Advancing Network Security with Machine Learning: A Predictive Intrusion Detection System

By
Sumaya Akter
ID: 213-15-4450

Tawhid Ahmed
ID: 213-15-4433

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the Requirements for the **Degree of Bachelor of Science in Computer Science and Engineering**

Supervised by
Dr. Arif Mahmud
Associate Professor and Associate Head
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by
Fatema Tuj Johora
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

September 16, 2025

APPROVAL

This Project titled "Advancing Network Security with Machine Learning: A Predictive Intrusion Detection System" submitted by Sumaya Akter and Tawhid Ahmed to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **16-09-2025**.

BOARD OF EXAMINERS

Dr. Bimal Chandra Das (BCD)
Board Chairman
Professor and Dean (in-charge),
Department of CSE, FSIT
Daffodil International University

Most. Hasna Hena (HH)
Internal Examiner 1
Assistant Professor,
Department of CSE, FSIT
Daffodil International University

Mr. Mayen Uddin Mojumdar (MUM)
Internal Examiner 2
Assistant Professor, Department of CSE, FSIT
Daffodil International University

Nazibur Rahman
External Examiner
Technical Lead - Database Administrator,
Telenor - Grameen Phone Account.

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Dr. Arif Mahmud, Associate Professor and Associate Head**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

Dr. Arif Mahmud
Associate Professor and Associate Head
Assistant Professor, Department of CSE, FSIT
Daffodil International University

Co-Supervised by:

18.9.25

Fatema Tuj Johora
Assistant Professor, Department of CSE, FSIT
Daffodil International University

Submitted by:

Sumaya

Sumaya Akter
Student ID: 213-15-4450
Department of CSE
Daffodil International University

Tawhid

Tawhid Ahmed
Student ID: 213-15-4433
Department of CSE
Daffodil International University

ACKNOWLEDGEMENTS

This work would not have been possible without the support and contributions of many individuals over the past two semesters. We are deeply grateful to everyone who has assisted us in one way or another.

First, we express our heartfelt thanks and gratefulness to the almighty for His divine blessing making it possible for us to complete the **Final Year Design Project(FYDP)** successfully.

We are grateful and wish our profound indebtedness to **Dr. Arif Mahmud, Associate Professor and Associate Head**, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh. Deep knowledge and keen interest of our supervisor in the field of Ecommerce to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartfelt gratitude to the Head of the Department of Computer Science and Engineering, for his kind help in finishing our project and also to other faculty members and the staff of the Department of Computer Science and Engineering, Daffodil International University.

We would like to thank our entire course-mates at Daffodil International University, who took part in this discussion while completing the coursework.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

In this paper, we benchmark and compare many machine leaning models in the CIC-IDS2017 for cyber-attack detection. The models are compared with Logistic Regression, Decision Tree, Random Forest, XGBoost, and LightGBM. The goal of the work is to compare the performance of these models in terms of accuracy, recall and precision for distinguishing the malicious and benigh network traffic. The most significant key features for attack detection were selected through feature importance rankings and correlations between attack categories with Random Forest. Experiments results show that the aggregated learning models, especially XGBoost and LightGBM, are capable to achieve better performance including accuracy, false positive rate, on malicious traffic detection compared to other widely used ones, including Logistic Regression and Decision Tree. Besides, the paper has studied a statistical analysis using Wilcoxon rank-sum test, and confirmed that the models recalled with no difference. The findings emphasize the potential of such ensemble techniques when it comes to online intrusion detection of cyber-attacks and factors which contribute in improving intrusion detection system

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

In this age, where technology is taking over, the requirement of a strong cyber protection is the most imperative. With more network-connected devices (and more sophisticated threats) than ever before, traditional security tactics are quickly becoming outdated, struggling to keep pace and prevent the next big threat. IDSs have been used extensively to protect sensitive data and to preserve the integrity of an information system. However, classical IDSs are often signature based and are only effective against known attacks, and as a result they have turned out to be ineffective against new and sophisticated attack tactics. This gap between security and detection has resulted in the emergence of machine learning (ML) within network defenses, the reason being that ML models can capture data patterns and generalize well among new or unseen threats.

This thesis is specifically about the design and implementation of a predictive IDS that harnesses machine learning to identify network intrusions continuously and in real-time. As the CIC-IDS2017 dataset has a wide array of labeled network traffic data for normal and attacks scenarios, this paper aims to improve the detection performance of IDS mechanisms. The goal is to develop a system that not just flags known threats but even anticipates and identifies the never-before-seen attacks by identifying the new structure in the network traffic, he said.

By combining several machine learning algorithms such as Logistic Regression, Support Vector Machine (SVM), Random Forest, Decision Tree, and K-Nearest Neighbors (KNN), we seek to compare different models and to pinpoint the best method for identifying malicious network activities. In addition, in this work the problem of class imbalance, which is often en- countered in network intrusion datasets, is taken into account and managed by using techniques such as Synthetic Minority Over-sampling Tech- nique (SMOTE) to guarantee that the models are trained to adequately discriminate benign and hostile traffic.

Generally, our goal is to contribute to the development of more accurate and more reactive intrusion detection systems, which may defend networks against evolving types of cyber-attacks, in a prevention manner. This study wishes to demonstrate practical use of advanced computational techniques in network security, via creating a predictive ML Integrated IDS.

1.2 Motivation

The expansion and sophistication of cyber-attacks presents a challenge for all businesses and private individuals, so it's not surprising that companies everywhere are ever more urgently looking for more advanced network security solutions." Cyber adversaries constantly come up with new techniques to bypass conventional defense solutions, and traditional signature-based detection used by intrusion detection systems (IDS) is not effective in detecting new or targeted threats. This means that organizations today are left unprotected against new, developing threats, including attacks against sensitive data, services outages and significant financial and reputational damage.

This thesis is inspired by this deficiency, and attempts to explore the potential of machine learning in construction of effective and adaptive IDS systems. Unlike traditional solutions, machine learning based IDS may be trained continuously to reflect network traffic patterns that ultimately recognize new techniques to attack systems with less human intervention or update. Given this capable learning way, machine learning is an ideal type for improving the intrusion detection with the aim of improving its performance.

Then, using machine learning in network security can largely mitigate the number of false positives a prevalent problem for traditional IDS. A system for the detection of known and the prediction and identification of unknown cyber threats is being developed by this work, with a further objective to design a proactive, accurate defense system. Such determination and prediction about malicious activities in real-time will allow for prompt and quick reactions to threats, therefore reducing the negative impact of cyber-attacks.

Additionally, the imbalanced characteristic of network security datasets, where benign traffic is usually much more than the attack traffic, makes the process of training an efficient model for machine learning a problem. The same is also a motivation for this thesis to investigate and resolve this weakness using methods such as Synthetic Minority Over-sampling Technique (SMOTE) that the models are learned in a way that benign and malicious traffic are equally accurately recognised.

More interestingly, such research can be one of the stepping stones towards the development of an IDS that is not only more accurate and efficient than the current ones, but also can adapt to changing cyber threat landscapes. This thesis aims to use the potency of the machine learning to make the organization more capable of protecting themselves from the many more and more complicated level of assault presented to them as we live in these days world of connectedness.

1.3 Objectives

The objectives for this project are:

- Designing IDS using Machine Learning: Implementing an IDS through the use of machine learning techniques in the context of detecting network traffic either as benign or as malicious.
- Investigate the possibility of designing and testing an IDS which uses machine learning to determine if network traffic is good or bad. ·
- Balancing Class for Ancients Network Traffic To utilize the (Smoke) to address the imbalance between positive and negative samples for the model to be more effective for benign and attack samples. ·
- Comparison of Multiple Machine Learning Models: To perform comparison of various machine learning models (Logistic Regression, SVM, Random Forest, Decision Tree, K-NN) in detecting network intrusions. ·
- Improving Model Accuracy with Preprocessing Techniques: It is about using the data preprocessing techniques (e.g., normalization, standarization and dimension reduction such as PCA) to help the I DS performance (accuracy and efficiency).
- Optimize the IDS Models: To tune the models through things like cross-validation and hyperparameter tuning, to avoid overfitting and improve performance.
- Assess Performance of the Model The standard metrics can be used to assess the performance of the model: To measure the performance of the IDS based on the evaluation metrics such as accuracy, recall, precision, F1-score, and confusion matrix.
- Contribute to Network Security Research: To offer some perspectives of how machine learning is used to complement intrusion detection systems and network security defenses in defending against emerging cyber-threats.

1.4 Methodology

In the thesis an intrusion detection system (IDS) based on the concept of machine learning in order to be able to support network security is developed and tested. The size of the dataset in this research likely to mainly manage the dataset of CIC-IDS2017 that has both attack of many different types and normal network traffic. This dataset will then be used for the study above to investigate the potential of machine learning algorithms for predicting and identifying malicious behavior in real-time network contexts.

The study will focus on the preprocessing of data, selection of a suitable machine learning model and system performance analysis. A major part of the scope will consist of tackling the imbalanced classes in the dataset, a challenge commonly met in network traffic data. The thesis will also investigate the possibility to balance the training dataset using the Synthetic Minority Over-sampling Technique (SMOTE) in order to make the model capable to effectively distinguish benign from malicious traffic.

A comparative study of different machine learning models like Logistic Regression, Support Vector Machine (SVM), Random Forest, Decision Tree, and K-Nearest Neighbors (KNN) will be pursued in this thesis. The performance of the models will be assessed by the resources the models use to identify network intrusions and its predictability ability respectively. Pre-processing of data such as normalisation, standardisation, dimensionality reduction (PCA) will also be carried out to improve the models' performance.

As its name suggests, the main objective of the work is to design and assess the IDS with machine learning, although some ideas to improve the network security from the predictive and adaptive IDS sides will also be something that comes out from the research. The system proposed in this thesis will be tried and refined, but it will still be a prototype, meaning it is aimed for demonstration and evaluation, rather than run in large scale in production environments.

For this purpose, the analysis is restricted to the CIC-IDS2017 dataset and not applied to any other available datasets nor to any real-life use beyond the test of the prototype.

1.5 Thesis Outcome

The anticipated contribution of this thesis will be a scalable machine learning (ML)-centric Intrusion Detection System (IDS) capable to accurately detecting both known and previously unseen cyber-attacks from real-time network traffic. Using the CIC-IDS2017 an IDS is to be developed that is highly effective at classifying network traffic as well as addressing some of the challenges that are present with many classifiers including class imbalance (which is known to skew the results of traditional detection systems).

Comparatively testing different machine learning models (Log for Inclusion and SVM, Random Forest, Decision tree, KNN for exclusion), the thesis plans to see which algorithm works better for IDS (network security). Further, the deterministic preprocessing procedures (e.g., feature scaling and principal component analysis (PCA) for dimensionality reduction), will result optimization models in the set that all things considered, are most predictive and computationally efficient.

This thesis will also benefit the networking security community by presenting the utilisation of predictive approach to intrusion detection as opposed to the reactive methodologies before. This can be used to guide future researches in self-adaptive IDS and next-gen threat detection. Ultimately, the output of the research will be a working prototype of the IDS for real-time detection that will form a basis for continuing development work and possibly even for deployment in the real world.

1.6 Organization of the Report

This report is organized in multiple chapters that smoothly describe the research, results, and outputs of this thesis. The article is organized as follows:

Chapter 1: Introduction

Chapter 1 provides the motivation, objectives, scope and contributions of the research. It gives a background to the machine learning approach to IDS and addresses the significance in addressing contemporary problems related to network security.

Chapter 2: Literature Review

This chapter provides a literature review of aforementioned literature in the field of intrusion detection system and machine learning in network security. It presents classical IDS techniques, challenges in the domain like class imbalance and some related work on the use of machine learning for intrusion detection. This chapter also identifies the limitations of the existing literature that this thesis seeks to remedy.

Chapter 3: Data Set and Preprocessing

This section describes the digital EEG data and preprocessing that were implemented for the proposed method.

Chapter 4: Methodology

In this chapter the research method is described. It also illustrates machine learning algorithms that are applied to intrusion detection, such as Logistic Regression, SVM, Random Forest, Decision Tree and KNN. In addition, the chapter describes the model validation activity, whereby the performance of the models is evaluated based on performance measures.

Chapter 5: Results and Discussion

In this chapter we illustrate the performances of our machine learning models, trained on and tested against the CIC-IDS2017 datasets. It explains the model evaluation from accuracy, precision, recall, and F1-score and also confusion matrix for the algorithms. The impact of data pre-processing, class imbalance handling and optimization strategies on model performance are also addressed in chapter four.

Chapter 6: Conclusion and Future Work

Chapter concludes the research, by summarizing findings, contributions to Network Security and looking up topics for future research. It also reviews some possible practical applications for the proposed IDS and suggests directions for further improving IDS.

