TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

KANTIPUR ENGINEERING COLLEGE

DHAPAKHEL, LALITPUR

SUBJECT CODE: CT654

A MINOR PROJECT PROPOSAL REPORT ON

NETWORK INTRUSION DETECTION SYSTEM USING RANDOM FOREST

SUBMITTED BY:

Aakash Rana	KAN078BCT004
Aayush Maharjan	KAN078BCT005
Alka Basnet	KAN078BCT008
Bigyan Moktan	KAN078BCT021

SUBMITTED TO:

DEPARTMENT OF COMPUTER AND ELECTRONICS ENGINEERING

IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR IN COMPUTER ENGINEERING

JULY, 2024

A MINOR PROJECT PROPOSAL REPORT ON

NETWORK INTRUSION DETECTION SYSTEM USING RANDOM FOREST

SUBMITTED BY:

Aakash Rana	**KAN078BCT004**
Aayush Maharjan	**KAN078BCT005**
Alka Basnet	**KAN078BCT008**
Bigyan Moktan	**KAN078BCT021**

IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR IN COMPUTER ENGINEERING

SUBMITTED TO:

DEPARTMENT OF COMPUTER AND ELECTRONICS ENGINEERING

KANTIPUR ENGINEERING COLLEGE

AFFILIATED TO TRIBHUVAN UNIVERSITY

DHAPAKHEL, LALITPUR

JULY, 2024

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

ABSTRACT

In today's interconnected world, cybersecurity is vital for protecting organizations' assets and sensitive information from threats. An Intrusion Detection System (IDS) continuously watches network activity and system operations to quickly detect and respond to possible security breaches. This proposal aims to create an IDS using advancement of machine learning like the Random Forest Algorithm making it better at spotting unusual activities, detecting suspicious patterns, and providing helpful insights to prevent threats. Our goal is to create a robust IDS that handles complex data efficiently and adapts to new threats. Real-time monitoring will ensure we quickly detect unauthorized access attempts or suspicious behavior, allowing prompt responses to reduce risks. Seamless integration with existing security systems will improve scalability and efficiency across various organizational sizes and network setups. This proposal highlights our commitment to using advanced technology to enhance cybersecurity, reduce cyber risks, protect sensitive data, and get better accuracy in detecting intrusions.

Keywords : CSE-CIC-IDS2018, Intrusion detection, Machine learning , Random Forest , Decision tree

LIST OF ABBREVIATIONS

AB	AdaBoost (Adaptive Boosting)
CART	Classification and Regression Tree
CIC	Canadian Institute for Cybersecurity
CSE	Computer Science and Engineering
DDoS	Distributed Denial of Service
DL	Deep Learning
DT	Decision Tree
DoS	Denial of Service
FAR	False Alarm Rate
FTP	File Transfer Protocol
GB	Gradient Boosting
IDS	Intrusion Detection System
ISCX	Information Security Centre of Excellence
KDD	Knowledge Discovery in Database
KNN	K-Nearest Neighbor
LDA	Linear Discriminant Analysis
LOIC	Low Orbit Ion Cannon
ML	Machine Learning
NIDS	Network Based Intrusion Detection System

OpenSSL	Open Secure Sockets Layer
RF	Random Forest
SIEM	Security Information and Event Management
SMO	Sequential Minimal Optimization
SMOTE	Synthetic Minority Over-sampling Technique
SQL	Structured Query Language
SSH	Secure Shell
SVM	Support Vector Machine
XSS	Cross-Site Scripting

CHAPTER 1 : INTRODUCTION

1.1 Background

Over the past decade, advancements in internet and communication technologies have significantly highlighted the importance of network security as a key area of research. To protect the network and its assets within cyberspace, tools such as firewalls, antivirus software, and intrusion detection systems (IDS) are employed[1]. Among these, network-based intrusion detection systems (NIDS) are crucial for providing security. They continuously monitor network traffic to detect any malicious or suspicious activities[2].

Jim Anderson introduced the concept of IDS in 1980[3]. Since then, many IDS products have been developed and improved to meet the changing demands of network security. However, recent advancements in technology have led to significant growth in network size and the complexity of applications handled by network nodes. This has generated large volumes of critical data exchanged among nodes, posing new challenges for ensuring their security. The rise of both evolved and novel attacks has made nearly every network node susceptible to security threats. For instance, compromising a data node can profoundly impact an organization's reputation and finances. Current IDSs have shown limitations in effectively detecting various types of attacks, including zero-day threats, and in minimizing false alarm rates (FAR)[4]. Consequently, there is a rising need for efficient, precise, and cost-effective NIDS solutions to provide robust network security.

To tackle these challenges, researchers are increasingly developing IDSs using machine learning techniques. Machine learning, an artificial intelligence approach, excels in automatically extracting valuable insights from large datasets[5]. IDSs based on machine learning can achieve effective detection rates provided there is ample training data available. These models also demonstrate the ability to generalize well, enabling them to detect various attack variants and novel threats. Importantly, machine learning-based IDSs require less dependency on domain-specific knowledge, making them more straightforward to design and implement. This circumstance justifies the relevance of conducting applied research in this area aimed at developing specific proposals for building detection models and the prospects for their practical implementation.

The research aims to develop a machine learning model for constructing a computer attack detection system. Achieving this goal involves solving the following main tasks: selecting a training dataset, evaluating the significance of features and forming the feature space, justifying the choice of a machine learning model and selecting quasi-optimal model parameters, evaluating the quality and testing the model in real conditions.

1.2 Problem Statement

Without a good detection system, just relying on firewall and antivirus software isn't enough to stop unauthorized people from accessing a computer network. These intruders could steal data, including sensitive information, and even cause a Denial of Service (DoS) attack, disrupting normal operations.

