A Machine Learning Approach for Multi-Attack Classification in Intrusion Detection Systems with the CICIDS2017 Dataset

Didarul Islam Sifat

Bachelor of Science

DAFFODIL INTERNATIONAL UNIVERSITY

DAFFODIL INTERNATIONAL UNIVERSITY

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name	:	Didarul Islam Sifat
Date of Birth	:	26/05/2003
Title	:	A Machine Learning Approach for Multi-Attack Classification in Intrusion Detection Systems with the CICIDS2017 Dataset.
Academic Session	:	2022-2025

I declare that this thesis is classified as:

☐	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
☐	RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☐	OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Daffodil International University reserves the following rights:

1. The Thesis is the Property of Daffodil International University.
2. The Library of Daffodil International University has the right to make copies of the thesis for the purpose of research only.
3. The Library of Daffodil International University has the right to make copies of the thesis for academic exchange.

Certified by:

sifat

_____
(Student's Signature)

_____
Student ID: 221-35-839
Date: 27/11/2025

_____
(Supervisor's Signature)

_____
Name of Supervisor: Dr. Md. Abdul Kader
Date: 27/11/2025

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

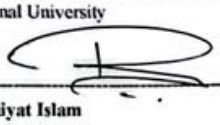
APPROVAL

This thesis titled on "**A Machine Learning Approach for Multi-Attack Classification in Intrusion Detection Systems with the CICIDS2017 Dataset**", submitted by **Didarul Islam Sifat (ID: 221-35-839)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS

Chairman

Dr. A. H. M. Saifullah Sadi
Professor
Department of Software Engineering
Faculty of Science and Information Technology Daffodil International University



Internal Examiner 1

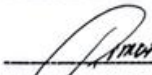
Dr. Rubaiyat Islam
Associate Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2

Dr. Md. Abdul Kader
Associate Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 3

Nuruzzaman Faruqui
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University



External Examiner

Md. Mostafiz Khan
Managing Director
Tecognize Solutions Limited

THESIS DECLARATION LETTER (OPTIONAL)

Librarian,
Daffodil International University,
Daffodil Smart City,
Ashulia.Dhaka,Bangladesh

Dear Sir,

CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name Didarul Islam Sifat
Thesis Title A Machine Learning Approach for Multi-Attack Classification in Intrusion Detection Systems with the CICIDS2017 Dataset

Reasons (i)

(ii)

(iii)

Thank you.

Yours faithfully,

_____
(Supervisor's Signature)

Date:27/11/2025

Stamp:

Note: This letter should be written by the supervisor and addressed to the Librarian, *Daffodil International University* with its copy attached to the thesis.

SUPERVISOR'S DECLARATION

I/We* hereby declare that I/We* have checked this thesis/project* and in my/our* opinion, this thesis/project* is adequate in terms of scope and quality for the award of the degree of *Bachelor of Science/ Master of Science.

(Supervisor's Signature)

Full Name : Dr. Md. Abdul Kader

Position : Associate Professor

Date : 27/11/2025

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Daffodil International University or any other institution.

sifat

_____

(Student's Signature)

Full Name : Didarul Islam Sifat

ID Number : 221-35-839

Date : 27 November 2025

A Machine Learning Approach for Multi-Attack Classification in Intrusion Detection Systems with the CICIDS2017 Dataset

DIDARUL ISLAM SIFAT

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Bachelor of Science

Department of Software Engineering (Major in Cyber Security)

DAFFODIL INTERNATIONAL UNIVERSITY

December 2025

ACKNOWLEDGEMENTS

I would like to show that my thesis supervisor has been of great help, constantly supporting and giving constructive feedback during the whole research process, which is why I want to thank him. The experience and patience he has gained have been crucial in developing this work and prompting me to think critically and on my own.

I also want to express my appreciation to the member staff of the Department of Software Engineering of Daffodil International University who have offered me the academic background and support to make this research a successful one. They have given much support and enthusiasm towards my learning process.

I would like to express my heartfelt gratitude to the Canadian Institute of Cybersecurity that released the CIC-IDS2017 dataset that helped to conduct this research. Having such a quality and realistic dataset has played a significant role in the creation and testing of the intrusion detection framework.

My family as well as my parents have been of unconditional support, encouragement and sacrifice during my academic life and I am very grateful to them. My strength and motivational factor has always been the people who believed in me either way or another.I owe my success in this research to the people.

DEDICATION

This thesis is dedicated to my adorable parents, the unconditional love, support, and sacrifices are what have made my academic life. Their encouragement and prayers always been my best strength. I also dedicate this work to my teachers and mentors, whose help and encouragement have made me passionate about learning and research.

ABSTRACT

A fast increase in the digital connectivity in different areas like the finance sector, healthcare sector, and the military has made it much easier to be targeted by cyberattacks, which have revealed the weakness of the traditional signature-based intrusion detection systems (IDS). To solve this, machine learning (ML) will provide intelligent solutions that are adaptive. This paper creates an explicatory ML-based IDS framework based on the CIC-IDS2017 dataset, having more than 2.8 million records. The data pre-processing consisted of noise elimination, label encoding, feature scaling and feature selection with the help of the Random Forest. The top model was the XGBoost that was optimized and trained through RandomizedSearchCV. Accuracy, precision and recall as well as the F1-score and AUC-ROC were used to compare model performance, and the interpretability was achieved through SHAP (SHapley Additive Explanations). This is the best model, with XGBoost having a perfect accuracy (1.00) of all types of attack and a high F1-Scores of 0.99 of Web Attacks and 0.89 for Infiltration. These findings highlight the fact that XGBoost is better positioned to deal with common and rare cyberattacks, and can therefore be very useful in the intrusion detection systems of the real world. Huge recall and low false positive rates also prove that the model can be adopted in large-scale, real-time cybersecurity systems in which accuracy and ability to react is paramount. The SHAP analysis indicated the following characteristics, which were variable in terms of packet length, destination port and flow-based attributes which are critical in understanding and interpretation of the decision making process of the model. This study points out that XGBoost, with its accuracy, efficiency, and explainability, is the right model to use in the creation of scalable, interpretable IDS. The proposed framework offers a robust, reproducible approach that can be seamlessly deployed across diverse, high-throughput network environments, providing significant value to cybersecurity efforts.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

$I(p_i)$	impurity measure
n_i	number of samples at node i.
n_{left}, n_{right}	number of samples in left and right child nodes.
p_{left}, p_{right}	class probability distributions in child nodes.
K	Total number of features.
S	any subset of features that doesn't contain i.
$\|S\|$	cardinality of a subset S.