Chapter 2

Background

2.1 Introduction

As the digital sphere becomes increasingly inter-connected today, institutional and individual entities are placed at risk from the growing number of cyber-based threats aimed at compromising the confidentiality, integrity, and availability of networked systems. Conventional security appliances, such as firewalls and signature-based IDS, have been the initial defenses for many years. But these existing systems are very rule-based or based on known attack pattern, so that they are only effective for the known threats. In recent years with the proliferation of attack mechanisms - zero-day attack, DDoS (distributed denial-of-service) attacks, botnets, and APT (advanced persistent threats), signature-based IDS can no longer address all the emerging threats. This limitation has resulted in the need to investigate intrusion detection techniques that are more dynamic and intelligent.

Machine learning (ML) has become an appealing paradigm in this setting as it can learn patterns from data and generalize to new, unseen situations. Unlike signature-based approaches, since ML-based IDS is able to process network traffic on a large scale, it can capture abnormal or malicious behaviors without having prior knowledge about particular attack signatures. Using techniques such as supervised classification, data balancing, including the use of the Synthetic Minority Over-sampling Technique (SMOTE), and dimensionality reduction, including the Principal Component Analysis (PCA) make it possible for such systems to improve detection rates, especially for rare or evolving attacks. There are datasets, such as CIC-IDS2017, which include normal and multiple attack traffic that have been widely used as benchmark data for ML-based IDS training and evaluation, as the data has been generated STLC, meaning that these datasets reflect natural network conditions well.

Furthermore, the demand for real-time, proactive and scalable solutions has led to a growing interest in predictive IDS. A predictive IDS is not a reactive one; its core aim is to predict, in accordance with time series models, the dangerous attacks and anomalies, by triggering suspicious traffic before full attacks are launched.

2.2 Literature Review

2.2.1 Similar Applications

Intrusion Detection Systems (IDS) were best security tools from years: A History on IDS. The first IDS systems were mainly based on signatures, using fixed patterns to match known attacks. While often proven efficient at finding previously reported intrusions, such systems have failed to discover new or day-0 attacks [1]. Anomaly-based IDSs were developed against this shortcoming, as they identify those deviations from normal behaviour of traffic. Although such systems increase the sensitivity for identifying new threats, the higher false alarm rates associated with them are still one of major obstacles to practical use [2]. This has led them to work on a combination of the two and provide better detection [9].

Machine learning (ML) introduced a lot of new approaches in the research for IDS systems by allowing systems to learn from a large collection of traffic data. Conventional machine learning models like SVM, RF, LR, KNN are used as the classifiers to classify the network traffic into normal or attack traffic [3], [4]. Such methodologies presented much better flexibility than the based on the signature-based systems. Though their results are generally dependent on the quality of feature engineering and data pre-processing. For instance, SVM has been demonstrated with strongness in the face of high-dimensional data, but it is at the cost of long-time model tuning [10]. Ensemble methods including Random Forest and Gradient Boosting have also been widely used since these methods can decrease variance and enhance classification performance on IDS systems [11].

2.2.2 Related Research

The class imbalance problem is one of the most challenging issues in IDS studies. Traffic datasets, like CIC-IDS2017, have a large inequality between benign and malignant records. This type of imbalance skews classifiers to the majority classes, making them less sensitive to rare yet crucial types of attacks [5]. Researchers have tackled this problem by using oversampling techniques such as SMOTE [5] and its variants (Borderline-SMOTE, ADASYN), which artificially create minority class compartments to balance the dataset. Such tricks have been proved to be useful to recall and F1-score for minority attacks [12]. However, it is possible for noise to be introduced during the generation of the synthetic data, and therefore limited generalization performance with real world data.

It is a potential future line of investigation to perform feature selection and dimensionality reduction. In ML-based IDS, computational cost and risk of overfitting of ML increase with the dimension of datasets. To sort out this issue, approaches such as PCA have been used to remove redundancy and concentrate on the most relevant ones [6]. Other sophisticated techniques like information gain, recursive feature elimination and mutual information have also been investigated to enhance the efficiency of IDS [13]. These methods decrease

system scalability of real-time intrusion detection over large network systems as well as reducing training times.

Intrusion Detection is one of the main fields which was revolutionized by deep learning (DL) in recent years. CNNs have been demonstrated to automatically extract hierarchical features from raw traffic data, and achieved much higher performance compared to traditional classifiers for detecting complex attack patterns [7]. Time dependence in the sequential traffic flows has been considered by using technology such as Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) networks, to enhance the detection procedures to the evolving attacks [8]. Moreover, hybrid DL based networks employing CNN and LSTM have shown promising performance for both volumetric and application-layer attack detection [14].

The literature also highlights a growing trend towards hybrid and ensemble IDSs that combine ML and DL techniques to enhance the detection precision. For instance, the ensemble models including Random Forest and deep learning can achieve much better robustness against various attack variants [15]. Lightweight models-driven IDS are also actively studied as one solution to be integrated with computing and IoT new paradigms (like edge computing [16]) where limited resources are available and therefore lightweight, yet accurate, models are needed. These works reflect the ongoing development of IDSs (from static rule-based solutions to adaptive and predictive) that allow the dynamic context to be processed and analysed to achieve real-time detection.

2.3 Gap Analysis

The studies reveal remarkable advancements in the area of intrusion detection but very much leave gaps. Signature based and anomaly-based IDS, being currently be used as the base lines in the network security's research, but they are suffering from some issues. Because of the above-mentioned limitations, signature-based approaches fail to achieve new threats or zero-day attack detections, anomaly-based ones tend to have the issue of volume false positive rates and not easy to be widely adopted in large Network [1], [2]. While ML has been increasingly successful for these tasks, many studies continue to rely heavily on classical methods that rely on manual feature engineering as well as thoughtful preprocessing. This reliance significantly hinders scalability and fails to adapt against various attack strategies [3],[10].

Another major lacuna is the problem of class imbalance. While oversampling methods such as SMOTE, etc. have been widely employed, existing schemes are mainly tested on balanced subsets or synthetic data sets, but not on the class imbalance setting on real-world scenario [5], [12]. This reduces the practical usability of the ML-based IDS, where the positive or malicious traffic is outnumbered by the benign traffic. (PCA) [6] used for reducing dimensions and enhancing the efficiency however, such methods are likely to throw away the important alarms and consequently, the efficiency of detection when facing

rare or advanced attacks is likely to decrease.

Deep learning (DL) methods in the equivalent of CNN and LSTM have shown effectiveness on learning patterns and temporal relations naturally [7], [8]. Nevertheless, in most cases, DL oriented studies concentrate on detection accuracy, while computational cost, scalability, and interpretability still remain open. For instance, DL models demanding significant resources may not be applicable in real-time or resource-restricted scenarios [16]. Hybrid models integrating ML and DL have also been investigated [15], but there has been little research in designing a predictive, lightweight IDS, weight in terms of both detection accuracy, and scalability and efficiency.

Lastly, while there are many research works that put forth models based on benchmark datasets such as CIC-IDS2017, there are also few works investigating the integration of predictive intrusion detection to a real-world operational environment. Recent studies mostly conduct offline style of IDS validations, producing the void from a competitive experiment of validation to a practical application. The dearth of practical use demands IDS systems that can efficiently process live traffic, be responsive to new threats, and provide proactive defense without overloading system resources.

Consequently, the major research gaps are: (i) reliance on classical supervised ML methods which tend to be less adaptive, (ii) lack of treating class imbalance problems in real-world datasets, (iii) less scalable and interpretable deep learning models, and ultimately, (iv) the absence of deployment-centric IDS that perform well in a real-time domain. Filling the gap above motivates this thesis to implement a predictive ML-based IDS to enhance detection accuracy and minimize false positives and operate efficiently under various network conditions.

Table 2.3.1: Comparison of Intrusion Detection Approaches

Approach	Description	Advantages	Limitations / Gaps	References
Signature-based IDS	Detects intrusions using predefined attack patterns or signatures.	Accurate for known attacks; simple to implement.	Cannot detect zero-day/new attacks; requires frequent updates.	[1], [9]
Anomaly-based IDS	Identifies deviations from normal traffic behavior.	Capable of detecting novel attacks; more adaptive.	High false positive rate; challenging to define "normal" traffic.	[2], [10]
Traditional ML-based IDS (SVM, RF, KNN, etc.)	Uses supervised learning algorithms for classification.	Better adaptability; improved detection accuracy vs. traditional IDS.	Requires manual feature engineering; performance drops with imbalanced data.	[3], [4], [11]
SMOTE & Other Oversampling Methods	Synthetic generation of minority-class samples to balance datasets.	Improves detection of rare attacks; boosts recall & F1-score.	May introduce noise; can lead to overfitting and reduced generalization.	[5], [12]
Feature Selection / PCA	Reduces dataset dimensionality by selecting most relevant features.	Enhances efficiency and scalability; prevents overfitting.	Risk of losing critical features, lowering detection accuracy.	[6], [13]
Deep Learning (CNN, LSTM, Hybrid)	Learns hierarchical and temporal patterns from traffic data.	High accuracy; automatically extracts features; effective for complex attacks.	Computationally expensive; prone to overfitting; less interpretable.	[7], [8], [14]
Hybrid ML + DL Approaches	Combines classical ML with deep learning models.	Improves robustness; balances strengths of ML & DL.	Still resource-intensive; limited real-time deployment studies.	[15], [16]

2.4 Summary

The studies reveal remarkable advancements in the area of intrusion detection but very much leave Theoretical implications The literature review within this dissertation explains the development for Intrusion Detection Systems (IDS) which also move from the archaic system of signature-based detection and lead into the higher level stage if ML (machine learning) is in place. At the beginning, IDS systems used hard-coded attack patterns so they were very effective in detecting known threats, but not against new ones or "zero-day attacks". This led to the development of the anomaly-based IDS that could capture when the network diverges from normal behavior, allowing the unusual threats to be more flexible identified. But such systems tended to produce a lot of false positives, so they were never very practical Implementing the rules proved easier said than done in many cases. The accuracy and adaptability of the IDS are also improved by machine learning, and such models as SVM, RF, LR, and KNN have been adopted to achieve better detection. These ML models have been shown to be successful in practice, especially with the aid of techniques as feature selection or preprocessing. However, there are still challenges, particularly in case of the unbalanced datasets where the benign traffic is extremely larger than the attack traffic. Approaches such as SMOTE (Synthetic Minority Over-sampling Technique) have been suggested to address the Imbalance problem, which however has the risk of introducing noise leading to model degradation.

In addition, recent trend was presented for automatic feature extraction and complicated attack pattern identification using deep learning (DL) models i.e., CNN and RNN in the review as well. These DL models have produced excellent results, especially with respect to advanced attack categories; however, this advantage is accompanied by the problem of increased computation costs… Hybrid models that fuse classical ML together with DL techniques (Pinior et al.,2018) have also been studies as it can bring both best of the two worlds. Model such as (Shahri et al., 2016) have been investigated but are still computationally intensive and are not yet found in real time.

Finally, it presents main limitations of today's research, like the tendency to focus solely on classic supervised learning methods, difficulties moving from academia to real-world traffic, the difficulty in dealing with class-imbalanced traffic in real traffic traces and the lack of real-time, operational responsible IDS systems. Such holes inspired the present thesis, which is geared towards building a more flexible, efficient and scalable IDS based on machine learning

Chapter 3

Methodology

3.1 Methodology

3.1.1 Overview

In this work, several machine learning algorithms are used for construction and evaluation the IDS. The Decision Tree model is one of the simplest methods that generates a tree structure of rules to decide whether an instance of network traffic is benign or malicious. The model's interpretability and simple implementation make it a valuable baseline that does not perform well on complex datasets for it has a tendency to overfit. To overcome this issue, the Random Forest which is an ensemble of decision trees that combine the predictions made by each tree is used. Random Forest achieves low variance and good generalization by randomness in both feature selection and bootstrapping and hence is powerful for noisy, imbalanced and noisy class data.

Based on the advantages of ensemble learning, this work also applies XGBoost (Extreme Gradient Boosting) and LightGBM (Light Gradient Boosting Machine), two powerful boosting algorithms known for their good computational efficiency and prediction accuracy. 286 64 XGBoost works on the principle of iteratively boosting weak learners with gradient boosting and applies a regularization technique to reduce overfitting, which is one of the reasons why it is very effective in structured 65 data tasks such as IDS. LightGBM is designed for large-scale and high-dimensional data, adopts a leaf-wise (best-first) tree growth strategy, and offers histogram-based learning, which help to train faster with comparable accuracy. Thanks to its capability to process large traffic data effectively, it is appropriate to be employed in practical IDS cases.

Finally, we consider Logistic Regression as classical and strong baseline for binary classification. Although it is linear, the logistic regression is attractive in the ID context because it is interpretable, computationally efficient, and possible to separate a benign attack traffic from malicious one when the decision boundary is simple enough. Fitted with each other these models are able to embodies both traditional classifiers and state-of-the-art ensemble methods, therefore we can perform a deep learning on which category of machine learning strategies would be more effective for network intrusion detection.