With the rapid evolution of information technologies and the increasing complexity of cyber threats, traditional signature-based Intrusion Detection Systems (IDS) face limitations in effectively detecting novel and evolving types of attacks. This necessitates the development of advanced IDS solutions leveraging machine learning techniques. The challenge lies in designing and implementing IDS models capable of robustly detecting diverse attack variants, optimizing feature selection, and parameters while ensuring practical applicability and scalability in real-world environments. Addressing these challenges requires innovative approaches to enhance IDS capabilities and mitigate cybersecurity risks effectively.

1.3 Objectives

The primary objectives of this project are as follows:

i. To detect and classify network intrusions accurately to enhance overall network security posture.
ii. To optimize the RF model to minimize false alarms and improve accuracy in distinguishing between normal network behavior and potential intrusions.

1.4 Application and Scope

The main application and scopes of the proposal are :

i. Telecommunications: IDS monitors communication networks for cyber attacks,

ensuring uninterrupted service and regulatory compliance in telecommunications sectors with robust network traffic analysis and threat mitigation capabilities.

ii. Financial Institutions: IDS secures financial transactions and customer data from cyber attacks, maintaining trust in banking systems with real-time monitoring capabilities and integration into existing security infrastructures.

iii. Government Organizations: IDS defends government networks and critical infrastructure against cyber threats, supporting national security efforts with detailed network analysis and proactive threat management capabilities.

iv. Corporate Environments: IDS detects insider threats, malware, and unauthorized access within corporate networks, protecting intellectual property and business-sensitive information while ensuring compliance with regulatory requirements.

v. Retail and E-commerce: IDS secures online transactions and customer data, ensuring compliance with PCI standards, and provides continuous monitoring and rapid response to cyber threats targeting retail and e-commerce platforms.

1.5 Project Features

These features collectively ensure robust security monitoring, effective threat detection, and efficient incident response capabilities within our IDS project.

i. Machine Learning and AI: Utilize advanced algorithms to enhance threat detection accuracy and adaptability to new and evolving threats.

ii. Signature-based Detection: Match observed patterns against a database of known attack signatures to recognize and respond to specific threats.

iii. Behavioral Analysis: Monitor and analyze entity behavior to detect abnormal activities indicative of security breaches.

1.6 Feasibility Analysis

A feasibility study serves several critical purposes in the context of evaluating a project. The feasibility analysis is crucial for the project development. It can be segmented into the following components:

1.6.1 Technical Feasibility

The proposed Intrusion Detection System (IDS) will leverage machine learning algorithms and be developed using the Python programming language. This choice ensures compatibility across different operating systems, including Windows and Linux, and facilitates integration with existing network infrastructures. The system's capability to analyze network traffic patterns and detect intrusions in real-time is expected to enhance its technical feasibility. By utilizing efficient computational resources, the IDS aims to monitor and respond to potential security breaches promptly.

1.6.2 Operational Feasibility

Operationally, the proposed Intrusion Detection System (IDS) will be designed with a user-friendly interface that simplifies monitoring and management for security personnel. It will support easy deployment and operation across diverse network environments, ranging from small businesses to large enterprises. The system's ability to generate actionable alerts and provide comprehensive threat analysis is intended to help organizations respond proactively to security incidents.By minimizing false positives and streamlining incident response workflows, the IDS aims to enhance operational efficiency and reduce the likelihood of prolonged network downtime or data loss. This operational readiness is expected to ensure that organizations can maintain compliance with cybersecurity regulations and effectively protect their digital assets.

1.6.3 Economic Feasibility

The IDS project is economically feasible because it doesn't require expensive software licenses or specialized hardware. It addresses a growing market need for effective cybersecurity solutions, which is crucial for protecting sensitive data and maintaining operational continuity. Both government agencies and businesses are increasingly investing in cybersecurity measures to safeguard against evolving cyber threats, making this project potentially lucrative. By enhancing network security and reducing the risk of costly data breaches or operational disruptions, the IDS project offers significant economic value.