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
APT	Advanced Persistent Threat
AUC	Area Under the Curve
CIC	Canadian Institute for Cybersecurity
CNN	Convolutional Neural Network
DL	Deep Learning
DoS	Denial of Service
DDoS	Distributed Denial of Service
FPR	False Positive Rate
IDS	Intrusion Detection System
IoT	Internet of Things
KDD	Knowledge Discovery in Databases
LIME	Local Interpretable Model-Agnostic Explanations
LSTM	Long Short-Term Memory
MLP	Multi-Layer Perceptron
ML	Machine Learning
NSL-KDD	Network Security Laboratory–KDD dataset
ROC	Receiver Operating Characteristic
RNN	Recurrent Neural Network
SHAP	SHapley Additive exPlanations
SMOTE	Synthetic Minority Over-sampling Technique
SVM	Support Vector Machine
TPR	True Positive Rate
XAI	Explainable Artificial Intelligence
XGBoost	Extreme Gradient Boosting
RF	Random Forest
CPU	Central Processing Unit
GPU	Graphics Processing Unit
CSV	Comma-Separated Values
ReLU	Rectified Linear Unit
SGD	Stochastic Gradient Descent

TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background of the Study

Within the recent decades, the fast development of information and communication technologies radically transformed the economic, social and institutional systems throughout the globe. Now banking, healthcare, e-commerce, education, public administration and critical infrastructure are running on highly interconnected computer networks and on cloud-based platforms[1], [2]. This ubiquitous digital connectivity has provided significant improvements in efficiency, flexibility and accessibility and enables real-time transaction, remote services and huge data transfer. Simultaneously, it has put forward a vast and ever-growing attack surface on malicious actors, including individual hackers and cybercriminal groups as well as state-sponsored attackers [3], [4], [5]. With increasing dependence on digital services by organisations, the number, nature and complexity of cyberattacks, e.g. denial-of-service (DoS), distributed denial-of-service (DDoS), brute-force logins, botnet, internal system intrusion and advanced web-based attacks, has grown significantly[6], [7].

Conventional security mechanisms were created during a time when networks were smaller, traffic patterns were less complex and threats have developed more slowly. Such mechanisms such as the access control policies, perimeter firewall and signature based antivirus software continue to hold significant importance as the first line of defence[8], [9]. Mostly, however, they are mostly fixed and rule-based: firewalls are based on written rules concerning which packets should be permitted, antivirus engines are based on lists of known malware signatures and access control lists encode relatively fixed concepts of what users are permitted to do. The purely static defences are getting weaker and weaker as the attackers are embracing more dynamic and evasive attacks - polymorphic malware, encrypted command-and-control channels, stealthy lateral movement, etc.[10], [11]. To combat this challenge; Intrusion Detection Systems (IDS) was developed to monitor network or host activity in an effort to identify suspicious behaviour and alert to be investigated upon[12], [13].

There are two historical paradigms of IDS: signature-based and anomaly-based detection. Signal-based IDS compares observed traffic to a database of signatures. They are accurate and can be interpreted to known threats but are in nature reactive as they are unable to detect newly introduced or slightly altered attacks that are not within existing signatures [14]. Anomaly-based IDS on the other hand would strive to learn normal behaviour and report on anomalies as a possibility of an intrusion. The estimation of baseline behaviour in systems and networks has been done through the use of statistical profiling, clustering and rule-based heuristics[15]. Theoretically these systems provide the capability to identify unknown or zero-day attacks, however practically they are susceptible to unstable baselines in dynamic environments and a large false-positive rate that overloads the security personnel[16], [17]. Hybrid IDS architectures integrate the two paradigm elements to use their complementary advantages, but share some of their limitations as well[18].

The transition to machine learning (ML) and deep learning (DL) methods of intrusion detection has been very rapid with the growing accessibility of network traffic data and the development of artificial intelligence. As an alternative to solely using rules that are manually developed, ML- and DL-based IDS study patterns of normal and malicious behaviour based on past data and apply these patterns to classify new connections or flows[19], [20], [21], [22]. Decision Trees, Support Vector Machines (SVM), Naive Bayes and k-Nearest Neighbours are widely examined classical ML algorithms, whereas ensemble algorithms like the Random Forests, XGBoost and CatBoost have become immensely effective in tabular network data[23], [24], [25], [26]. Parallel to this, there has been the investigation of DL networks- ANN, MLP, CNN , RNN and LSTM to learn complex representations based on traffic features or even raw packet data[20], [21].

In order to justify the intensive consideration of these methods, multiple benchmark datasets are suggested. The older datasets like the KDD99 and NSL-KDD were heavily utilised until they are more broadly criticised due to synthetic traffic patterns, old attacks and artefacts which could result in misleadingly high performance [23], [24]. Generalized later datasets like CIC-IDS2017 have been created to represent more closely the current realities of networks. CIC-IDS2017 is realistic with benign traffic and a wide range of modern attacks, DoS, brute force (FTP/SSH), botnet, infiltration and web attacks, defined by 79 flow-level features on a base of over 2.8 million records[25], [26]. This is the reason why it is currently one of the most widely

used datasets in the IDS research. CIC-IDS2017, however, is also extremely skewed, where benign traffic and a small number of high-volume attacks are the largest and most prevalent, with such critical classes as infiltration and some web attacks severely underrepresented[27].

Though the application of ML and DL has a promising future, there are several crucial issues associated with the application to the IDS. Poor generalisation can be caused by high dimensional, noisy, imbalanced data; in particular, rare attacks[28], [29]. Deep neural networks and other models that are very complex in nature tend to work like black boxes whose internal decision process is obscure and as such, security analysts can hardly trust or comprehend their results[30], [31]. Simultaneously, the regulatory, organisational requirements are affecting an increasing need to be transparent, accountable and audit AI-based systems, particularly in areas sensitive to security. Respondent explainable AI (XAI) approaches like SHAP (SHapley Additive Explanations) and LIME have been developed to offer local and global interpretation of what models are choosing as well as to justify whether the model has learnt relevant patterns instead of spurious artefacts [32], [33], [34].

In this larger picture, it is evident that an intrusion detection framework utilizing the advantages of contemporary machine learning, which works with a realistic dataset such as CIC-IDS2017, is needed to deal with the issues of imbalance and excessive dimensionality and includes explainability in an organized manner. This thesis satisfies this requirement by creating and analyzing a explainable machine learning-based IDS framework which integrates feature selection, ensemble and neural models, and SHAP-based analysis into a single and reproducible pipeline.