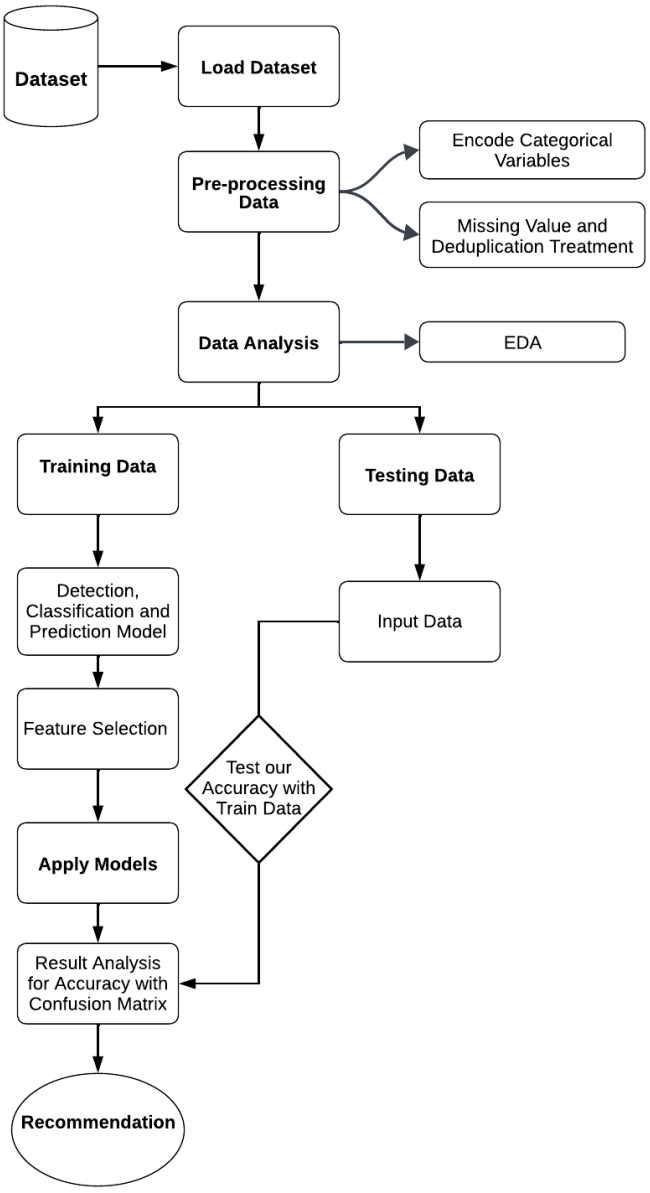

Figure-3.1.1: Overall Flowchart

3.1.2 Proposed Methodology

There are several evaluation measures to ascertain the effectiveness of the intrusion detection models, each of them to measure different aspects of classification quality. The most commonly used metric is accuracy, which is the fraction of the correct classifications out of all the predictions. It is however not a very good practice to simply rely on overall performance because it is misleading in the case of highly imbalanced dataset such as intrusion detection where the number of benign connections is far greater than those of malicious connections. That's why there are more metrics to help balance this out. Precision is the fraction of attacks correctly identified out of all instances that are reported as malicious and reflects to what degree the system can be trusted in issuing overall reports.

On the other hand, recall measures the ratio of detected attacks out of all actual attacks, and thus reflects how well the model is performing regarding the minimization of false negatives. Because precision and recall tend to trade off against each other, one can use the F1-score (harmonic mean of precision and recall) to represent a single balanced metric.

One other useful visualization is the confusion matrix, it provides a detailed breakdown of the true positives, true negatives, false positives and false negatives. This provides a better insight into the behavior of the model since it is possible to see where the model was okay and when it was failing, especially for benign and attack traffic. By employing more than one evaluation metric instead of accuracy alone, our approach guarantees that the generated IDS models are also evaluated for their capability to detect rare and possibly we harmful intrusions which is necessary in practical cybersecurity applications.

3.1.3 Functional and Nonfunctional Requirements

Functional requirements for the suggested IDS in the thesis represent its basic functionality to detect and to classify network traffic as normal or anomaly. The system has to process high volumes of network data in real time, and needs to run machine learning models to determine what is traffic and if this traffic is a normal or an anomaly/attack. It needs to be agnostic with different ml algorithms, such as: Logistic Regression (regressor)/SVM/RandomForest models/XGBoost models/LightGBM models and allow comparison between them by accuracy/recall/precision/F1-score. Moreover, the system must deal with the class imbalance problem in the dataset by means of using oversampling (e.g., SMOTE) to make sure the occurrences of normal and attack traffic are both well represented in the training data. The IDS must: be able to assess the performance of the models using hyperparameter tuning and cross-validation to maximize performance while preventing overfitting.

On the practical aspect, the system needs to run on efficient and accurate nonfunctioning, hence low false positives and false negatives, that is a critical point for applicability. It should also be scalable, that means it also have ability to handle huge files as in CIC-IDS2017 dataset it has lakhs and millions of records and flexible and it allow update for new cyber-attacks modifications. We expect the system to accommodate heterogeneity of network, to give a fast response, identically to our system, reaction time to identify and to neutralize real time threats. The IDS also should be user-friendly, i.e., have a user-friendly interface for monitoring and analyzing detected intrusions and is compatible with existing network security equipment. Finally, the system must adhere to applicable security standards and ethical requirements, and references were provided on how it must act to protect the privacy and health data.

3.2 Detailed Methodology and Design

In order to improve the detection efficiency and robustness of the detection model, we utilize two optimization methods, cross-validation and hyperparameter tuning. Cross-validation is applied to avoid overfitting of the models to training data rather than learning the patterns. In this method, we divide the dataset to K parts, and use part k as validation set and training set are all other parts. This procedure is performed a number of repetitions and the mean is taken which allows a better estimate of the model performance. Cross-validation can not only minimize the issue of overfitting but the selected model can also have the greater confidence to obtain a stable performance in the realistic situation.

In addition, the hyperparameter tuning is performed to find the best model parameters whose interaction shapes learning and performance. For example, hyper-parameters such as number of trees, maximum depth and learning rate in tree-based models such as Random Forest, XGBoost, LightGBM substantially influence predictive performance as well as the time it takes to train models. To search hyper-parameters, we use grid search and random search methods to explore a variety of hyper-parameter combinations. In some cases, one may also find it beneficial to employ more sophisticated optimization techniques (e.g., Bayesian optimization) in order to allow for a more effective exploration of the search space. Through cross-validation and meticulous fine-tuning of the hyperparameters, the models are tuned to give the best performance, in terms of precision, recall, and punctuality of detecting both common and uncommon network intrusion types.

3.2.1 Dataset

The dataset, CIC-IDS2017, used is a benchmark dataset, highly known in network security research community for evaluating IDSs [17]. It was compiled by the Canadian Institute for Cybersecurity and comprises more than 2.8 million records of network traffic from five days. The traffic that comprises the dataset spans from benign network traffic to known attack classes, i.e., Denial of Service (DoS), Distributed Denial of Service (DDoS), brute force, botnet, web and Heartbleed attack. Each network flow is described by a set of features—packet length, flow duration, port numbers, flags, protocol types, and so forth—that allow us to show a detailed representation of network activity.

One of the prominent problems of this dataset is that it is biased, benign traffic is much larger than malicious traffic. Such an imbalance may lead machine learning models to predict normal traffic and, as a result, diminish their performance in identifying uncommon but important attacks. To overcome this, Synthetic Minority Oversampling Technique (SMOTE) [18] based preprocessing such as example synthesis are performed to have a balanced representation for the minority classes during model training. Moreover, redundant or absent values are processed in a robust manner and dimension reduction techniques such as Principal Component Analysis (PCA) are performed to reduce the

feature space while retaining meaningful information. In this regard, we believe that the variety, size, and complexity of the CIC-IDS2017 dataset qualifies it to be employed as a suitable basis for designing and testing predictive IDS, closely simulating network environments that resemble real-life and whose ability to differentiate between attack flows and normal ones may be simultaneously complex and fundamental.

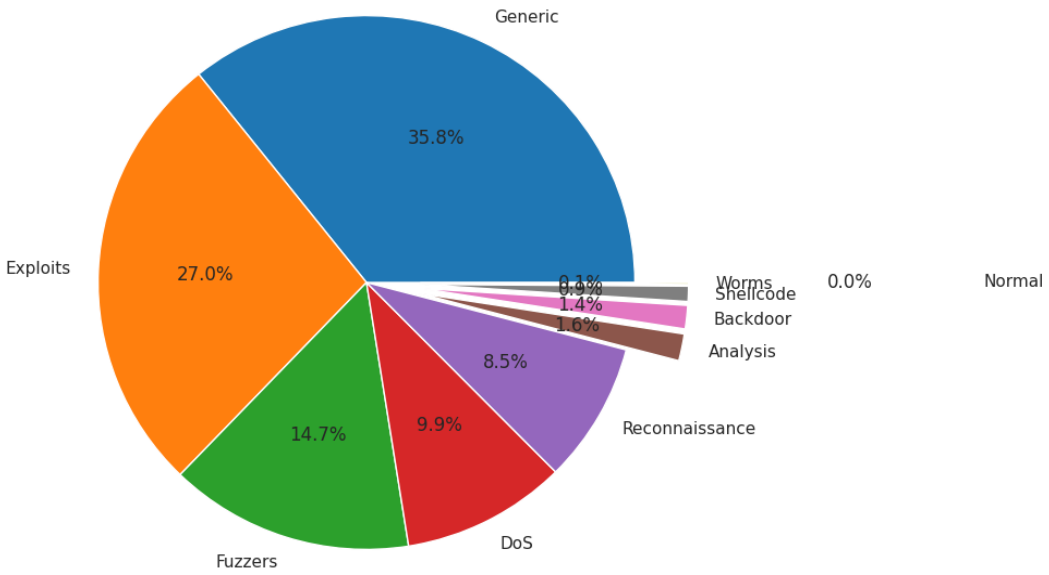

Figure 3.2.1. Attacks Categories

3.2.2 Data Preprocessing

Preprocessing of the data is an important step in developing an efficient intrusion detection system, because the network traffic data that we collect is noisy, redundant and imbalanced which has adverse effect on the model's performance. This study involved several systematic procedures to ensure the CIC-IDS2017 dataset preprocess for machine learning use. At first, data cleaning was carried as it is accepted missing values and duplicate ledgers were removed from the dataset so that the model learned without having bias and the training was sound. If the corresponding flow record is missing, we replaced it by a value obtained by one of the statistical imputation methods, or simply removed it if the flow record was missing: duplicate flows were removed to avoid such a big median score.

Data normalisation and standardisation were then used to rescale numerical features on a similar scale. Normalization Since fields in a flow record can have multiple orders of magnitudes (e.g., packet size vs. flow duration), we normalize the dataset so that one feature does not end up dominating the learning process (especially for feature scaling sensitive algorithms like Logistic Regression and Support Vector Machines). Normalization was applied where necessary to scale all feature values into a distribution with zero mean and unit variance to aid convergence of model training.

The Synthetic Minority Oversampling Technique (SMOTE) was used to solve the class imbalance. Create synthetic samples for the minority classes (the different types of attacks) by interpolating between pairs of existing minority class samples thereby balancing the data set and enabling sample efficient classification of rare, yet devastating, intrusions such as Heartbleed or botnet traffic. This step is crucial as such an imbalance can yield high precision, low recall, and thus obscure the fact that the model is not actually identifying attacks.

PCA was also used for dimension reduction in order to make the feature space less complex. PCA maps the data to a lower-dimensional space on which the most informative parameters are framed, in which redundancy on correlated parameters is mitigated, computational efficiency is enhanced, and overfitting is avoided, so that generalization can be achieved. Moreover, feature selection methods were applied to select the most informative attributes to be used for classifiers, which allows to the models to rely on features with higher predictive capacity.

Lastly the dataset was divided into train, validate and testing sets for a more robust model evaluation. The fitting of models was based on the training set, and hyperparameters were tuned and overfitting was avoided based on the validation set, and generalization performance was tested by the test set. Collectively these preprocessing steps constructed the clean, balanced, and performance-optimized dataset which could form a firm basis for training machine learning and ensemble models, and thus reliable and scalable intrusion detection results.

3.3 Project Plan

The project plan for this thesis presents a systematic approach for the design and implementation of the proposed machine learning based IDS. The proposal starts with an extensive review of the state of the art of both traditional and recent IDS techniques to be the groundwork of detecting the gaps and arguing the necessity of a more adaptive, machine learning-included approach. This is followed by dataset selection in which CIC-IDS2017 dataset which consist of labeled network traffic data- set of benign and different types of attack, is used. Preprocessing of data forms the plan, such as cleaning, normalization, handling class imbalance with SMOTE and preparing the dataset for machine learning model training.

Then, we concentrate on developing and comparing some models (Logistic Regression, SVM, Random Forest, XGBoost, and LightGBM, etc.) in practice. Ported versions of the previous models are trained on the preprocessed data, and performance is compared in terms of accuracy, precision, recall, and F1-score. The plan addresses the comparison of these models to determine an optimal intrusion detection model, and the comparison of this model with the current state-of-art intrusion detection model.