1.6.4 Schedule Feasibility

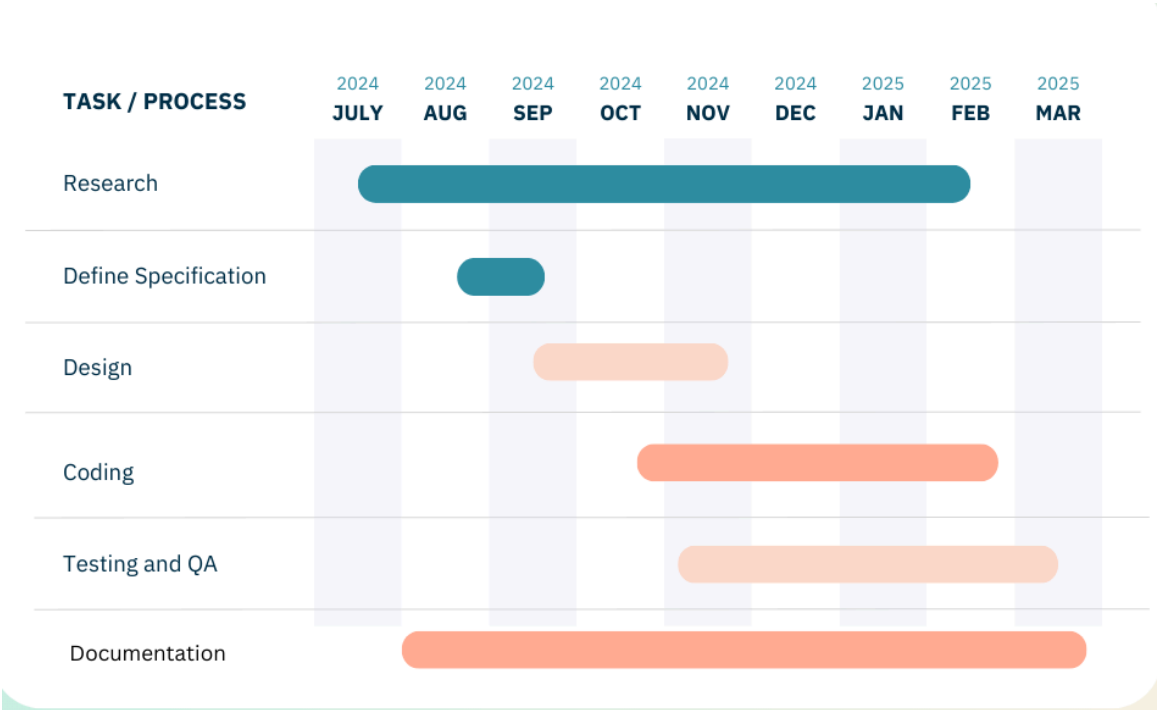

Figure 1.1 Gantt Chart

1.7 Project Requirements

1.7.1 Development Requirements

Table 1.1 Development Requirements

Hardware Requirements	Software Requirements
RAM: 8GB and dedicated graphics card.	Operating system: Windows 10, 11 or Linux.
Intel i5 series processor or higher.	Web browser: Google Chrome, Brave.
	Environment and IDE: Jupyter Notebook

1.7.2 Deployment Requirements

Table 1.2 Deployment Requirements

Hardware Requirements	Software Requirements
PC with 4GB RAM.	Web browser: Google Chrome, Brave.
Intel i5 series processor.	Internet connection.

CHAPTER 2 : LITERATURE REVIEW

2.1 Related Papers

R.A.Jamadar(2017) developed a network-based intrusion detection system (NIDS) using ensemble method, specifically testing it on the CIC-IDS 2017 dataset. Their approach centered around employing a decision tree classifier, known for its tree-like structure where each leaf node represents a decision or outcome based on the input features. This methodology enabled them to classify network traffic data into either benign or malicious (attack) categories by recursively partitioning the training set, aiming to achieve purity in the resulting subsets[6]. The reliance on the CIC-IDS 2017 dataset means that the system's ability to detect and classify new and emerging cyber threats may be limited. NIDS systems need regular updates and training on current datasets to stay effective against evolving attack strategies.

Schonlau and Zou (2020) provide an in-depth examination of the random forest algorithm, emphasizing its utility in statistical learning. The paper begins by highlighting the algorithm's origins, tracing back to Breiman (2001), and explaining its core principle of constructing multiple decision trees and aggregating their results for improved prediction accuracy. This ensemble approach, known as bootstrap aggregating or bagging, is shown to mitigate the overfitting problem commonly associated with single decision trees[7]. Key findings of the paper include the random forest's robustness in handling datasets with a large number of predictor variables, its capacity to produce reliable variable importance measures, and its generally superior predictive accuracy compared to linear models. The authors also discuss the algorithm's computational efficiency and scalability, making it suitable for a wide range of applications in social science research. In conclusion, Schonlau and Zou (2020) offer a comprehensive guide to the random forest algorithm, making it accessible and practical for social scientists. Their work not only elucidates the theoretical underpinnings and operational mechanics of the algorithm but also provides practical tools and examples to facilitate its adoption and application in empirical research.

Abdulhammed et al. (2019) conducted an in-depth study to improve the effectiveness of Intrusion Detection Systems (IDS) using the CIDDS-001 dataset. They explored several machine learning algorithms including Voting, Deep Neural Networks, Variational

Autoencoder, Random Forest, and stacking methods. Their focus was on handling imbalanced datasets, which is a common challenge in IDS development due to the unequal distribution of attack and normal instances[8]. The study also emphasizes the necessity of using up-to-date and balanced datasets to train machine learning models for IDS, ensuring better performance metrics in real-world scenarios. The effectiveness of each algorithm was evaluated in terms of detection rates, and the results were compared with those from previous studies to underscore improvements and validate their methodologies. Overall, Abdulhammed et al.'s research underscores the importance of integrating advanced machine learning techniques and addressing data imbalance to enhance the detection capabilities of IDS, ultimately contributing to more secure network environments.

“An adaptive ensemble machine learning model for intrusion detection”(2019) paper investigated an adaptive ensemble learning model for network intrusion detection using the NSL-KDD dataset. They aimed to enhance detection accuracy by leveraging multiple machine learning algorithms, including Decision Tree, Random Forest, K-Nearest Neighbor, and Deep Neural Networks. The ensemble model used an adaptive voting mechanism to combine the strengths of these individual classifiers. In their experiments, they observed that the ensemble adaptive model achieved an overall detection accuracy of 85.2%, which was superior to the performance of any single classifier[9]. The study demonstrated the effectiveness of ensemble methods in improving the robustness and accuracy of intrusion detection systems. However, they also noted that the proposed methodology did not perform satisfactorily for detecting weaker attack classes, indicating an area for further improvement.