1.2 Problem Statements

Although there has been significant advancements in intrusion detection research, current networks are yet to have explainable machine learning-based IDS solutions that are widely adopted and can be trusted to detect various types of attacks in real-life situations. Most of the systems that are currently in place are either very coarse (typically they only distinguish between benign and malicious traffic) or they are black boxes whose internal reasoning is dominated by non-transparency[19], [35]. More specifically, in the popular CIC-IDS2017 dataset, there exists no unified, common framework that integrates effective pre-processing,

multi-class modelling, class imbalance treatment and systematic application of explainable AI into a single and coherent framework. The main issue explored in this thesis is thus that there is no explainable machine learning intrusion detection framework of CIC-IDS2017 that is capable of performing fine-grained multi-attack classification, effectively dealing with imbalanced data and can give clear, security-relevant explanations of its conclusions.

This issue is also very topical in terms of practice since the most important services, including online banking, electronic health records, e-learning websites and government portals are more and more offered via the internet and intra-corporate networks[36], [37]. An effective breach of these systems may result in stolen credentials, sensitive information corruption or leakage, financial fraud, denial of important services and a serious reputational damage. When organisations install IDS and other tools in an effort to scan their networks, the systems often prove ineffective in differentiating minority forms of attacks, they tend to produce high amounts of false alarms, or they fail to present clear explanations of their alerts[38], [39]. Moreover, regulatory measures and internal governance rules are changing, which has increased the focus on the necessity of explainable and auditable decisions made by AI, especially in areas with security, privacy and trust implications[40]. Such IDS cannot justify their types of classifications, and therefore, may not be easily adopted, validated or defended in those situations.

The implications of the issue can be severe in case the issue is not addressed. Machine learning-based IDS can seem to be very precise when measured using aggregate measures on unbalanced data, and it still is not capable of identifying rare, but critical, intrusion, like infiltration or the targeted web-based attacks, which can give attackers permanent access to internal resources [41]. Security analysts can become overwhelmed with floods of alerts the origins of which they cannot completely comprehend and grow less confident in automated systems and more vulnerable to missing real threats in false positives. The adoption of AI-based detection tools may be slowed down or its use limited through questionable reliability and accountability of these tools by decision makers and regulators. In the long-run these vulnerabilities can be exploited by attackers creating the campaigns which specifically exploit the vulnerabilities of existing IDS models resulting in breaches that would otherwise have been avoided through more resilient and interpretable detection systems.

The current literature on CIC-IDS2017 and other related datasets demonstrates that there are some gaps that lead to it. A large body of research still makes intrusion detection a binary

classification problem (benign vs attack) or extremely broad attack categories, which restricts the practical utility of their results since incident responders have to be aware of which particular type of attack is being attempted so that they can make decisions about relevant and suitable countermeasures[35]. Figure 1 displays a substantial variation in the way researchers preprocess the data: they might choose various subsets of days, reassign various labels, treat missing values differently and choose features in dissimilar ways [22]. This not-so-good standardisation complicates the reproduction of results, or even the fair comparison of models. The extreme level of classes imbalance in CIC-IDS2017 implies that minority classes infiltration and certain web attacks are underrepresented; however, some works disregard this imbalance, or discuss it online, which leads to models with high overall accuracy and low recall and F1-scores of critical rare attacks[27]. Lastly, as ensemble and deep models are common models used and can deliver high predictive accuracy, relatively few studies consider structured explainability models, including SHAP, to demonstrate how features are used to predict various attack types and how to justify the behaviour of models in an open manner[40], [42].

Considering these weaknesses, this thesis develops a list of research questions that outline the area and focus of the research:

1. How do we pre-process, regroup, and subset the CIC-IDS2017 data to enable effective, multi-class intrusion detection and yet maintain a realistic model of the behaviours of different attacks?

2. The question is, what are the comparative performances of different machine learning models, that is, XGBoost and CatBoost, Artificial Neural Network, Multi-layer Perceptron, with regards to accuracy, precision, recall, F1-score and AUC-ROC across majority and minority attack classes when they are trained and evaluated through a single experimental pipeline?

3. What can explainable AI methods, specifically SHAP-based analysis be added to the best-performing model to give explicit, security-related information on what characteristics lead the model to make judgments on various types of network traffic?

These questions and this issue are worked out considering various audiences. Among machine learning and cybersecurity researchers and postgraduates, they point at methodological challenges of reproducibility, fair comparison, multi-class evaluation and incorporation of

explainability in IDS research design. Practitioners in the network defence business offer more practical requirements: efficient identification of various attack variables, great performance in small classes, sensible computing demands and explanations to assist analysts to comprehend and accept the model results. By maintaining both sides of the coin, the thesis seeks to fill the commercialization gap between the academic implementation and operational implementation.

All these considerations combined result in a refined research problem statement. The key point is not only to achieve high accuracy on CIC-IDS2017 but to design and test a sensible, explainable machine-based intrusion detection model, which: (i) is able to classify benign traffic and a variety of attack types in detail; (ii) does so without being severely affected by the class imbalance; (iii) is computationally viable with large-scale traffic; and (iv) is able to explain its predictions in a clear, understandable way using SHAP-based analysis. The proposed solution to this dilemma is likely to not only serve the scientific knowledge of IDS design, but also the actual creation of more reliable, useful and deployable intrusion detection systems.

1.3 Aim and Objectives

The primary aim of this study is to design, develop, and evaluate an explainable machine-learning-based intrusion detection framework using the CIC-IDS2017 dataset that achieves high detection performance across multiple attack types, handles class imbalance, remains computationally efficient, and provides transparent explanations suitable for real-world cybersecurity deployment.

Objectives

I. To implement Random Forest-based feature-importance analysis to identify and rank the most relevant network features influencing intrusion detection.
II. To train and evaluate four algorithms, XGBoost,CatBoost, ANN, and MLP within a unified experimental pipeline, ensuring fair and reproducible comparisons.
III. To perform hyperparameter optimisation using Randomised Search Cross-Validation (RandomisedSearchCV) for achieving optimal model generalisation.
IV. To evaluate models comprehensively using accuracy, precision, recall, F1-score, and AUC metrics to assess overall detection performance and apply SHAP explainability methods to interpret model outputs, visualise feature contributions, and enhance analyst understanding.

1.4 Research Scope

The scope of this research determines the limits according to which this research will be conducted and what lies therein and what is left out deliberately. Well defining the scope is also necessary to make sure that the objectives are achievable, the methodology is not too difficult and findings are not interpreted in too broad terms. This paper is concerned with the construction of a explainable machine learning based intrusion detection model with a current benchmark dataset and is not intended to exhaust all the potential intrusion detection model, algorithm, and deployment environment. Rather, it focuses on a clear set of information, models and methods that enable a profound and methodical exploration of performance and explainability.