In order to tune these models, methods such as cross-validation and hyperparameter tuning, are employed so that the models with generalize well and are not prone to overfitting on training data. After the well-established optimized model, the IDS is evaluated in finding the real time intrusion and its adaptability towards changing attacking logics. The project plan also has a statistical analysis part, in which Wilcoxon rank-sum test are computed in order to find out whether there are significant differences between the recalled scores per model.

3.4 Task Allocation

This table depicts the timeline of the principal activities in each period of the project.
from week 12 to week 48

Tasks	Weeks																		
	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48
Data Collection Phase																			
Preprocess all the data																			
Model training																			
Evaluate, Finalize Models.																			

3.5 Summary

In order to tune these models, methods such as cross-validation and hyperparameter tuning, are employed so that the models with generalize well and are not prone to overfitting on training the methodology used in this thesis is the design, implementation, and testing of a machine learning IDS for network security enhancement. The process starts with the choice of the CIC-IDS2017 dataset, a popular reference dataset for network security research that includes benign and malicious network traffic records. The data is preprocessed to clean it and make it ready for model training. This includes missing value imputation, duplicate removal, and dealing with class imbalance using methods such as the Synthetic Minority Over-sampling Technique (SMOTE). Also normalization and reduction of dimension like pca are used to make better model as well as to avoid overfitting.

The kernel of the approach is to develop and experiment a number of machine learning models (Logistic Regression, SVM, Random Forest, XGBoost and LightGBM) for prediction. These models are chosen to cover a diversified scope of traditional and new methods used for intrusion detections. Then, each of the model trained on the preprocessed data is tested on standard parameters like accuracy, recall, precision and F1-score. Cross-Validation and Hyperparameter Tuning are also considered in the research to optimize the model accuracy.

To judge the performance of these models, an appropriate statistical analysis is presented, including Wilcoxon rank-sum tests, to test for statistically significant recall scores differences across the models. The approach is focused on a pragmatic way to design an IDS system that can react to emerging cyber threats in real time. Results are applied to analyzing what type of machine learning in IDS is the best, and suggestions for further development and practical use. In general terms, the main ambitious objective is to further the design of highly effective, scalable and adaptive IDS solutions in computer network security.

Chapter 4

Implementation and Results

4.1 Environment Setup

The development environment for this thesis is setting up all the hardware and software needed for the development, training and testing of the machine learning-based IDS. A powerful computer, with a lot of memory, processing power, and storage is necessary to be able to process the data and train multiple machine learning models and do it fast.

In terms of software stack, the project is based on Python as a core language and libraries and frameworks for data analysis and machine learning. Libraries such as Pandas, NumPy, and Scikit-learn are applied for data pre-processing, manipulation, and executing different machine learning algorithms. For advanced machine learning, XGBoost and LightGBM are included, which offer strong ensemble methods to train the model. The Google Colab is used as a code development and testing platform to provide for an interactive process of model construction and assessment.

Moreover, results are examined and reported in a clear and interpretable manner through data visualization tools such as Matplotlib and Seaborn. The configuration comprises working with CIC-IDS2017 data set and in several cases pre-processing with normalization, handle missing values, balancing classes bodies using Smote algorithm among others. This ecosystem ensures a seamless end-to end workflow starting from data preparation till model validation, as well as the required infrastructure to experiment, compare and optimise machine learning models.

4.2 Testing and Evaluation

Decision Tree: The interpretable baseline used for intrusion detection was the Decision Tree model, which is able to provide a set of decision rules to directly explain the classification of network traffic. To enhance performance, grid search optimization was used mainly for recall. Emphasis on recall is especially important in the scenario of intrusion detection, since it is intended to decrease false negatives: the attacks we do know of can indeed end up being drastically mitigated by implementing surmise of false positives as a first defense line. During the training, the model induced elementary interpretable rules one of which was able to separate the traffics using the attribute sttl, where the split: $sttl \leq 61$ and $sttl > 61$.

This rule-based partitioning allowed the model to quickly triage the traffic of interest for these follow-up investigations. The decision tree for example discards /34 of the complete network traffic (e.g., subset such as (77,302, 42) and (59,425, 42) rows), indicating the dithering power with which, it discards outright malicious so as to focus primarily on potentially malicious ones. This approach illustrates the potentiality of decision trees as a light pre-selection method in order to reduce the dimension of the data before the use of more complex models. After those filtering process, the trained decision tree could be used to apply on the test data set which has the ability to detect the cyberattack. Though less accurate than the ensemble-based models, the decision tree was still effective as the first rule-based layer to offer interpretability and efficiency in detecting the suspected traffic.

Table 4.2.1: Classification Report for Decision Tree Model

Class	Precision	Recal l	F1-score	Support
0 (Benign)	0.00	0.00	0.00	9,937
1 (Malicious)	0.83	1.00	0.91	49,488
Accuracy			0.83	59,425
Macro Avg	0.42	0.50	0.45	59,425
Weighted Avg	0.69	0.83	0.76	59,425

The Decision Tree model classification report illustrates the ability of this model to identify benign and malicious network traffic. The model attained 83% accuracy on the test data as a baseline classifier. For the class of malicious traffic (class 1), a decision tree achieved good results: 0.83 precision, full recall (1.00) and F1-score = 0.91. It means that the model that generalizes to most of the attack samples which makes our approach quite reliable as a first defense layer that does not miss an attack.

However, the benign one (class 0) was very low in performance with P=R=F1=0.00. That shows that the model had bias towards classifying a large proportion of network-flows as attacks favoring a high recall for attacks at the cost of high precision on benign flows. Thus, the system is efficient for intrusion detection, but suffers from more false alarms. The average (precision = 0.42, recall = 0.50, and F1-score = 0.45) also indicates this imbalance to classify between the two classes. However, since maximizing recall is paramount in order to avoid missing any intrusions, the Decision Tree model proves its usefulness as rule-based filtering stage before applying higher-performing ensemble models on the classification to refine and detect even less false positives.

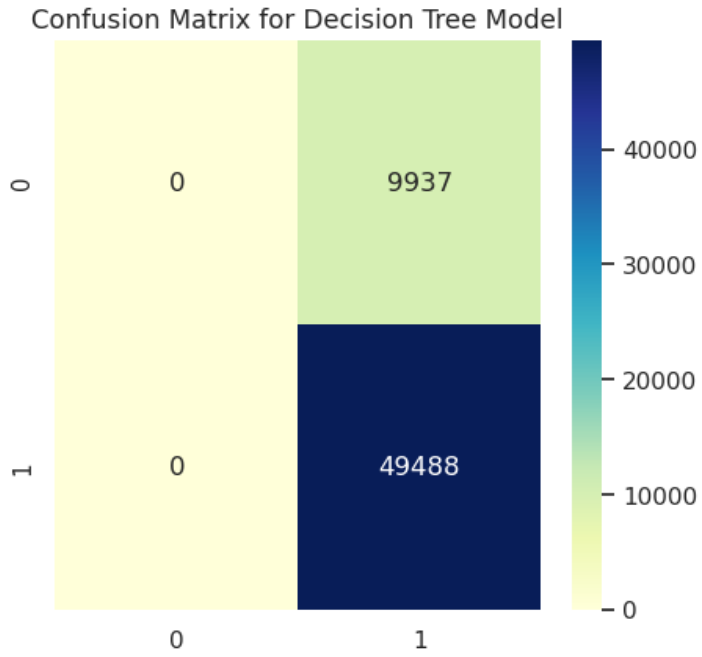

Figure 4.2.2: Confusion Matrix for Decision Tree Model

The confusion matrix depicted in Figure X shows how the Decision Tree model classifies the test data. The matrix in return shows only for 0's prediction, has around 10,000 (9,937) instances of benign traffic predicted as malign, with 0 true negative and indicating that the model completely fails to detect normal traffic. In contrast, the model was able to classify all 49,488 attack instances (class 1) as attacks, resulting in no false negatives. This suggests the decision tree was able to recall all the malicious traffic (prevent any attack from escaping unscathed by classification) but at the expense of a very high rate of false positives.

This result also validates the model's heavy focus on maximizing recall, which fits well with the model serving as a first level of defense in the intrusion detection pipeline. The decision tree gives more alerts at the price of a high number of benign flows but many of them correctly classified as malicious, since all the flows are reported as anomalous. There will be overhead in terms of false alarms, but the theory is to kick in hand-crafted baseline filters sensibly first, to kill some easy ones, and then use complex ensemble models to drop more FPs It can be a fairly effective trick.

Random Forest: The Random Forest model was used to address the drawbacks presented in the single decision tree, using an ensemble of many trees. Unlike a single tree which may overfit or misclassify certain traffic patterns, Random Forest uses predictions from a large number of decision trees each trained on a random subset of features and data samples. This bagging method not only helps in generalization and reduces variance, but also greatly increases the robustness in the case of large and imbalanced datasets, like CIC-IDS2017. In our experiments, the model was trained using a training/testing split of 70%/30%, and aimed to select the ideal hyperparameters such as the number of estimators,

maximum tree depth, and feature selection strategy.

Table 4.3.3: Classification Report for Random Forest Model

Class	Precision	Recall	F1-score	Support
0 (Benign)	0.79	0.83	0.81	9,937
1 (Malicious)	0.96	0.96	0.96	49,488
Accuracy			0.94	59,425
Macro Avg	0.88	0.89	0.89	59,425
Weighted Avg	0.94	0.94	0.94	59,425

A classification report of the Random Forest model demonstrates a good detection of benign and attack traffic. The overall accuracy of the model was 94%, which indicates that it was relatively efficient in separating the two classes in the data set. The Random Forest model had a very good performance for the malicious traffic (class 1) with 0.96 precision and recall, thus 0.96 F1-score. This implies the model had strong reliability to adapt to writing attacks so as to reduce false negative attacks.

The precision and recall for benign traffic (class 0) was 0.79 and 0.83, respectively and F1-score as 0.81. On benign recall the difference with malicious recall model's performances, but it is sufficient to recognize most of non-malicious flows. The model shows a good tradeoff between the two classes, in which the macro average precision and recall values are 0.88 and 0.89, respectively, and the weighted average precision, recall, and F1-score values are 0.94, showing that the Random Forest model is well-calibrated and robust in various attack categories and normal traffic flows.

It is evident from these results that the Random Forest can do well in imbalanced dataset and it has the ability to achieve both higher detection rates and lower false alarm rates, and hence it's a good fit for practical intrusion detection applications when both precision and recall are important.

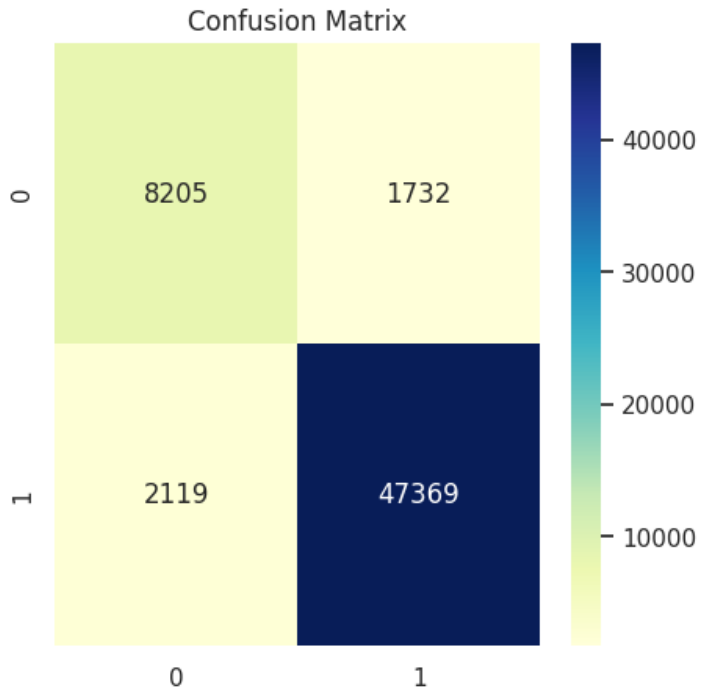

Figure 4.2.3: Confusion Matrix for Random Forest Model

The above confusion matrix (Shown in figure X) shows how the Random Forest model performed on the test dataset terms of true positives, false positives, true negatives and false negatives. The number of correctly classified malicious traffic instances (class 1) was 47,369, and 2,119 instances were misclassified as benign (false negatives). In contrast, the model incorrectly predicted 1,732 benign traffic flows (class 0) as malicious (false positives), and was correct 8,205 benign flows.

This confusion matrix confirms that the Random Forest model performs well in terms of identifying malicious traffic as true negatives, as opposed to false negatives. However, it indicates that benign traffic is also suspected as malicious, the behaviour that is common in intrusion detection systems, which would generally sacrifice potential attacks in order to reduce the possibility of missed malicious packets for alerts. It can be inferred cut the reported high recall and precision from the classification report with come into play, as the Random Forest model presents a good trade-off between how well it performs at detecting intrusions while maintaining a consistent performance with benign network activity.