The paper “Network intrusion detection system: A systematic study of machine learning and deep learning approaches”,2020 provides a comprehensive survey on the application of machine learning (ML) and deep learning (DL) techniques in designing Network Intrusion Detection Systems (NIDS). With the increasing complexity and volume of network traffic, traditional IDS face significant challenges in detecting intrusions effectively. The paper starts with a detailed explanation of the basic concepts of IDS, including their classification methods. It then delves into the methodologies adopted for both ML and DL approaches in NIDS. The authors review recent articles published between 2017 and the first quarter of 2020, focusing on the proposed methodologies,

strengths, weaknesses, evaluation metrics, and datasets used in these studies. Ahmad et al. then delve into machine learning-based methods, discussing a variety of classifiers including support vector machines (SVM), decision trees, k-nearest neighbors (k-NN), and ensemble methods like random forests[10]. The authors emphasize the enhanced detection accuracy and adaptability of these methods compared to traditional approaches.

Chandra et al. (2019) proposed a hybrid model for intrusion detection using the KDD Cup99 dataset. Their approach incorporated Filter-Based Attribute Selection to reduce feature dimensions, enhancing the model's efficiency by removing irrelevant and redundant features. For attack detection, they employed K-Means clustering to categorize the data into distinct groups, followed by Sequential Minimal Optimization (SMO) for the classification task[11]. This hybrid model demonstrated a significant improvement in accuracy compared to traditional methods, underscoring the effectiveness of combining different machine learning techniques for IDS.

The paper "A Survey of Intrusion Detection Systems Based on Ensemble and Hybrid Classifiers" by Abdulla Amin Aburomman and Mamun Bin Ibne Reaz reviews intrusion detection systems (IDSs) using machine learning techniques, particularly ensemble and hybrid classifiers. It discusses the prevalence of malicious network activities and the necessity for IDSs to counter unauthorized network usage. The study categorizes ensemble methods into homogeneous (same type of classifiers) and heterogeneous (different types of classifiers), noting their superior generalization ability compared to single classifiers[12]. Special emphasis is placed on methods utilizing voting techniques due to their simplicity and effectiveness.The survey covers various ensemble techniques like bagging, boosting (including AdaBoost), stacking, and mixtures of experts. It also examines hybrid methods, which combine feature selection or reduction with a classifier to enhance performance. The paper provides an in-depth analysis of these methods' application in IDSs, including a review of recent literature and machine learning techniques used in different ensemble approaches. Concluding remarks highlight the strengths and limitations of these methods, offering insights into future research direction.

2.2 Proposed System

The studies mentioned have certain limitations that are addressed in our system. The studies often rely on old datasets such as KDD-Cup99, NSL-KDD, or their proprietary datasets. These datasets do not reflect the latest attack patterns and techniques used by cyber adversaries. Consequently, systems developed and tested with these outdated datasets struggle to detect new and emerging threats. Our project will adopt the average accuracy as the primary performance metric. Average accuracy assigns equal weight to all class types, providing a more accurate and comprehensive assessment of the IDS's performance. This metric will ensure that the system's ability to detect all types of attacks, regardless of their frequency, is accurately represented. By addressing these limitations, our project will develop a more robust and reliable IDS using the Random Forest algorithm, leveraging an up-to-date dataset and appropriate performance metrics to enhance its effectiveness against contemporary cyber threats.