I. **Dataset Scope:** The study is limited to the dataset of CIC-IDS2017 that contains flow-level logs of malicious and benign network traffic on several days. The initial dataset has a multitude of attack labels which in this study are pooled into six high level groups namely: Benign, DoS, Botnet/Scan, Brute Force, Web Attack and Infiltration. The rationale behind this regrouping is to strike a balance between realism and practicality; the grouping retains a significant distinction between major types of attacks without creating an overly fragmented label space which would be hard to model in a reliable way. Other intrusion detection data and live traffic captures are not factored in the study. Consequently, any inferences made are all within the particular context of CIC-IDS2017 and can be interpreted to mean what is possible in a realistic but controlled dataset and not all network environments.

II. **Modelling Scope:** This thesis is based on a limited scope of modelling of supervised learning with four algorithms, including XGBoost, CatBoost, ANN, and MLP. These models are chosen to cover two large families of models such as gradient-boosted tree ensembles and fully connected neural networks that are commonly utilized in intrusion detection studies. This work does not implement or compare other categories of models, including SVM, KNN, CNN or RNN, and purely unsupervised methods of anomaly detection. The paper concentrates on tabular and flow-level features available by the dataset and does not involve extraction of deep features of raw packets. It is hoped to study, in more detail, the behaviour of these four representative models within the

context of a common experimental system, instead of conducting a general benchmark of all possible algorithms.

III. **Scope of Pre-processing and Feature Engineering:** Regarding the pre-processing and feature engineering, the study uses a well-organized but minimal set of steps. These involve the combination of the many CSV files that constitute the data, erasing or replacing gaps and infinity, encoding nominal codes into a numerical pattern, and stabilising the model training by means of scaling continuous data. Random Forest based feature-importance ranking is conducted to perform feature selection. More advanced or domain-specific feature engineering as protocol-aware parsing, payload inspection, or handcrafted time-series features is all out of scope.

IV. **Scope of Evaluation and Explainability:** The model of the proposed framework is tested under the offline experimental conditions. The standard metrics of classification, such as accuracy, precision, recall, F1-score, and AUC-ROC are used to assess performance and are analysed in detail both overall and at a level of classes. The visual representation in the form of confusion matrices, ROC curves and radar charts is provided to have a better idea of the advantages and disadvantages of each model in relation to the various categories of attacks. The question of explainability is tackled by implementation of SHAP on the most efficient model. Explainability is restricted to the post-hoc examination of the model results, and concentrates on the importance of global features and patterns by classes.

V. **Scope of deployment and operation:** Deployment wise, this study is theoretical but not fully functioning. Experiments are done on stored data under controlled conditions and the models are not incorporated into a live network or a production-based intrusion detection platform. The problems addressed at a conceptual level but not implemented include real-time data ingestion, system latency, alert fatigue, integration with other existing SIEM systems, and continuous model retraining. Hardware constraints are primarily considered in order to make sure that the offered framework can be computed on a sufficiently powerful hardware (typical research or enterprise) but large-scale distributed deployment, edge computing, and cloud-native systems are clearly beyond the scope of feasibility. Therefore, the results of the current research should be considered as a starting point of future research, which will modify and implement the suggested framework to the actual setting.

1.5 Research Contribution

The present study has a number of interconnected contributions to the state of intrusion detection because it unites the latest machine learning methods, an experimental benchmark dataset, and explainable artificial intelligence into one, unified framework. To begin with, the research does not apply binary classification as it is widely used, but rather runs multi-attack classification that is fine-grained on the CIC-IDS2017 dataset. The work gives a more operationally useful perspective on intrusions by re-categorizing the original labels in six meaningful ones, namely Benign, DoS, Botnet/Scan, Brute Force, Web Attack, and Infiltration. Not only do security analysts and system administrators know that an attack is in progress they also have an idea of what type of attack is taking place, which is vital in prioritising response and crafting mitigation approaches. This granularity is practical given the fact that research only distinguishes between normal and abnormal traffic.

I. **Standardized and Reproducible Evaluation Pipeline:** In this study used pre-processing, feature selection, data-splits, class-weighting and evaluation pipeline among models. This removes methodological errors that were inherent in previous studies and allows an objective and impartial comparison of machine-learning methods.

II. **Comparing Model Performance Equally to Ensemble and Neural Network models:** When all models are run through identical experimental conditions, differences in model performance could be explained by only model architecture and not by some latent differences in data processing. This creates a stable standard of multiple attack intrusion detection.

III. **Better Interpretability via Explainable AI (XAI):** SHAP is used to explain the XGBoost model predictions. The analysis offers both the global and local description, and explains how the particular network traffic properties can affect the benign or malicious categories. This enhances transparency, confidence, and decision making among the cybersecurity practitioners.

IV. **An Applied and Scalable IDS Design Framework:** The paper provides a foundation of how to design scalable and explainable intrusion detection systems by contrasting the performance of ensemble and neural network models across attack types

particularly minority-class attacks whilst taking into account the computational efficiency.

1.6 Research Activities

The research operations in this study were laid out in a clear flow that slowly transitioned between the knowledge of what was known to a developing a new model and lastly testing its functionality in a critical and understandable manner. The stages were based on the previous ones in a way that the final results were not merely a collection of experimental findings, but the result of a well-planned and organized research. Generally, the work may be subdivided into three large-scale activities, which are the review of the pertinent literature, design and implementation of the proposed intrusion detection framework, and in-depth benchmark and analysis of the resulting models.

1.6.1 Review the Literature

The initial research undertaking was aimed at coming up with a concrete conceptual and technical grounding of the research via a thorough analysis of the available literature. This entailed considering how intrusion detection systems have been developed over time; between the old signature and anomaly-based systems and the newer machine learning and deep learning systems. The research conducted by the study on how the IDS technologies have evolved over the years has assisted in determining the strengths and weaknesses of the various detection paradigms and the reasons why the data driven approaches have kept gaining prominence in cybersecurity today. Classical machine learning models and ensemble models, as well as neural network architectures that applied to intrusion, were also covered in the literature review.

Besides models, the literature activity discussed methodological factors including dataset construction, pre-processing the data, feature selection, managing the class-imbalance and evaluation performance. Particular focus was placed on the discourse of the shortcomings of older benchmark data, the impediments of skewed distributions of attacks, and the rise of explainable AI methods, including SHAP, in making IDS more transparent. In the process, this research found some obvious gaps: the absence of unified pipelines, fine-grained classification of attacks, and insufficient attention to minority attacks, as well as weak integration of

interpretability. The specified insights served as the direct influence on the design decisions and priorities of the framework suggested.