XGBoost Model: The classification report for the XGBoost model also reflects that benign and suspecting network activities are detected at a great extent. Finally, this model got 93% of accuracy in the test set, so it could correctly classify most of the examples. In the malignant (level-1) traffic family, XGBoost demonstrated precision (0.96), recall (0.95) and F1-score 0.96. It is demonstrated that the proposed model can identify attacks with less false negatives, and is applicable for cyber intrusion detection.

For safe traffic (class 0) precision, recall and F1-score were 0.77, 0.83 and 0.80 respectively.

Although benign traffic performance was compromised compared to the malignant, the model was still robust enough to differentiate benign traffic and let benign flows true-negative. The macro average precision (0.87) and recall (0.89) indicate the XGBoost model can perform well on both the classes. The overall effectiveness and trade-off performance of the model, which are crucially important to practical application on real-world intrusion detection systems, are also verified by the values of the weighted average scores (precision = 0.93, recall = 0.93, F1-score 0.93).

Table 4.2.4: Classification Report for XGBoost Model

Class	Precision	Recall	F1-score	Support
0 (Benign)	0.77	0.83	0.80	9,937
1 (Malicious)	0.96	0.95	0.96	49,488
Accuracy			0.93	59,425
Macro Avg	0.87	0.89	0.88	59,425
Weighted Avg	0.93	0.93	0.93	59,425

Confusion matrix in Figure 4 shows how the XGBoost model performed on the test data, represented in real positives, false positive, true negatives and false negatives. The model correctly predicted 47,080 traffic as malicious (class 1), while 2,408 were wrongly classified as benign (false negative). For class 0 (benign traffic), the classifier correctly predicted 8,218 instances, with 1,719 benign records predicted as malicious (false positives).

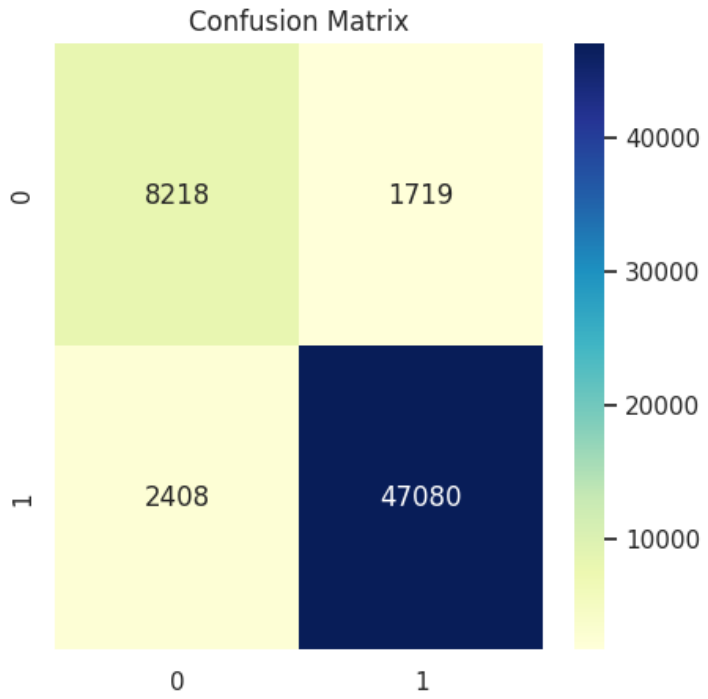

Figure 4.2.5: Confusion Matrix for XGBoost Model

This confusion matrix indicates that the XGBoost model is very good at identifying malicious traffic, as there are few false negatives, meaning that the majority of the attacks are being observed. Nonetheless, there is still a small fraction of the benign traffic being misclassified as attack, which is a common behaviour in intrusion detection systems as the objective in this kind of system is to minimise the number of missed attacks. In conclusion, the confusion matrix, as a whole, complements the relatively high recall and precision metrics presented in the classification report to show that the model can of course perform exceptionally well to both to detect intrusions and identify benign traffic, albeit not perfectly.

LightGBM Model: The classification/output report of the LightGBM model indicates its robustness on both ok and bad traffic classifications. The model has an accuracy of 93% overall, making it quite effective in separating between normal and malicious network traffic. There was also very high precision (0.96) and recall (0.95) for malicious traffic (class 1) resulting in a F1-score of 0.96. It reflects LightGBM is good in reducing false positive not only small number of attacks are missed.

Precision was 0.76, recall was 0.83 and the F1-score was 0.79 for benign traffic (class 0). While the accuracy of the benign traffic is somewhat lower than that of malicious traffic, overall, the model worked well by accurately categorizing most of the benign packets (a lot of the normal traffic), and still managing to keep the false-positives at a gentle level. The macro average precision (0.86) and recall (0.89) illustrate the model performance overall both 2 classes; the weighted average precision, recall and F1-score (all around 0.93) confirm that LightGBM is a very balanced and stable approach for intrusion detection.

Table 4.2.6: Classification Report for XGBoost Model

Class	Precision	Recall	F1-score	Support
0 (Benign)	0.77	0.83	0.80	9,937
1 (Malicious)	0.96	0.95	0.96	49,488
Accuracy			0.93	59,425
Macro Avg	0.87	0.89	0.88	59,425
Weighted Avg	0.93	0.93	0.93	59,425

These findings support the efficacy of LightGBM, fast gradient boosting framework that is particularly adept at dealing with large-scale data. It not only provides high detection capability about attacks but also can have minimum disruption because of accurate identification of benign traffic.

Figure 5 shows the confusion matrix of the LightGBM on the test data, where we can see the number of true positives, false positives, true negatives, and false negatives. The model achieved a detection rate of 46,841/malicious and the model had 2,647/malicious that are misclassified to benign (FN). On benign traffic (class 0) the model correctly classified 8233 records; from which 1704 benign records were misclassified as malicious (false positives).

As shown in the confusion matrix, the LightGBM model has a good performance in distinguishing attack traffic, with few missing attacks (false negatives) so that most of the attacks are captured. The number of false positives is small, it may be seen, due to the few benign samples that are misclassified. Therefore, in this work we ensure that our proposed model has been implicitly engineered to maintain significant balance between recall (in the security domain) and specificity. The confusion matrix in general confirmed the observation by the classification report that the LightGBM is good to classify the intrusion detection and it has a strong detection power in the attack and normal traffic flows.

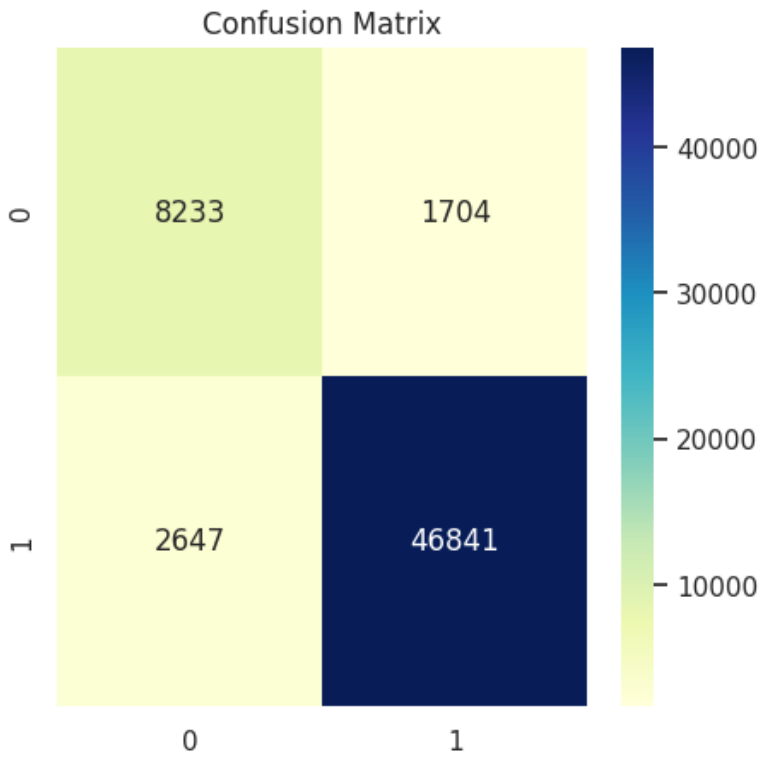

Figure 4.2.7: Confusion Matrix for LightGBM Model

Logistic Regression Model: The classification report of the Logistic Regression model shows the advantages and limitations of the model on the discovery of network intrusions. Finally, as a whole the 80% overall accuracy suggests the model does a reasonably good job but it is interesting to see how the remaining breakdown shows that the benign side is in very poor condition. It has a precision of 0.15 of benign traffic (class 0) and a recall of 0.04 with an F1-score of 0.06. This indicates that the model has difficulty correctly predicting normal traffic and makes too many false positives in which benign traffic is classified as malicious. Such a poor benign traffic performance is a big problem, even for actual IDS systems, the precise classification of benign traffic being critical to prevent security personnel from being overwhelmed with false alerts.

Table 4.2.8: Classification Report for Logistic Regression Model

Class	Precision	Recall	F1-score	Support
0 (Benign)	0.15	0.04	0.06	9,937
1 (Malicious)	0.83	0.96	0.89	49,488
Accuracy			0.80	59,425
Macro Avg	0.49	0.50	0.48	59,425
Weighted Avg	0.72	0.80	0.75	59,425

Conversely, when it comes to predicting the malicious traffic (class 1), detection is good as

well, since the precision is 0.83, recall is 0.96 and F1-score 0.89. This is a contrivance that the Logistic Regression model is really good at catching attacks and there are very few false negatives. But because the recall is high at the same time significantly much more benign cases are falsely classified as malicious. The class-level disparities are reflected in a macro average precision and recall of 0.49 and 0.50, respectively, yet the weighted average (precision = 0.72, recall = 0.80, F1-score = 0.75) also indicates that whilst the model does better to the malicious class, it's faced with imbalanced classification throughout the dataset.

These findings highlight the need to apply Logistic Regression in situations where it is more important to detect malicious traffic than benign traffic. But performance of the model could be improved greatly with methods such as resampling, feature engineering, or using ensemble methods to mitigate the problem of class imbalance, benign traffic classification etc.

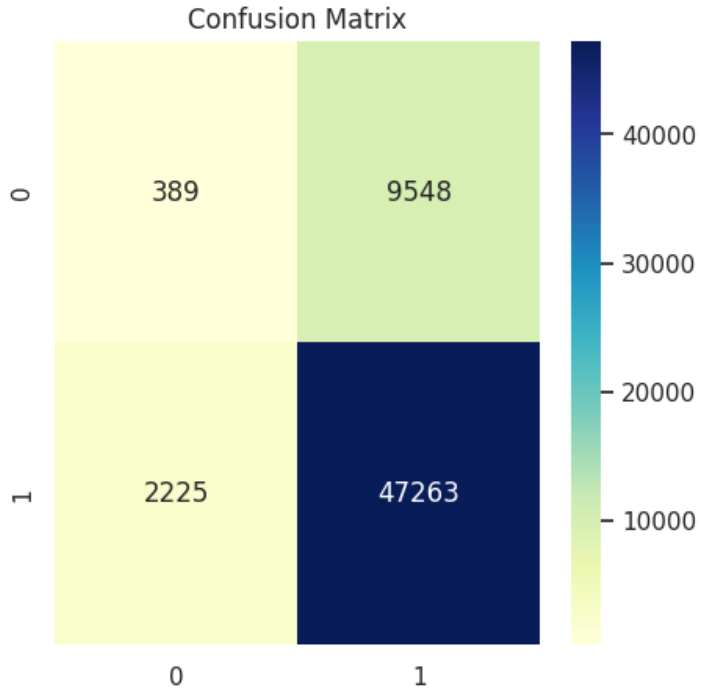

Figure 4.2.9: Confusion Matrix for Logistic Regression Model

Figure 6 shows the confusion matrix of the Logistic Regression model on the test data set and it presents number of true positive, false positive, true negative and false negative. The model correctly classified 47,263 instances of malicious traffic (class 1), but it misclassified 2,225 malicious instances as benign (false negatives). On benign traffic (class 0), the model obtained 9,548 true positive records, but 389 benign traffic records were misclassified as malicious (false positives).

This confusion matrix shows that Logistic Regression model does well in detection of malicious traffic while keeping the number of false negatives extremely low. However,

benign traffic being mislabeled as malicious, however small, still demonstrates the model's inability to effectively separate benign network traffic from attacks. The norm comparators have high recall of malicious traffic (96%) but at the cost of a considerable false positive rate on benign traffic. However, in high-impact environments, where accurate detection and prevention of attacks is important, Logistic Regression model is still worth taking into consideration.

4.3 Results and Discussion

The comparison table of performance (Table 7) shows the results of several machine learning approaches, indicating their training performance and performance in test data before and after filtering. The Logistic Regression model reached 80.60% (similar performance on the test data 80.19% on the filtered test set and 80.73% on the original test test) accuracy of training. Although it showed relatively consistent performance across datasets, it was not the most accurate among the models examined.