CHAPTER 3 : METHODOLOGY

3.1 Working Mechanism

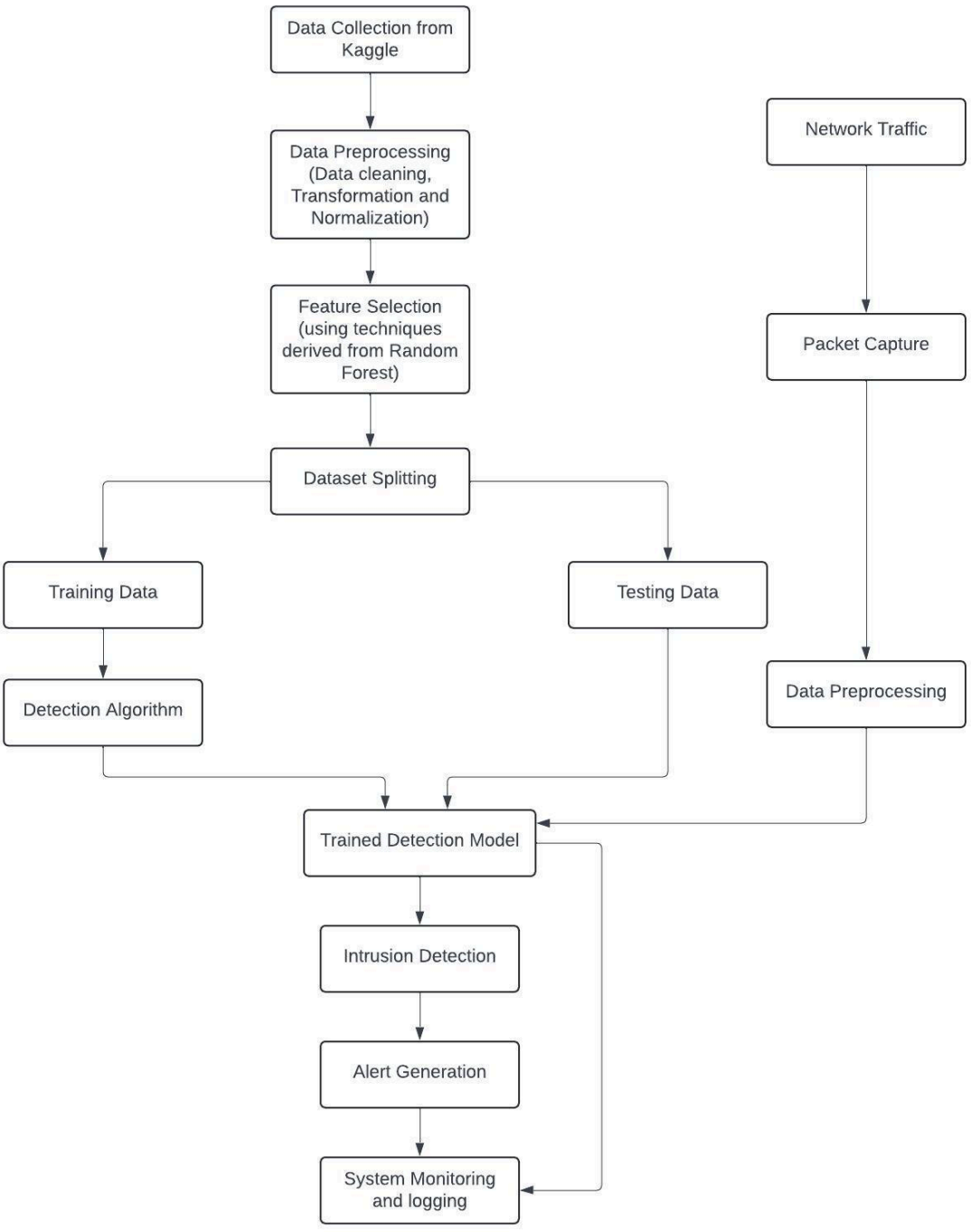

Figure 3.1 : Block diagram of NIDS

The block diagram of the Network Intrusion Detection System (NIDS) using the Random Forest algorithm and the CSE-CIC-IDS-2018 dataset is composed of several critical stages. Each stage plays a vital role in detecting intrusions and potential intrusions in network traffic. Here is a detailed explanation of each block, focusing on their implementation and contribution to the final result.

3.1.1 Network Traffic

This block represents the raw network traffic that flows through the network infrastructure. It includes all data packets transmitted across the network, which need to be monitored for any signs of intrusion or malicious activity. This step is fundamental as it serves as the source of data for the intrusion detection system.

3.1.2 Packet Capture

The packet capture process involves capturing the raw network traffic. Tools such as Wireshark or Tcpdump are deployed to intercept and log the network packets as they travel through the network. This process ensures that all relevant data is captured for subsequent analysis.

3.1.3 Data Collection

In this stage, the captured packets are aggregated to form the dataset. For the CSE-CIC-IDS-2018 dataset, data collection involves using specific network configurations and environments designed to simulate normal and malicious activities. This dataset includes a variety of attack scenarios, making it comprehensive for training and evaluating the NIDS.

3.1.4 Description of CSE-CIC-IDS-2018 dataset:

The CSE-CIC-IDS-2018 dataset is a comprehensive dataset specifically designed for network intrusion detection research. It was created by the Canadian Institute for Cybersecurity (CIC) in collaboration with the Communications Security Establishment (CSE). The dataset includes a wide range of network traffic data, simulating both normal behavior and various types of cyber-attacks, making it highly valuable for training and evaluating intrusion detection systems. The dataset encompasses several common and sophisticated attack types, which are categorized into the following:

i. **Brute Force Attacks:** These include SSH and FTP brute force attacks, where an attacker attempts to gain unauthorized access by systematically trying many passwords or passphrases.

ii. **Denial of Service (DoS) Attacks:** These attacks aim to make a machine or network resource unavailable to its intended users by overwhelming it with a flood of illegitimate requests. Examples include DoS Hulk, DoS GoldenEye, DoS Slowloris, and DoS Slowhttptest.

iii. **Distributed Denial of Service (DDoS) Attacks:** Similar to DoS attacks but launched from multiple sources simultaneously, making them more difficult to mitigate. The dataset includes DDoS attacks such as LOIC (Low Orbit Ion Cannon).

iv. **Infiltration Attacks:** These involve unauthorized access to internal network systems, where the attacker attempts to penetrate the network and extract or manipulate sensitive data.

v. **Botnet Attacks:** These attacks involve the use of a network of compromised computers (bots) to perform various malicious tasks, such as spamming or further propagation of malware.

vi. **Web Attacks:** These include various attacks targeting web applications, such as SQL injection, cross-site scripting (XSS), and file inclusion attacks.

vii. **Heartbleed Attacks:** These exploit a vulnerability in the OpenSSL cryptographic software library, allowing attackers to read the memory of the systems protected by vulnerable versions of OpenSSL.

The dataset is meticulously labeled, providing clear distinctions between normal traffic and various attack types. It includes detailed features extracted from the captured network traffic, such as flow duration, total packets, bytes, packet lengths, and various statistical features related to the network protocol behavior. This extensive feature set enables comprehensive analysis and the development of robust detection models.

By providing a diverse and realistic set of attack scenarios, the CSE-CIC-IDS-2018 dataset serves as an excellent resource for developing and testing network intrusion

detection systems, helping researchers and practitioners improve the security and resilience of network infrastructures.

3.1.5 Data Preprocessing

Once the raw data is collected, it undergoes preprocessing to clean and prepare it for analysis. This involves several steps:

i. Data Cleaning: Removing irrelevant or redundant data, such as duplicated packets or corrupted entries.

ii. Data Transformation: Converting raw packet data into a structured format suitable for analysis.

iii. Normalization: Ensuring that all data features are on a common scale to improve the performance of the detection algorithm.

3.1.6 Feature Selection

In this stage, the most relevant features for intrusion detection are selected. Feature selection is performed using techniques such as correlation analysis, mutual information, or feature importance scores derived from algorithms like Random Forests. By selecting the most informative features, the complexity of the model is reduced, and its performance is improved. This step ensures that the detection algorithm focuses on the most critical aspects of the network traffic.