1.6.2 Design and Implementation

The second significant activity was the design and implementation of the proposed machine learning based intrusion detection framework. This started with the technical preparation of CIC-IDS2017 dataset where various raw CSV files of various capture days were merged into one, consistent dataset. Several pre-processing operations were then implemented, such as the processing of missing and infinite values, coding of attack labels into six meaningful groups, and scaling of the numerical features in order to effectively process them by the chosen models. To reduce dimensionality and make the learning problem easier, a feature selection step was added to use the features of the network flows, which are most informative in separating between the benign and malicious traffic, which are figured out using a random forest-based feature selection step. The fundamental framework models were put into practice after the data pipeline was established. It was decided to use four algorithms representing two strong branches of methods: gradient-boosted decision trees (XGBoost and CatBoost) and fully connected neural networks (ANN and MLP). An integrated experimental setting was then designed such that all the models are trained and tested in the same setting. This involved the determination of a shared stratified train-validation-test split, class-weighting. To reduce the effects of imbalance, and randomized hyperparameter search with help of RandomizedSearchCV.

1.6.3 Benchmarking and Analysis

The third research exercise was aimed at benchmarking the performance of the applied models and making a close analysis of their behaviour. Each model was tested on the test data after training on the training data in a number of complementary performance measures. This was not the only way that these metrics were analyzed, but on a case-by-case basis, in order to be able to make the strengths and weaknesses of any model used to tackle certain types of attack known. Confusion matrices, ROC curves, and radar charts were used to help give an intuitive view of which models best differentiate between benign traffic and each type of attack, and the areas of most probable misclassification.

On the basis of these results, a more interpretable analysis was made on the best performing model. This model was applied to SHAP to examine the effects of individual features on its

predictions both on a global and per category of attack level. The analysis provided feature-importance scores and visual descriptions which demonstrated that various feature values shifted a prediction to benign or malicious categories. These descriptions were then explained relative to the previous literature review and domain information regarding network traffic and attack behaviour. Lastly, the benchmarking and analysis exercise ended by comparing the empirical results with the initial research problem, and how the framework can be used to fill the gaps identified, the limitations that still exist, and how the work can be done in the future. By this conglomeration of quantitative assessment and qualitative extraction, the research practices as a whole assurance was not only correct, but also sensible and insightful in the overarching market of intrusion detection.

1.7 Structure of the Thesis

To show the reader the way through the research process in a coherent and well-structured manner, this thesis is divided into a sequence of chapters that lead to a general context, specific practices, findings, and concluding comments and thoughts. The chapters are constructed in a way that leads to the next that by the conclusion, one is in a position to grasp the entire contribution of the work and how it is practically relevant towards intrusion detection. The structure is made simple to follow even in cases where the reader is not very conversant with all the aspects of machine learning or cybersecurity.

In chapter 1 the general background and motivation of the study is provided as to why intrusion detection is still a critical issue in today, highly connected environment. It explains the main issue that will be considered in this study, the purpose and the specific purpose, and the scope wherein the work will be carried out. The key contributions of the research are also presented in the chapter and the chronology of the research activities undertaken. By doing it, it gives a high-level map of what the thesis aims to accomplish and why the fixed direction is significant.

Chapter 2 contains an elaborate literature review which places the study in the context of research done on intrusion detection systems. It follows a trail of progression of traditional signature based and anomaly based IDS into more modern machine learning and deep learning techniques, and evaluates the strengths and weaknesses of each. Other datasets where databases are frequently used, especially CIC-IDS2017, are also reviewed in the chapter, and the critical methodological problems discussed, including but not limited to data pre-processing, feature selection, class imbalance, evaluation metrics, and explainable AI. This chapter by pointing

out the gaps and limitations of the earlier work creates a clear rationale of the framework that is being created in the rest of the thesis.

Chapter 3 is structured in a clear and systematic way of explaining how this study will be done. It describes the features of the CIC-IDS2017 dataset, the pre-processing procedure to clean and prepare the data, and feature selection procedure using the Random Forest importance. The chapter goes on to describe the design of the proposed intrusion detection framework, configuration of XGBoost, CatBoost, Artificial. The hyperparameter tuning strategy, the Multi-Layer Perceptron models, and neural Network, and the strategy used to address the issue of class imbalance. Lastly, it gives the metrics of the evaluation and the SHAP-based explainability process through which the model performance and interpretability are analysed.

Chapter 4 records and addresses the results of the experiment through the framework implemented. It provides the results of the feature selection procedure, the performance of each model on the various categories of attacks, and visualisation of the results in confusion matrices, ROC curves, and radar charts. The next aspect of the chapter is concerned with the most efficient model and discussed in greater detail with the help of SHAP, indicating the characteristics that have the strongest impact and how they impact predictions of various traffic. The results are discussed according to the purposes of the research and the available literature, and the strengths, weaknesses, and opportunities of the offered method are also critically analyzed.

Chapter 5 is the final part of the thesis that summarises the key findings and the personal reflection on how the research objectives were fulfilled. It restates the main contributions of the work, specifically when it comes to multi-attack classification, unified evaluation, and explainable intrusion detection. The practical relevance of the work to the real-life implementation of IDS is also described in the chapter and to the potential future research directions, i.e. extending the framework to other data, using more models or investigating the real time and operational factors. These chapters are collectively a full and consistent description of the design, testing and analysis of a comprehensible machine learning-driven intrusion detection model.

CHAPTER 2

LITERATURE REVIEW

Overview

The chapter surveys the available literature on the intrusion detection systems (IDS), machine learning (ML), deep learning (DL), benchmark datasets, data pre-processing, feature selection, class imbalance, evaluation metrics, and explainable artificial intelligence (XAI). The objective of the literature search is two-fold. To begin with, it offers the theoretical and technical contexts of what is needed to comprehend the design decisions involved in this thesis. Second, it outlines certain gaps and shortcomings of previous studies that will be driving the creation of the proposed explainable ML-based IDS framework. The paper starts with the history of IDS development and transitions to more recent data-driven methods based on ML and DL. It then dwells upon the properties and drawbacks of current benchmark datasets, especially CIC-IDS2017 and discusses methodological problems, including pre-processing and feature selection. The chapter then explores the challenge of class imbalance and the growing importance of interpretability in IDS. Finally, it positions the present study in relation to this body of work and summarises the key insights that guide the remainder of the thesis.