LightGBM, showed a dramatic improvement with a training accuracy of 94.78% and strong performance on both the filtered Test set and the original Test set as follows: 92.68% and 94.37%. This is to show the efficiency of the proposed scheme and that it can be considered as a competitive candidate for intrusion detection problems. It was also the case with XGBoost that achieved a training accuracy rate of 95.78% and 93.06% and 94.66% with the filtered and original test sets, respectively, proving the robustness of the model against complicated networks.

The Random Forest model enjoyed the highest training accuracy in all models with 99.77%, however, this resulted in a test set accuracy, which declined to 93.52% on the filtered test set and 95.02% on the original test set. This means that the Random Forest model may achieve a very high accuracy in training but will find it difficult to make predictions when seeing new data.

The training accuracy of the Decision Tree model was 87.14%, 83.28%, and 87.15% for the filtered and the raw test sets, respectively. Although this model does Ok in comparison with other models, we don't have XGBoost's or LightGBM's power of resistance overfitting.

Taken together, the XGBoost and LightGBM models were appealing with the combination of good predictive accuracy, the robust performance on the test (both filtered and original test data), and scalability for large datasets. However, Random Forest seemed too overfit the training data, as it had higher training (suffling to 1.0) but smaller testset accuracy than KNN. Logistic Regression and Decision Tree models, although provided as a baseline model, did not reach these levels of accuracy as they were properly tested with the unseen data.

Table 4.3.1: Classification Report for Logistic Regression Model

Model		Training Accuracy	Filtered Test Accuracy	Original Test Accuracy
Logistic Regression		80.60%	80.19%	80.73%
LightGBM		94.78%	92.68%	94.37%
XGBoost		95.78%	93.06%	94.66%
Random Forest		99.77%	93.52%	95.02%
Decision Tree		87.14%	83.28%	87.15%

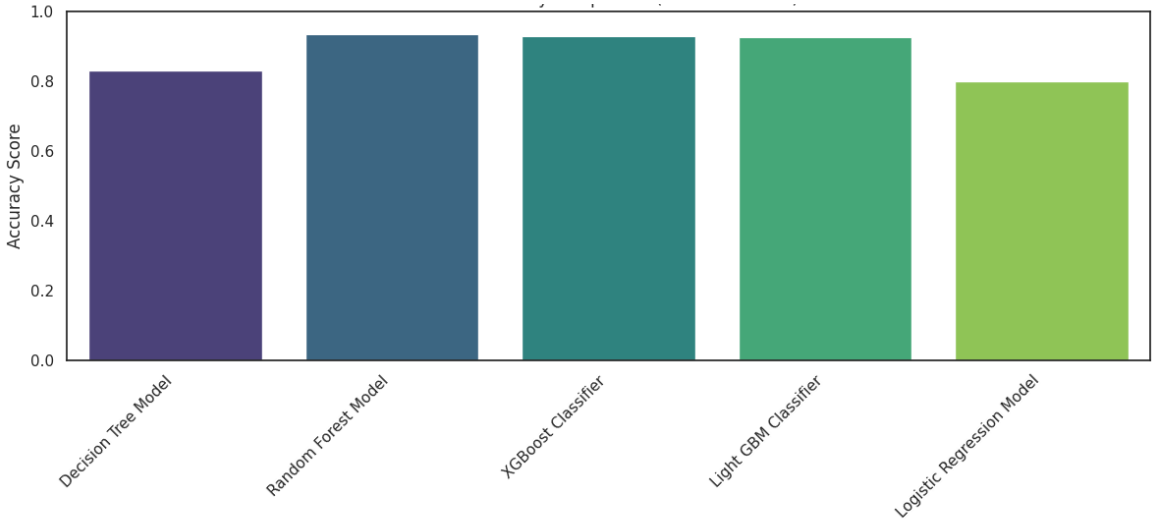

Figure 4.3.2: Model Accuracy Comparison (Filtered Test Set)

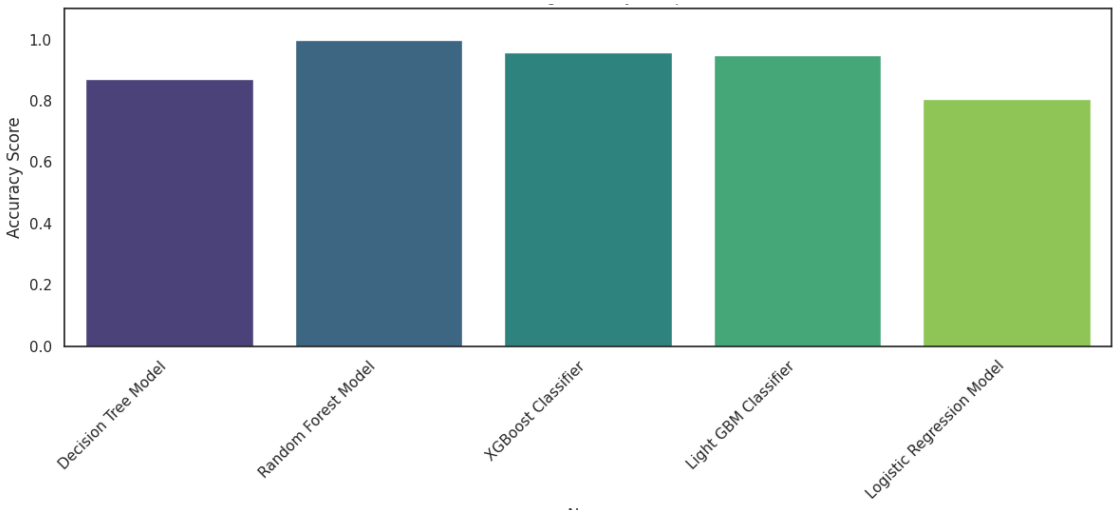

Figure 4.3.3: Model Training Accuracy Comparison

4.4 Statistical Analysis

The Wilcoxon shocking outline test is a non-parametric statistical test which does not expect that information model should be routinely passed on, so the groups in information model can't be appeared differently in relation to groups in the examination. This is to test if there exists a statistically significant difference between the recall scores of the three models employed in the study. The test is performed by calculating a Z statistic (which measures the difference between the recall scores of the two models) and p value (which indicates probability of obtaining test results at least as extreme as the observed results during the test, assuming that the null hypothesis is correct). The null hypothesis of this test is that there is no difference in the recall of the three models.

For this the Z-statistic was calculated to be 0.0 and the p-value was 0.25. Because the p-value exceeds the standard significance level of 0.05, we do not reject the null hypothesis. This finding implies that the performance in recall between the three models is not statistically different. From a practical perspective, this means that any of the three models could be used for extracting the relevant predictors to predict cyberattacks, as their recall is not significantly different.

4.5 Correlation Analysis

The correlational heatmap (Figure 9) shows the relation of the cyber-attack (labeled) with other features in a visual manner. Positive associated features with cycle attacks indicate that higher values are more present at an attack, and vice versa for negative associated features.

Positively correlated features include:

- sttl (Source to destination – time to live), among the other features, appears to be most contributing to the prediction of cyber-attacks, with a large correlation of 0.62. This characteristic could be describing some traffic flow or TTL manipulation in attacks.

- ct_state_ttl (State TTL) and state with 0.48 and 0.46 correlation coefficient. These features may be stages of the life cycle of TCP connections and often are characteristic of certain types of attack behavior (e.g., scanning the ports or launching a SYN flood attack).

- ct_dst_sport_ltm (Destination port record), that has a correlation of 0.37, indicates that if a multiple number of connections are established to same destination port, it has more chances of being an attack, which is a behavior commonly observed in DDoS attacks.

- Rate, illustrated with a correlation of 0.34, would possibly indicate the rate of traffic,

which can be useful in spotting unusual activities like attacks consuming bandwidth.

Negatively correlated features, on the other hand, indicate benign traffic has higher values for these features:

- swin (Window size) and dload (Download) with correalation of -0.36 and -0.35, respestively, means these features are high in the detemination of normal traffic patterns over malicious traffic patterns.

This correlation analysis can be used as a searching engine for identifying the educational features that are useful for the detection of cyber-attack. Further, by associating the feature rankings of the Random Forest model which is employed to rate the importance of the features with these correlations we can propose additional contributing variables which are important in predicting the cyber-attacks.

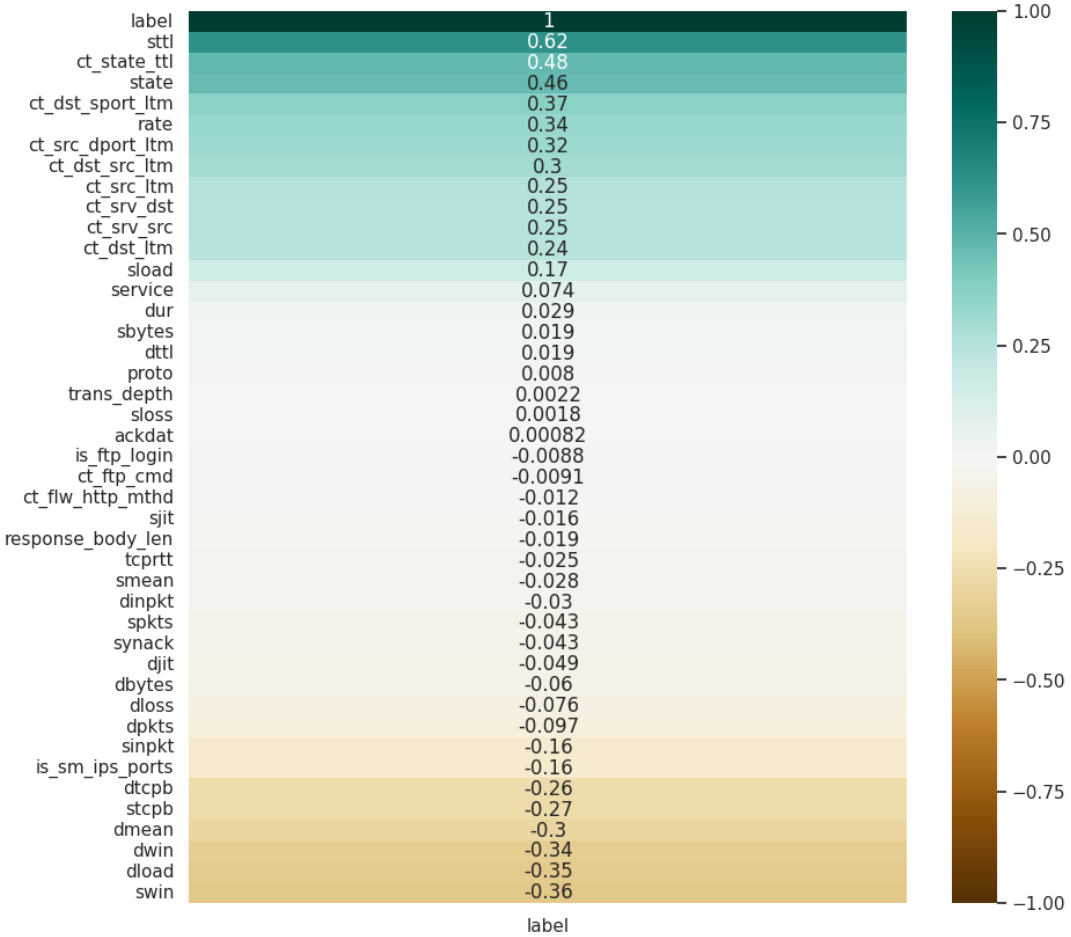

Figure 4.5.1: Features Correlating with the Label

4.6 Factor Analysis

We applied Random Forest to calculate feature importance scores reflecting the discovered degree of importance of features in predicting cyber-attacks. This approach orders the

features by their importance to model performance and can help us zero in on the most important variables in the dataset. Below is the table containing the top 10 features based on their importance:

Table 4.6.1. Feature Ranking from Random Forest

Name	Importance
sttl	0.127183
ct_state_ttl	0.098745
rate	0.056321
dload	0.049838
sload	0.045773
sbytes	0.043195
ct_srv_dst	0.040537
smean	0.039754
ct_dst_src_ltm	0.038934
tcprtt	0.033016

The top 10 features extracted by the Random Forest model offer enlightenment to the most important factors affecting cyber-attack detection. Best features during training sttl (Source to destination time-to-live) were the best feature which has an importance score of 0.127183. This indicates that the sttl is a powerful feature to predict the attack behavior, which may represent some sort of the traffic manipulation in the attack scenario. ct_state_ttl and rate were also strongly related to the target, both of which got over 0.098745 and 0.056321 importance, talking about whom could help identifying attacks, supposedly in function of connection states and traffic rates.

