://ro.uow.edu.au/cgi/viewcontent.cgi?article=5287&context=eispapers1. [Accessed: 19-Mar-2023].
12. https://www.researchgate.net/publication/356301665_Network_Intrusion_Detection_Systems_Using_Supervised_Machine_Learning_Classification_and_Dimensionality_Reduction_Techniques_A_Systematic_Review.
13. M. Labonne, “Anomaly-based network intrusion detection using machine learning.” [Online]. Available: https://theses.hal.science/tel-02988296v1/document. [Accessed: 19-Mar-2023].
14. O. Olabiyi and A. Osofisan, “Deep reinforcement learning and games [guest editorial] - IEEE xplore.” [Online]. Available: https://ieeexplore.ieee.org/document/8764633. [Accessed: 19-Mar-2023].

15. Y. Wu, “Forvizor: Visualizing Spatio-temporal team formations in soccer | IEEE ...,” 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8440804. [Accessed: 19-Mar-2023].
16. P. Vanin, T. Newe, L. L. Dhirani, E. O’Connell, D. O’Shea, B. Lee, and M. Rao, “A study of network intrusion detection systems using Artificial Intelligence/machine learning,” MDPI, 18-Nov-2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/22/11752. [Accessed: 19-Mar-2023].
17. Jan Lansky, “Deep learning-based Intrusion Detection Systems: A systematic review,” 2021. [Online]. Available: https://www.researchgate.net/publication/353473498_Deep_Learning-Based_Intrusion_Detection_Systems_A_Systematic_Review. [Accessed: 19-Mar-2023].
18. https://www.kaggle.com/datasets/sampadab17/network-intrusion-detection

.