2.1 Evolution of Intrusion Detection Systems

The concept of intrusion detection was initially coined in the early 1980s when scholars realised that the preventive strategy against intrusion, including the use of firewalls, passwords, and password policies, could not help secure networks. The design of IDS was based on the conceptualisation of an audit trail by Anderson (1980) to identify the anomaly in user behaviour. Initial applications were signature-based systems which were based on predefined rules to match a known pattern of malicious activity. Although they were useful in detecting the old attacks, they were unable to detect altered or new attacks[34].

2.2 Anomaly-Based IDS and Emergence of Machine Learning

Researchers came up with anomaly-based IDS, which establishes a statistical profile of normal behaviour and indicates abnormalities as possible intrusions to address these drawbacks. This method made it easier to detect the zero-day attacks but posed difficulties in the definition of normality and high false-positive rates[43]. This became feasible as the computing power and availability of data grew and machine learning (ML) techniques were now considered as alternative methods to detecting patterns that could be used by the IDS instead of depending on expert-created signatures[44].

2.3 Traditional IDS Models: Misuse vs Anomaly Detection

The traditional IDS systems employ either misuse detection or anomaly detection models. Misuse detection works by identifying traffic using of known attack signature, which is highly accurate but not flexible. Anomaly detection on the other hand model's normal behaviour and identifies outliers and thus with a higher recall of novel attacks at the cost of higher false alarms[45].

2.4 Machine Learning Paradigm for Intrusion Detection

Machine learning was a shift of paradigm because it offered algorithms that enabled generalisation using past data. The decision trees, the NB, and SVM are supervised learning models that are utilized to classify the network flows as either benign or malicious, using the labelled training data. Label less anomalies are detected by unsupervised learning strategies, such as clustering and autoencoders. The methods of semi-supervised and ensemble methods are integrative of the two paradigms. It is always shown in comparative analyses that ensemble approaches perform better than individual classifiers through the utilization of several weak learners[45].

2.5 Supervised Ensemble Models for IDS

The use of supervised models has become common due to their predictive ability and comprehensibility. Random Forest and Decision Trees classifiers are transparent and have

feature importance measures. Specifically, Random Forest is appreciated due to the resistance to overfitting and the fact that it can process large datasets (high-dimensional). XGBoost and CatBoost gradient boosting techniques are also effective predictors that enhance the accuracy of predicting the model by optimising the gradient at each stage[46].

2.6 Unsupervised and Hybrid IDS Techniques

Unsupervised techniques would prove to be beneficial in cases where the labelled data is limited. The K-Means, DBSCAN and self-organising maps are techniques used to identify new patterns by grouping similar behaviours. They can however categorize harmless anomalies as an attack. Hybrid frameworks integrate supervised and unsupervised elements, i.e. using clustering to filter anomalies at the first stage and then classification, to be more adaptable and reduce false-positive rate[47].

2.7 Deep Learning Architectures in Intrusion Detection

Deep learning (DL) is an extension of machine learning, whereby multi-layer neural networks are used to automatically derive hierarchical representations of the data. Some of the first DL architectures to be used on IDS are ANN and MLP. CNN are able to capture the spatial relations of packet characteristics, and RNN and LSTM networks are able to capture the temporal associations in traffic sequences[48].

2.8 Performance and Trade-offs of Deep Learning vs Classical ML

It has been demonstrated by comparative analyses that the DL architectures tend to perform better than classical ML models when trained with large enough datasets. As an example, a study by Abbaspour et al. (2020) has shown that CNN-LSTM hybrids are better at recalling minority classes with respect to IoT network data[49]. Nevertheless, DL approaches have increased computational needs and have been accused of being less transparent. This has been one of the focal points of research dilemma including trade-off between accuracy and interpretability [50].

2.9 Overview of IDS Datasets and Pre-processing

Testing IDS models requires representative and good datasets. Older benchmarks like DARPA 1998, KDD Cup 1999, and NSL-KDD were initially used to test but were no longer relevant because of synthetic attack conditions and lack of feature variations. These data samples could not include current traffic trends or recent types of attacks[51]. In order to rectify these shortcomings, the Canadian Institute for Cybersecurity (CIC) released the CIC-IDS 2017 dataset containing over 2.8 million records of benign, multiple malicious (DoS, DDoS, brute-force, infiltration, botnet, and web attacks) categories. Each record has 79 features that include flow, packet, and content features. The realism and scale of the dataset has resulted in it becoming a standard to test the IDS based on ML/DL [52].

Still, CIC-IDS 2017 has its problems: the class imbalance between benign traffic and minority attacks is extremely high, attributes are redundant, and memory consumption is high. CSE-CIC-IDS 2018 data further introduced novel types of attacks and temporal consistency, but failed to address imbalance problems fully[53]. Researchers are still working on perfecting pre-processing and sampling schemes to normalize these datasets to even benchmarking. Raw network traffic data needs a lot of pre-processing in order to make sure that the models are reliable. Typical procedures are data integration, duplicate handling, filling in missing values, and feature scaling. Standardisation (zero-mean, unit-variance) is used to make sure that features equally contribute to distance-based algorithms. Label encoding transforms categorical types of attacks to an numeric format that can be used in ML systems[54].

2.10 Feature Engineering and Selection for IDS

The feature engineering converts raw attributes to meaningful indicators. Flow-based metrics (packet rate, number of bytes), time-based (inter-arrival time), and statistical measures (variance, entropy) can usually help differentiate healthy and unhealthy flows. This dimensionality reduction approach saves time during training and helps avoid overfitting through feature selection, like correlation filtering, Information Gain, or Random Forest importance[55].

Recent papers point to the use of automated feature learning with deep architecture as a complementary approach. Nevertheless, interpretable manual selection of features is also essential to be efficient and easy to interpret[56].

2.11 Class Imbalance and Sampling Strategies

The imbalance of data has a serious impact on the performance of the IDS. In some datasets like the CIC-IDS 2017, more than 80 percent of samples can be benign, and less than 1 percent of samples can be potentially critical attacks, such as infiltration. In the absence of mitigation, the overall accuracy of models is inflated though the recall of the minority classes is poor. Examples of such remedies are oversampling (Synthetic Minority Over-sampling Technique, SMOTE), under sampling and class-weighted loss functions [57].

2.12 Evaluation Metrics for Imbalanced IDS

Measurements of evaluation should thus go beyond the accuracy in order to encompass precision, recall and F1-score, as well as the Area Under the Receiver Operating Characteristic Curve (AUC-ROC). When there is a skewness in classes, the harmonic mean of precision and recall (F1-score) is a balanced measure. The application of several metrics guarantees the holistic evaluation and a viable comparison of the models[58].