3.1.7 Dataset Splitting

The processed and feature-selected dataset is then split into training and testing subsets using an 80-20 ratio. This means 80% of the data is used to train the model, and 20% is reserved for testing its performance. This division is crucial for evaluating the performance of the detection algorithm and ensuring its generalization capability.

3.1.8 Detection Algorithm

The core component of the NIDS is the detection algorithm, implemented using the Random Forest technique. The training data is fed into the Random Forest classifier, which constructs multiple decision trees. Each tree is trained on a subset of the data, and

the final model is an ensemble of these trees. The Random Forest algorithm is chosen for its robustness, ability to handle large datasets, and effectiveness in detecting patterns indicative of intrusions.

3.1.9 Trained Detection Model

After training, the Random Forest model is capable of classifying network traffic as either normal or anomalous. This trained model is then deployed within the network monitoring system. It continuously analyzes incoming network traffic in real-time, leveraging the patterns learned during training to detect potential intrusions.

3.1.9.1 Random Forest Algorithm

The random forest algorithm (RF) is an algorithm that is developed by combining the results of a large number of decision trees trained with different training clusters. It was developed by Breiman in 2001 as an algorithm that uses multiple classification techniques.

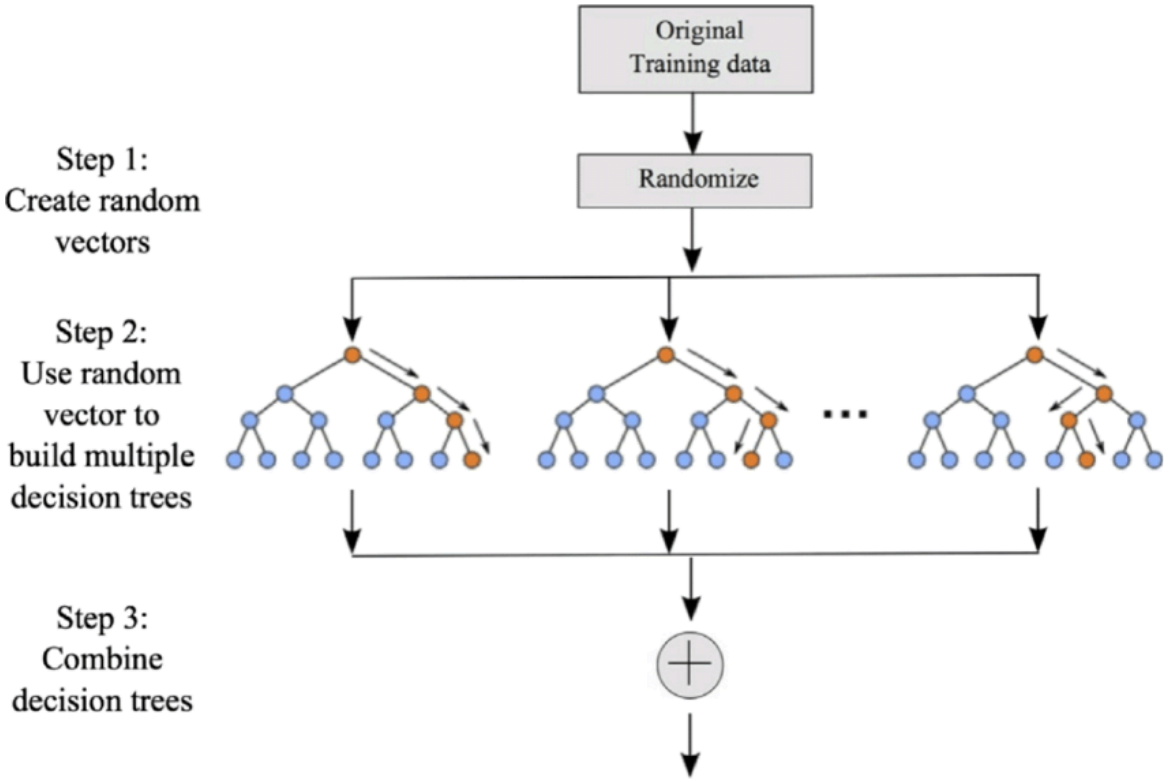

Figure 3.2 : Block diagram of the general operating principle of the RF algorithm.

[Source: Ü. Çavuşoğlu, "A new hybrid approach for intrusion detection using machine learning methods," 2019.]

In the random forest algorithm, different sub-training clusters are created. Preloading is performed in the creation of training clusters. For the expansion of the trees, a method in which the properties are selected at random is used. In the algorithm's operation, each node is divided into branches using the best value among the randomly selected values from each node. Derived trees are obtained by randomly selected variables. The Classification And Regression Trees (CART) algorithm is used for the tree development process from the obtained datasets. The sample to be classified is tagged according to each generated tree and the assigned classes are collected. The instance to be processed is included in the class to which it is assigned the most. Although pruning is found in the CART algorithm, pruning is not performed in the RF algorithm. The lack of pruning in the RF algorithm contributes to the RF algorithm being more successful than the other decision tree methods. Despite the use of multiple tree structures in the RF algorithm, the algorithm is quite fast, it can work with many tree structures, and its performance is better than other decision tree methods.

3.1.10 Intrusion Detection

The deployed model monitors the network traffic and detects intrusion based on the patterns it has learned. When a new data point (network packet) is processed, the model predicts whether it is normal or malicious. This real-time detection is critical for identifying suspicious activities as they occur.

3.1.11 Alert Generation

Upon detecting an intrusion, the system generates alerts. These alerts include details about the suspicious activity, such as the source and destination IP addresses, the type of attack detected, and the timestamp. The alert system can be integrated with network management consoles or security information and event management (SIEM) systems to notify administrators promptly.