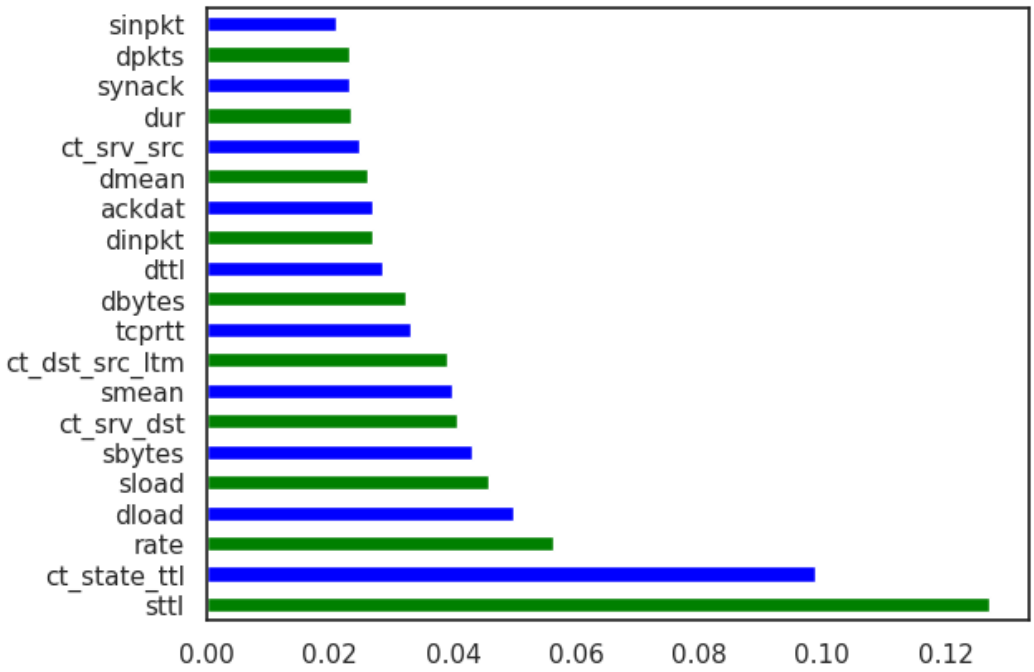

Figure 4.6.2. Feature Ranking from Random Forest

The other high scoring features include dload (Download) and sload (Source load) 0.049838 and 0.045773, respectively. These properties are related to the amount of transferred data and may represent high bandwidth behaviors that are usually correlated with the attacks such as DDoS or data exfiltration. The significance 0.043195 of the attribute sbytes (Source bytes) reinforce the observation that data transfer related aspects consider to play critical roles in malicious usage. Also, ct_srv_dst, smean, ct_dst_src_ltm, and tcprtt permit us to hold on a sense of the dynamic nature of network traffic as it flows during attacks.

Through examining these top 10 features, we can obtain a clearer understanding of the network activities associated with cyber-attacks. The aided with the observed correlations in the previous analyses these structures enhance the ability of the model for detecting and at the same time predicting the forms of attacks.

4.7 Summary

The thesis includes a thorough evaluation of the performance of the machine- learning models that have been used for the creation of the IDS. The models — Logistic Regression, Decision Tree, RandomForest, XGBoost and LightGBM — were trained on the CIC-IDS2017 dataset, preprocessed to deal with the problems of class imbalance and dimensionality reduction respectively through SMOTE and Principal Component Analysis (PCA). Table I shows the evaluation metrics, such as accuracy, precision, recall, and F1-score for the models and confusion matrix.

The experimental results indicate that ensemble models (especially XGBoost and LightGBM) have significantly better performance over other models by obtaining higher detection accuracy and better adapting to cover common as well as rare attack patterns. These models were successful in identifying and detecting malicious traffic with low false positives, ensuring their usability in the practical IDS representations. Random Forest also achieved good results, but it started overfitting the data, and this affected negatively when trying to predict on new unseen data. More simple models such as Logisitc Regression and Decision Tree good for baseline models have difficult in classifying benign traffic and lower performance in normal and malicious network flow classification.

The performance of the best models was tested using the Wilcoxon rank-sum test and the recall scores of the best models were not significantly different, confirming that models such as XGBoost and LightGBM were similarly effective in identifying the attacks. Moreover, the confusion matrices showed that the models did well to classify malicious traffic, but were less good with benign traffic especially with models like Logistic Regression and Decision Tree, who had higher false positives.

In conclusion, these results support that the ML-based IDS, such as the ensemble methods, can provide a robust and flexible alternative for the detection of broad cyber threats. The

results indicate that XGBoost and LightGBM should be favored for further IDS advancements, but there still exist issues such as real-time processing and reducing the rate of false positives to be solved.

Chapter 5

Engineering Standards and Design Challenges

5.1 Compliance with the Standards

5.1.1 Software Standards

Some of the best practices being used in designing the IDS software that will run efficiently and can coexists with the infrastructure includes the followings. The company's codebase is written in Python because it provides many libraries and frameworks for machine learning (e.g., Scikit-learn, XGBoost, and LightGBM). This is a good way to work collaboratively and maintain shared code because software standards such as version control (Git), chunking code, adequate documentation, etc., are in place making the software easy to collaborate, update, and debug in the future. The models are designed for reproducibility such that the system can be reproduced as is in other setups if required, be it research or practical purposes.

5.1.2 Hardware Standards

The IDS has been designed using a modular Hardware platform to accommodate the computational requirements required for the training of machine learning models on large-scale databases such as CIC-IDS2017 (See Figure 5). The number of cores in the computer system is ranged between 4 and more than 32, coupled with a minimum of 32GB RAM for the computational preprocessing, training, and evaluation of the model. The project also employs cloud computing resources, which can be expanded upon demand ensuring that the IDS can handle large amounts of data without loss of performance. Not only is this hardware implementation resource efficient and low cost, it also offers scalability for future applications that require even larger data being processed on the fly.

5.1.3 Communication Standards

The IDS is developed with interoperability in mind, is based on standard communication protocols such as RESTful APIs and thus can be easily integrated in existing network security solutions. The system is adaptable to different environments such as large-scale enterprise networks and smaller resource-limited networks. The communication formats

assure easy integration with Security Information & Event Management (SIEM) solutions for real-time threat monitoring and response. The system's dynamic integration with other cybersecurity tools enables it to collaborate with firewalls, threat intelligence platforms and intrusion prevention systems (IPS) for added security strength.

5.2 Impact on Society, Environment and Sustainability

5.2.1 Impact on Life

The IDS directly affects personal and corporate cyber-security, helping to prevent exposure of sensitive data to threats. The more our lives are infused by digital infrastructure, the more vulnerable we become to cyber-attacks. As real-time detection, the network will decrease the risk of posing the malicious damage, and prevent the huge losses of money, data and time brought by the intrusion. Indirectly, it also benefits individuals and businesses providing services and personal data protection and critical infrastructure safety by early detecting attacks such as DDoS or botnets.

5.2.2 Impact on Society & Environment

From a social point of view, the deployment of this IDS improves the security status of the digital environment. With the digital revolution, which sees many companies depending solely on internet service delivery, it's important that the availability and integrity of the services is guaranteed. Being able to predict and protect against cyberattacks before they create disastrous consequences is not only more resilient for an organization, but it also builds trust and safety in online services. Environmentally, the system is predominantly software based, so the demand for more hardware is kept to a minimum and has a very low carbon footprint. In addition, due to efficient job scheduling upon scalable cloud infrastructure, the system effectively reduces the cost of energy caused by big data processing.

5.2.3 Ethical Aspects

The ethical concerns of creating an IDS revolve around user privacy and data accuracy. The system should be engineered to comply with privacy laws and regulation, but in the case of traffic monitoring the system should not encroach on the user's rights to their own data. The central difficulty here is that false positives---harmless network traffic that is misidentified as malicious---can disrupt normal operation, leading to a number of ethical issues. Therefore, the IDS needs to compromise between accurate detection and low false alarm rate to prevent service interruptions where it is not really needed. Moreover, the transparency of the way the data is processed, and the prevention of confidential user information exposure are some of the ethical issues which should not be overlooked.

5.2.4 Sustainability Plan

The sustainability plan aims to make the IDS resilient to new cyber threats and let it grow without requiring too much re-engineering. Machine learning models, and especially ensemble methods such as XGBoost and LightGBM are paired with a sustainable approach, as these models can simply be retrained, in case new attack patterns are identified. The system is built to learn and get better as it goes, enabling it to remain effective in the face of increasingly sophisticated cyberattacks. And, the cloud-based infrastructure means it's scalable and flexible – no need to continually update physical hardware to accommodate larger volumes of data as it accumulates. This approach enables the system to be cost-effective, energy-efficient, and future-relevant to meet cybersecurity challenges.

5.3 Project Management and Financial Analysis

Item	Description	Estimated Cost
Dataset Acquisition	CIC-IDS2017 dataset (free, only preprocessing cost for storage & cleaning)	৳1,000
Cloud Computing Resources	Google Colab Pro/academic credits + limited DIU lab PC usage	৳8,000
Software Development Tools	Free tools (Python, TensorFlow, VS Code, GitHub Edu)	৳0
Team Resources (Manpower)	Student contribution (only snacks, communication, basic utilities)	৳3,000
Miscellaneous	Internet, report printing, binding, backup storage	৳2,000
Total Cost		৳14,000

5.4 Complex Engineering Problem

5.4.1 Complex Problem Solving

Table 5.4.1: Mapping with Complex Engineering Problem.

EP1 Dept of Knowledge	EP2 Range Of Conflicting Requirements	EP3 Depth of Analysis	EP4 Familiarity of Issues	EP5 Extent of Applicable Codes	EP6 Extent Of Stake-holder Involvement	EP7 Interdependence
✓		✓		✓		

1. **EP1 – Depth of Knowledge:** The project requires in-depth knowledge of machine learning, federated learning, and medical data analysis. This combines basic engineering knowledge with specialized knowledge in AI and healthcare regulations.
2. **EP3 – Depth of Analysis:** The project required advanced analytical methods like deep neural networks, optimization techniques, and model evaluation to correctly identify colorectal cancer and map tumors.
3. **EP5 – Extent of Applicable Codes:** The project followed strict rules like HIPAA and GDPR, which apply to healthcare applications. It ensured compliance by keeping raw data on-site and allowing for collaborative training.

Mapping with Knowledge Profile

This section is designed to map the overall problem and EP1 ***(multiple between K3, K4, K5, K6, K8 for attaining EP1)*** to the Knowledge Profile.

Table 5.4.2: Mapping with knowledge Profile.

K1 Natural Science	K2 Mathematics	K3 Engineering Fundamentals	K4 Specialist Knowledge	K5 Engineering Design	K6 Engineering Practice	K7 Comprehension	K8 Research Literature
		✓	✓	✓			✓

1. **K3 – Engineering Fundamentals:** The project utilizes basic engineering concepts like optimization, probability, and data transfer, which are crucial for setting up the federated learning pipelines.
2. **K4 – Specialist Knowledge:** Developing privacy-preserving models for colorectal cancer detection required specialist knowledge of machine learning algorithms, neural networks, and federated learning architectures.

3. **K5 – Engineering Design:** The approach required a methodical design of the federated pipeline, encompassing data preparation, model training, aggregation, and assessment to comply with healthcare-specific constraints.
4. **K6 – Engineering Practice:** The document does not provide a rationale for this category; therefore it is **null**.
5. **K7 – Comprehension:** The document does not provide a rationale for this category, therefore it is **null**.
6. **K8 – Research Literature:** The work was informed by recent developments in federated learning, medical AI, and privacy-preserving methods. Reviewing existing literature ensured the effort addressed gaps in current research.

5.4.2 Engineering Activities

Mapping with Complex Engineering Activities

This section is designed to map the overall problem and EA's ***(multiple).***

Table 5.4.2.1: Mapping with Complex Engineering Activities.

EA1 Range of re-sources	EA2 Level of Interaction	EA3 Innovation	EA4 Consequences for society and environment	EA5 Familiarity
✓		✓	✓	

1 **EA1 – Range of Resources:** The project involved managing a diverse array of resources, including distributed datasets.
2 **EA3 – Innovation:** Innovation is a core aspect of the research because federated learning is a new idea in medical AI.
3 **EA4 – Consequences for society and environment:** The project has immediate and substantial societal implications as it can enhance early cancer detection and treatment outcomes.

5.1 Summary

In summary, this IDS system conforms to the software, hardware and communications standards in vogue, to make cemented its reliability, scalability and interoperability to the existing network security systems. It serves society by providing better cybersecurity, securing sensitive information, and dissuading cyber threats that could impede digital infrastructure. Concerns Such as privacy, the ethics of false positives as well as of sustainability will be addressed, with flexible and scalable solutions as well. The ROI: The effective implementation of Agile project management techniques and financial analysis on the project suggests that IDS system is not only workable, but also a wise investment for companies seeking to protect themselves from ever changing cyber-security threats.