2.13 Explainable AI (XAI) for Intrusion Detection

With the introduction of IDS as a constituent of real time security activities, interpretability has taken a front seat. Black-box models Black-box models are particularly problematic in the interpretation of the process of making predictions, such as deep neural networks. Explainable AI (XAI) attempts to address such a shortcoming by offering an understanding of feature contributions and decision paths. SHAP algorithm, which was developed based on the cooperative game theory, is a method of quantifying the contribution of each feature to the model output.

When applied to IDS, SHAP will determine the network elements that most affect the detection of attacks, allowing trust and helping in the investigation of the case. LIME (Local Interpretable Model-Agnostic Explanations) approximates local decision boundaries of single prediction.

The use of XAI in IDS will improve the level of visibility, aid human-in-the-loop decision-making as well as adhere to new ethical AI regulations[59].

2.14 Advanced Trends in Modern IDS

Mechanisms of attention in neural networks are also examined more recently as intrinsic interpretability capabilities, where models are able to indicate key time steps or packets that are important to the classification. There are several comparative studies that compared and contrasted ML and DL practices using contemporary datasets. Research indicates that ensemble (Random Forest, XGBoost, CatBoost) architectures can be trained within shorter timeframes and be more interpretable than deep architectures [60]. On the contrary, DL techniques are better at identifying high-dimensional, fine grained correlations. Recent studies have been extended to domain specific scenarios such as IoT, cloud computing and 5G/6G networks where the limitations of latency, bandwidth and device power require detecting models that are lightweight but intelligent. The IDS deployed at the edges uses federated or distributed learning to share knowledge without centralising sensitive data[61].

The second rising trend is adversarial machine learning whereby the attackers devise inputs in order to cheat models. The inclusion of defensive approaches, such as adversarial training and robust optimisation, are provided in IDS frameworks now[62]. Lastly, explainability and automation meet with AutoML systems based on the search of the best model architecture with interpretability retained - a field where scalable, deployable IDS is becoming increasingly important.

2.15 Positioning of the Present Study

To address these issues, the current study suggests a single, justifiable machine-learning-based IDS pipeline constructed on the CIC-IDS 2017 dataset. This framework incorporates systematic pre-processing steps to maintain the data consistency, uses the Random Forest-based feature selection to reduce the dimensionality, and emphasize the most relevant network characteristics and performs a comparative analysis of XGBoost, CatBoost, ANN and MLP models under the same conditions of the experiment. Furthermore, SHAP-based explainable AI methods are included into the research to explain the contributions of features and to make the decisions of a model more transparent. The proposed approach will provide a common

metric of accuracy, efficiency, and interpretability in one framework and, at minimum, create a reproducible, and scalable standard in intrusion detection research and applicable network security operation

2.16 Chapter Summary

This chapter has reviewed the key strands of literature relevant to the development of an explainable machine learning–based intrusion detection framework. It began by outlining the evolution of IDS from traditional signature-based and anomaly-based systems to more adaptive, data-driven approaches that leverage machine learning and deep learning. The chapter has examined the major strands of literature applicable in the development of explainable machine learning based intrusion detection framework. It has started with the overview of how IDS has evolved as a part of more traditional signature-based and anomaly-based systems to more adaptive, data-driven systems that take advantage of machine learning and deep learning. The talk has drawn attention to the use of ensemble algorithms like the Random Forests, XGBoost and CatBoost as formidable contenders in detecting intrusions on tabular data and in many cases outperform deep learning. Simultaneously, the chapter also discussed the exploration of deep learning architectures, such as ANN, MLP, CNN and LSTM to identify intricate spatial and temporal patterns in network traffic together with their increased computing cost and reduced interpretability.

The literature review then highlighted the significance of realistic benchmark datasets and specifically CIC-IDS2017 and outlined the different issues that come with working with such data, such as noise, high dimensionality and extreme class imbalance. It has addressed typical pre-processing and feature selection alternatives, particularly involving tree-based feature importance and clarified why measures like precision, recall, F1-score and AUC-ROC are important in the appropriate assessment, particularly minority attack classes. Finally, the chapter explored the emerging field of explainable AI in IDS, showing how techniques like SHAP can be used to make powerful models more transparent and trustworthy.

Taken together, these insights define the context and motivation for the work undertaken in this thesis. They show that there is a need for a unified, multi-class, explainable IDS framework that combines strong predictive performance with robust treatment of imbalance and clear interpretability. The next chapter builds on this foundation by describing the methodology used

to design, implement and evaluate such a framework using the CIC-IDS2017 dataset and the selected machine learning models.

CHAPTER 3

METHODOLOGY

Overview

The chapter describes the process through which the research was conducted step by step, starting and ending with raw data and then on to trained, evaluated and interpreted models. The preceding chapters provided the incentive behind the requirement of a machine learning-based intrusion detection framework that can be explained but the current chapter is about the practical aspects of designing and developing the said framework. It starts with an explanation of the contents of CIC-IDS2017 dataset, the organisation of the flow-level records and the labels of the attacks and its reorganisation into six higher-level groups that can be used in multi-class intrusion detection. The chapter then outlines the data pre-processing pipeline, as comprising, the integration of various CSV files, treatment of the missing and infinity values, coding of the attack labels, and the scaling of the numerical features which combine to ready the data to be successfully model trained.

Following the data base, the chapter presents the feature selection strategy which is random forest feature importance. It is done to find the most informative network attributes, dimensionality reduction, and model interpretability. The proposed intrusion detection framework design is then outlined including the implementation of four models XGBoost, CatBoost, ANN, and MLP in a common experimental environment. The chapter explains the generation of stratified train-validation-test splits, class-weighting to alleviate the effect of class imbalance, and a hyperparameter-tuning procedure based on RandomizedSearchCV. Lastly, it outlines the measures of evaluation used to judge performance, and the SHAP-based explainability process used to understand the most effective model. By doing so, this chapter gives a very exhaustive and clear explanation of the methodological decisions that support the results that will be given later in the thesis.

3.1 Methodological Structure

The overall workflow of the proposed IDS framework is illustrated in figure 3.1 below.. It starts with data acquisition with the help of the CIC-IDS2017 dataset and includes a series of pre-processing steps which include data integration, noise elimination, label encoding and feature scaling.