3.1.12 System Monitoring and Logging

Continuous monitoring and logging of network traffic and system activities are essential for maintaining the effectiveness of the NIDS. Logs of detected intrusions, generated alerts, and other relevant events are maintained for further analysis and forensic

investigation. This helps in understanding attack patterns, improving the detection model, and complying with regulatory requirements.

In summary, the implementation of the Network Intrusion Detection System involves a systematic approach, starting from network traffic capture and data preprocessing, through feature selection and model training, to real-time attack detection and alert generation. Each block plays a crucial role in ensuring that the system can effectively identify and respond to network intrusions, thereby enhancing the security of the network infrastructure.

3.2 Overview of the System

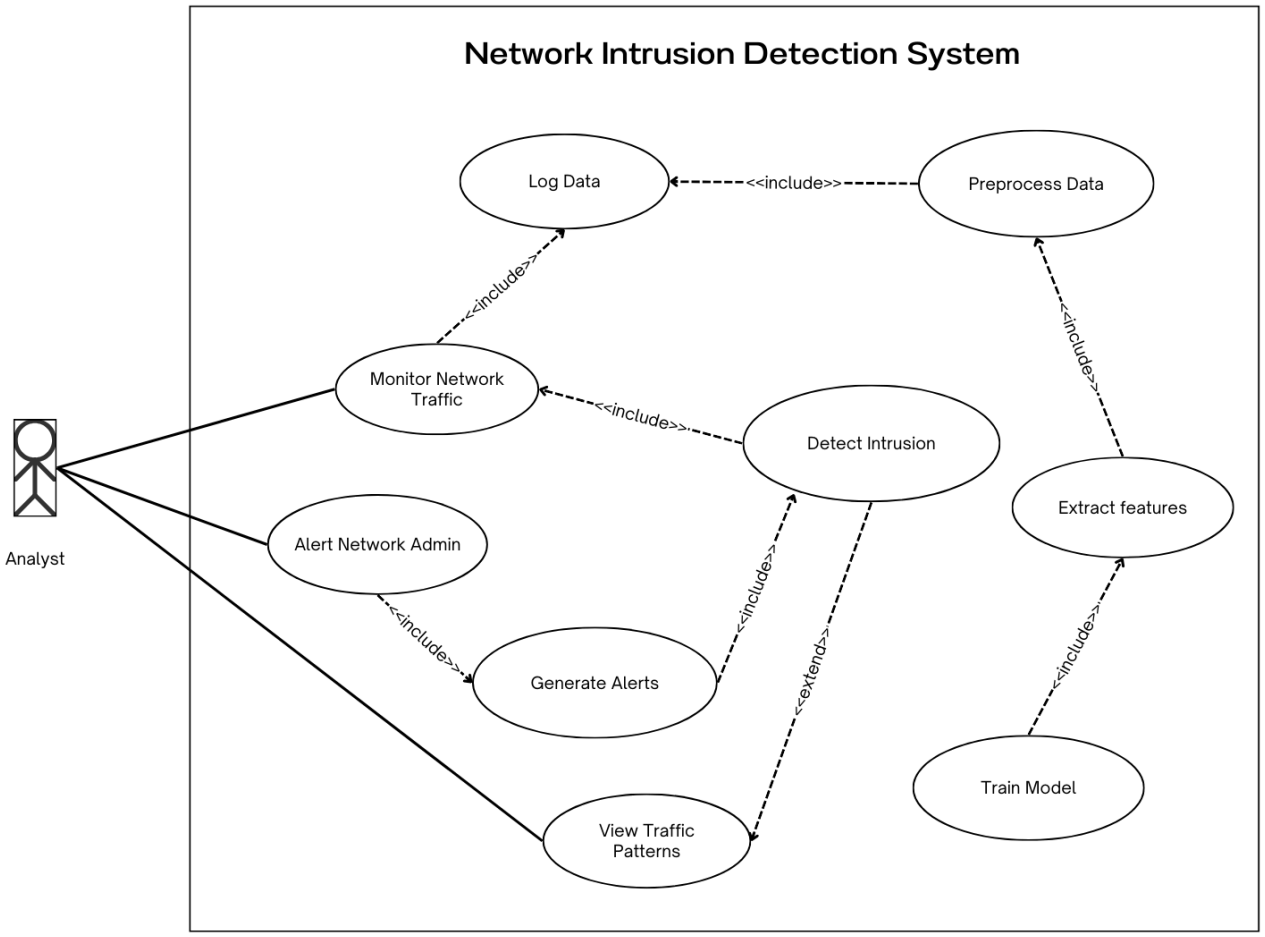

Figure 3.3 : Use Case Diagram

3.3 Software Development Model

The development of our Network Intrusion Detection System (NIDS) using the Random Forest algorithm and the CSE-CIC-IDS-2018 dataset will follow the incremental model, which involves building and delivering the system in small, manageable increments.