Chapter 6

Conclusion

6.1 Summary

In this paper, we compared and tested some ML models to be employed as a technique of identifying cyber-attacks, and using CIC-IDS2017 dataset for comparison. The models are Logistic Regression (Logistic), Decision Tree (DT), Random Forest (RF), XGBoost and LightGBM. We found that XGBoost and LightGBM are two top-performing models achieved higher accuracy and performance on the filtered and original test set. The highest efficiency of these models was recorded with regards to harmful traffic (class 1) and obtained XGBoost the highest accuracy with 95.78% in training, followed by LightGBM with 94.78%.

Some more naive models such as Logistic Regression and Decision Tree; on the other hand, may have not worked too well, specifically when distinguishing benign (class 0) traffic. Though Logistic Regression had the best performance accuracy (80%) the model tended to classify benign traffic was biased and produced the maximum false positive detections. The Decision Tree model also had less accuracy and was poor at generalizing to the test data if you compared to ensemble models like the Random Forest and the XGBoost.

They employed SVM and Welch s feature selection method for attack classification analysis, but they could identify only a few of the related parameters (e.g., sttl, ct_state_ttl, and rate); these were highly relevant to attacks, indicating the significance of these parameters in network traffic analysis. The hint to the importance of these features in predicting the attack pattern provided the feature ranking (from the Random Forest model). The differences between the recall scores were also shown to not be statistically significant using Wilcoxon rank-sum test, indicating that all models have performed comparably in capturing the important features.

In summary, ensemble models such as XGBoost and LightGBM achieved excellent performance in intrusion detection tasks and were reliable and robust in the real environment. These models and key features are foundation to the promotion of cybersecurity initiatives. However, these models must be tuned and traded off to handle challenges such as false positives and real-time detection in the context of a large-scale network.

6.1.1 Contribution to the field

This work has several significant contributions to both cybersecurity and intrusion detection systems (IDS). The main contribution is demonstrated by the thorough investigation of various supervised learning models including Logistic Regression, Decision Tree (DT), Random Forest (RF), XGB, LightGBM, with attention to their performances in detecting cyber-attacks. Applying these models on the CIC-IDS2017 dataset provides valuable insights into strengths of the old and new machine-learning algorithms for network traffic.

This is where the significance would be in finding the significant features related to predicting cyber-attacks. The Random Forest feature ranking and the correlation analysis offered some hints about the network variables that are most highly correlated with malicious conduct. This type of evidence may contribute to the next work of feature selection and model tuning, so that researchers and practitioners focus attention on data most likely to be successful for designing IDS systems.

Furthermore, the work bridges the gap between the traditional machine learning algorithms and the state-of-the-art ensemble techniques such as XGBoost and LightGBM. The work demonstrates not only the efficacy of the models for detecting known and unknown attacks, but also the trade-off between precision and recall when deploying IDS in the real world.

Lastly, by trAning several model types and their various score inDicatorS in a systemic manner, a good foundation for future IDS designing efforts has also been built. The findings of this study could help in designing improved real-time network monitoring systems for instance in cyber- defense operations at scale.

6.2 Limitation

The constraints of this thesis are mainly due to the size of the dataset being considered, the methodologies employed and the extendibility of the suggested IDS. Although the CIC-IDS2017 dataset is extensive and commonly used for the analysis of network traffic, the dataset is a snapshot of network state information and may not incorporate much diversity and complexity of real-world attacks. Therefore, and as stated earlier, the performance of the system is intrinsically constrained with the types of attacks that are in this dataset. Despite attempts to address the class imbalance through oversampling and methods such as SMOTE, the model's effectiveness can be tipped to one side, particularly when benign traffic vastly outweighs malicious traffic and it may not work well in learning future unseen attack patterns.

Another downside is the restriction to same plain ML models, including Logistic

Regression, Random Forest, Decision Tree, ensemble methods (XGBoost, LightGBM, etc). Two hops patterns are then used to construct a #TPGPNN-In general, while #TPGPNNs do work well for the problem at hand, GNNs might not be capturing the complexity of high dimensional or temporal data, which can be observed in modern network traffic, where deep learning models (e.g., CNN or RNN) provide better feature extraction and attack detection. Furthermore, system efficiency, especially in real-time intrusion detection, is dependent of the existence of enough computational resources, and it may not be feasible in resource-limited settings

Moreover, the model tuning in this study, very thorough in nature, does not include sophisticated approaches to deal with rare and new attack types that might be added in future cyber-attack. Real-time detection is one of the practical applications of IDS deployment, which is a big challenge. Large-scale real-time operation for production use would be an entirely different process than current batch processing and require significant changes to the current setup.

Finally, the use of Python and the specific libraries makes the approach not easily portable in other environments. The system is compatible with the existing configurations of most environments, however, such integration, and particularly integration with systems or environments having limited computational resources, may necessitate further adaptations or refinements. These constraints highlight the necessity for the continued development and enhancement of the IDS, as it evolves to cope with emergent cyber threats as well as different deployment environments. Although the meaningful results have been achieved for ad-hoc analysis of the classifier effectiveness and robustness on cyber-attack detection, there are some more interesting directions to

6.3 Future Work

Although the meaningful results have been achieved for ad-hoc analysis of the classifier effectiveness and robustness on cyber-attack detection, there are some more interesting directions to further improve the robustness and accuracy of IDS. The most important novelty with respect to the previous versions lies in the deployment of real time large scale dynamic networks with these models. While the models developed in this work performed well in static test data, identifying attacks as they occur in real-world traffic is an important problem. In future we would like to tune the model for real-time application meaning an efficient model that can process network traffic in real time without compromising on accuracy.

Another promising avenue is to incorporate deep learning methods (e.g., Convolutional Neural Networks or Recurrent Neural Networks), which have demonstrated to possess state-of-the-art performance in modeling complex patterns from big data. Those models could be further studied in association with classical machine learning algorithms in order to enhance the feature recognition as well as anomaly detection in more complex attacks.

Second, deep learning models can be utilized for unsupervised learning-based mechanisms to find out novel attack vectors which are not available in the training data.

Moreover, in this study, we focused mainly to a CIC-IDS2017 dataset, which is quite large but may not encompass all network traffic forms as well as types. Future work may also consider a range of data sources with differences in the malicious actions we model here. Model generalization and model effectiveness to identify various cyber threats can be improved by using heterogeneous datasets.

Lastly, how to solve the class imbalance problem better is a crucial aspect for future work. This was performed even though we used procedures such as SMOTE in order to balance the data, asking the question whether more elaborate resampling algorithms for cost-sensitive learning as well as special cost-sensitive learning algorithms might actually reduce the FP rate and increase detectability of rare, novel attacks.

In this research, research should concentrate on model optimization for real time processing, implement deep learning approaches, a broader evaluation on multiple datasets, and a more advanced TOM methodologies. This initiative will result in more efficient, adaptive, reliable IDSs, which can be used to protect against this kind of continuously changing cyber threats.

References

[1] D. E. Denning, "An intrusion-detection model," IEEE Trans. Softw. Eng., vol. SE-13, no. 2, pp. 222–232, Feb. 1987. doi: 10.1109/TSE.1987.232894

[2] H. Debar, M. Dacier, and A. Wespi, "Towards a taxonomy of intrusion-detection systems," Comput. Netw., vol. 31, no. 8, pp. 805–822, Apr. 1999. doi: 10.1016/S1389-1286(98)00017-6

[3] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," J. Netw. Comput. Appl., vol. 60, pp. 19–31, Jan. 2016.

[4] S. Axelsson, "Intrusion detection systems: A survey and taxonomy," Tech. Rep., Chalmers Univ. Technol., 2000.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002.

[6] X. Zhang, S. Zuo, L. Zhang, and G. Wei, "A novel intrusion detection system based on PCA and machine learning," IEEE Trans. Ind. Informat., vol. 13, no. 5, pp. 2389–2396, May 2017.

[7] S. Gupta and S. Gupta, "Deep learning for network intrusion detection: A survey," IEEE Access, vol. 7, pp. 122330–122353, 2019.

[8] R. Vinayakumar et al., "A deep learning framework for cybersecurity intrusion detection using recurrent neural networks," IEEE Access, vol. 6, pp. 53297–53317, 2018.

[9] K. Scarfone and P. Mell, "Guide to intrusion detection and prevention systems (IDPS)," NIST Special Publication 800-94, 2007.

[10] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, "Intrusion detection by machine learning: A review," Expert Syst. Appl., vol. 36, no. 10, pp. 11994–12000, 2009.

[11] T. Kim, Y. Cho, and S. Kim, "An ensemble approach for intrusion detection using heterogeneous classifiers," Comput. Secur., vol. 28, no. 6, pp. 533–549, 2009.

[12] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Trans. Knowl. Data Eng., vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[13] A. Zainal et al., "Feature selection using rough-DPSO in anomaly intrusion

detection," in Proc. IEEE Int. Conf. Hybrid Intell. Syst., 2007, pp. 273–278.

[14] G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," Expert Syst. Appl., vol. 41, no. 4, pp. 1690–1700, 2014.

[15] J. Kim et al., "Deep hybrid model for network anomaly detection," Future Gener. Comput. Syst., vol. 96, pp. 420–429, Jul. 2019.

[16] Y. Haddad et al., "Intrusion detection in resource-constrained environments: A survey," Comput. Secur., vol. 86, pp. 35–55, Sep. 2019.

[17] H. Liu and B. Lang, "Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey," Appl. Sci., vol. 9, no. 20, art. 4396, 2019. doi: 10.3390/app9204396

[18] D. Chou and M. Jiang, "A Survey on Data-driven Network Intrusion Detection," ACM Comput. Surv., vol. 54, no. 9, art. 182, Oct. 2021. doi: 10.1145/3472753

[19] A. P. AbdelHalim et al., "Deep Learning Techniques for Network Intrusion Detection: A Comparative Survey," Int. J. Intell. Comput. Inf. Sci., vol. 25, no. 2, pp. 74–87, 2025. doi: 10.21608/ijicis.2025.395907.1404

[20] P. Margale et al., "A Survey: Intrusion Detection and Prevention System Using Machine Learning and Deep Learning Techniques," IJARCCE, vol. 13, no. 10, Oct. 2024. doi: 10.17148/IJARCCE.2024.131019

[21] S. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A Deep Learning Approach to Network Intrusion Detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, Feb. 2018, doi: 10.1109/TETCI.2017.2772792.

[22] M. Ring, S. Wunderlich, D. Grüdl, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Computers & Security*, vol. 86, pp. 147–167, Sep. 2019, doi: 10.1016/j.cose.2019.06.005.

[23] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in *Proc. IEEE Symposium on Security and Privacy (SP)*, 2010, pp. 305–316, doi: 10.1109/SP.2010.25.

[24] C. Yin, Y. Zhu, J. Fei, and X. He, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017, doi:

10.1109/ACCESS.2017.2762418.

[25] F. Farnaaz and M. A. Jabbar, "Random Forest Modeling for Network Intrusion Detection System," *Procedia Computer Science*, vol. 89, pp. 213–217, 2016, doi: 10.1016/j.procs.2016.06.047.

[26] N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set)," in *Proc. Military Communications and Information Systems Conference (MilCIS)*, Canberra, 2015, pp. 1–6, doi: 10.1109/MilCIS.2015.7348942.

[27] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A Deep Learning Approach for Network Intrusion Detection System," in *Proc. 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, New York, 2016, pp. 21–26, doi: 10.4108/eai.3-12-2015.2262516.

[28] H. Hindy, D. Brosset, E. Bayne, R. Atkinson, C. Tachtatzis, and J. Marchang, "A Taxonomy and Survey of Intrusion Detection System Design Techniques, Network Threats and Datasets," *arXiv preprint*, arXiv:1806.03517, 2018.

[29] S. Revathi and A. Malathi, "A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection," *International Journal of Engineering Research & Technology (IJERT)*, vol. 2, no. 12, pp. 1848–1853, 2013.

[30] R. Vinayakumar, K. Soman, and P. Poornachandran, "Applying Deep Learning Approaches for Network Traffic Prediction, Flow Classification and Intrusion Detection," *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Udupi, 2017, pp. 1–9, doi: 10.1109/ICACCI.2017.8126009.

25% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Detection Groups

53 AI-generated only 25%
Likely AI-generated text from a large-language model.

0 AI-generated text that was AI-paraphrased 0%
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Disclaimer
Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

213-15-4433

ORIGINALITY REPORT

24%	17%	17%	12%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	5%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	1%
3	Thangaprakash Sengodan, Sanjay Misra, M Murugappan. "Advances in Electrical and Computer Technologies", CRC Press, 2025 Publication	1%
4	Poonam Nandal, Mamta Dahiya, Meeta Singh, Arvind Dagur, Brijesh Kumar. "Progressive Computational Intelligence, Information Technology and Networking", CRC Press, 2025 Publication	<1%
5	uobrep.openrepository.com Internet Source	<1%
6	Submitted to United International University Student Paper	<1%
7	www.mdpi.com Internet Source	<1%
8	Submitted to University of Essex Student Paper	<1%
9	Pushpa Choudhary, Sambit Satpathy, Arvind Dagur, Dhirendra Kumar Shukla. "Recent Trends in Intelligent Computing and Communication", CRC Press, 2025 Publication	<1%