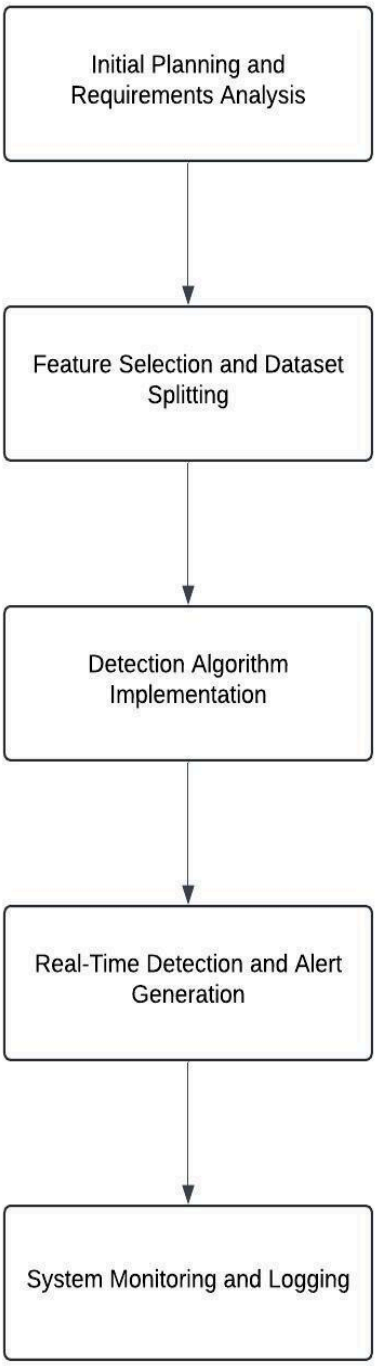

Figure 3.4 : Incremental Model

Initially, we will conduct thorough planning and requirements analysis to define the scope and objectives, followed by the implementation of the data collection and preprocessing infrastructure, where packet capture tools are deployed to collect and clean raw network traffic data.

The next increment focuses on feature selection and dataset splitting, ensuring that only the most relevant features are used and the data is divided into training (80%) and testing (20%) subsets. We then develop and train the Random Forest detection algorithm, tuning hyper parameters to optimize performance. The trained model is deployed for real-time detection and alert generation, enabling the system to analyze incoming traffic and promptly notify administrators of potential intrusions. Finally, we set up continuous monitoring and logging to track system performance and maintain detailed logs for further analysis and improvement. This iterative approach ensures flexibility, early issue detection, and the incorporation of user feedback, ultimately resulting in a robust and effective NIDS.

CHAPTER 4 : EPILOGUE

4.1 Expected Outcome

The primary objective of this project is to develop a robust Network Intrusion Detection System (NIDS) using the CIC-IDS 2018 dataset and a Random Forest classifier. The expected outcomes include the detection of various types of network intrusions such as DDoS attacks, brute force attacks, and botnets, leveraging the reliable and consistent results provided by the Random Forest algorithm. Additionally, the project aims to enable real-time monitoring of network traffic to promptly identify suspicious activities, thereby mitigating potential security threats effectively.

A comprehensive evaluation of the model's performance will be conducted using metrics such as accuracy, precision, recall, and F1-score to ensure effective distinction between benign and malicious traffic. This evaluation will help minimize false positives and false negatives. The project also focuses on designing an efficient system, optimizing data preprocessing and model training processes to ensure quick and reliable operation.

Furthermore, the project will provide valuable insights into network traffic patterns, contributing to further research and enhancement of network security measures. A simple webpage will be developed to demonstrate how the NIDS model works using the Random Forest algorithm with the CIC-IDS 2018 dataset. This webpage will simulate attacks and extract the corresponding data, fitting it into the selected features used to train the model. This demonstration will showcase the real-time application of the model in detecting intrusions.

REFERENCES

[1] A. Tarter, “Importance of cyber security,” in *Advanced sciences and technologies for security applications*, New York, 2017, pp. 213–230.

[2] J. Li, Y. Qu, F. Chao, H. P. H. Shum, E. S. L. Ho, and L. Yang, “Machine learning Algorithms for network intrusion detection,” in *Intelligent systems reference library*, 2018, pp. 151–179.

[3] J. P. Anderson, “Computer Security Threat Monitoring and Surveillance”. Fort Washington, PA: James P. Anderson Co., 1980.

[4] M. S. Hoque, Md. A. Mukit, and Md. A. N. Bikas, “An implementation of intrusion detection system using genetic algorithm,” *International Journal of Network Security and Its Applications/International Journal of Network Security and Applications*, vol. 4, no. 2, pp. 109–120, Mar. 2012.

[5] Michie, D.; Spiegelhalter, D.J.; Taylor, C. *Machine Learning, Neural and Statistical Classification*; Ellis Horwood Series in Artificial Intelligence: New York, NY, USA, 1994; Volume 13.

[6] R. A. Jamadar, “Network Intrusion Detection System using Machine learning,” *Indian Journal of Science and Technology*, vol. 11, no. 48, pp. 1–6, Dec. 2018.

[7] M. Schonlau and R. Y. Zou, “The random forest algorithm for statistical learning,” *the Stata Journal*, vol. 20, no. 1, pp. 3–29, Mar. 2020.

[8]R. Abdulhammed, M. Faezipour, A. Abuzneid, and A. Abumallouh, ‘‘Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic,’’ IEEE Sens. Lett., vol. 3, no. 1, pp. 1–4, Jan. 2019

[9] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, ‘‘An adaptive ensemble machine learning model for intrusion detection,’’ IEEE Access, vol. 7, pp. 82512–82521, 2019.

[10] Z. Ahmad, A. S. Khan, C. W. Shiang, J. Abdullah, and F. Ahmad, “Network intrusion detection system: A systematic study of machine learning and deep learning approaches,” *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, Oct. 2020.

[11] A. Chandra, S. K. Khatri, and R. Simon, ‘‘Filter-based attribute selection approach for intrusion detection using k-means clustering and sequential minimal optimization technique,’’ in Proc. Amity Int. Conf. Artif. Intell. (AICAI), Feb. 2019, pp. 740–745.

[12] A. A. Aburomman and M. B. I. Reaz, “A survey of intrusion detection systems based on ensemble and hybrid classifiers,” *Computers & Security*, vol. 65, pp. 135–152, Mar. 2017.