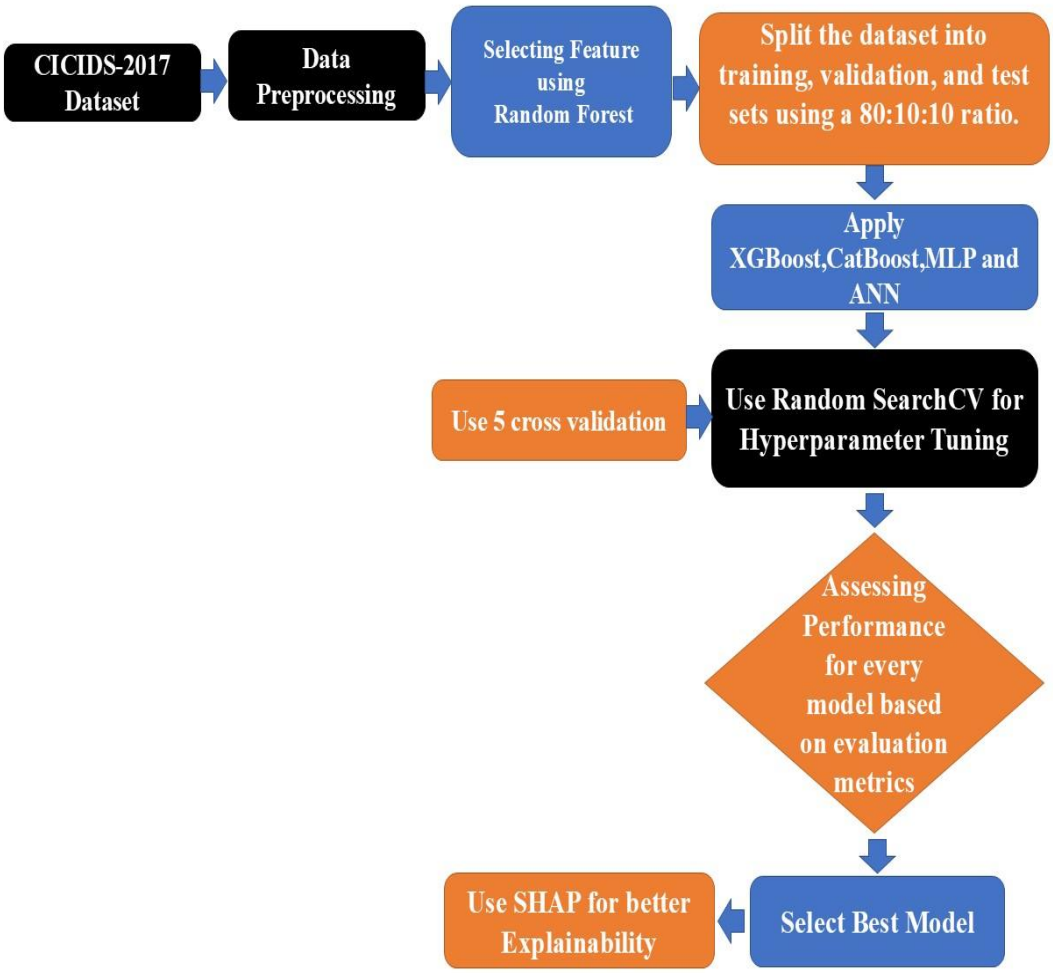


Figure 3.1 Workflow of the Proposed Intrusion Detection Framework

After this, a random forest-based feature importance is used to achieve the most significant attributes. Under a common experiment system, the refined dataset is trained in four models, which consist of XGBoost, CatBoost, ANN and MLP. RandomizedSearchCV is a method of hyperparameter optimization that is used to give the best model generalization. Lastly, the models are analysed based on performance measures such as accuracy, precision, recall, F1-

score, and AUC and SHAP explainability gives interpretability to the model outputs. This workflow unites all the parts of the study in a single and reproducible experiment pipeline.

3.2 Dataset Description

In this paper, we used CICIDS2017 data set. This dataset was created by the Canadian Institute for Cybersecurity and provides a comprehensive collection of network traffic that reflects real-world scenarios, including both benign and malicious activities. It has more than 79 feature set based on packet flows and encompasses different types of attacks including DoS, DDoS, brute force, infiltration, botnet as well as a web-based attack. The dataset is one of the most common testing systems to evaluate intrusion detection system, because of its variety and realistic traffic patterns. We have chosen four exemplary models: MLP, ANN, XGBoost and CatBoost. We compared their performance to evaluate their effectiveness in anomaly- based intrusion detection [59].

CICIDS2017 contains 2,830,743 records in total, consisting of both benign and malicious traffic across 15 labeled categories. The number of samples in each category is shown in Table 3.1. To evaluate model performance under both known and unknown attack scenarios, we divided the dataset into three subsets: Training Set, Test Set, and Validation Set[56].

Table 3.1 Label (Attack Category) count

Label	Count
BENIGN	2270397
DoS Hulk	231073

PortScan	158930
DDoS	128027
DoS GoldenEye	10293
FTP-Patator	7938
SSH-Patator	5897
DoS slowloris	5796
DoS Slowhttptest	5499
Bot	1966
Web Attack– Brute Force	1507
Web Attack– XSS	652
Infiltration	36
Web Attack– SQL Injection	21
Heartbleed	11

Due to limited computational resources and the absence of GPU support in our environment, the number of BENIGN samples was reduced to 500,000. Additionally, the original 15 labels were consolidated into 6 categories, and the remaining labels were renamed and grouped under the Attack category for analytical consistency.

Table 3.2 Update attack category count

Attack category	Count
BENIGN	500000
DOS	380688
Botnet/Scan	160896

Brute Force	13835
Web Attack	2180
Infiltration	47

3.3 Data Pre-processing

Pre-processing of data is an essential process in the development of a sound IDS, since it guarantees the quality, consistency, and readiness of raw network traffic data, to be used in training and evaluation of the model. This paper has used a number of systematic pre-processing tools to pre-process the data to be used in machine learning. To ensure data quality and usability, the following pre-processing steps were applied:

3.3.1 Data Integration

The data was as a number of CSV files that were gathered on 5 days of the week, and each of the files contained network traffic records. All the single files were combined into one dataset in order to establish a unified analytical framework. This integration made it possible to have a complete representation of network behaviors and attack patterns, and therefore consistency, and it reduced the data fragmentation to a minimum.

3.3.2 Duplicate Value

In contrast to most structured data sets, Network traffic data can include multiple entries which are actually a representation of true communication patterns and not redundant data. Thus, in this research, the cases were not eliminated because their presence might possibly alter the actual traffic distributions on the road and worsen the functionality of the anomaly detection models.

3.3.3 Noise Removal

The missing and infinite values were found and their substitution was with a median of respective features to reduce effects of incomplete or noisy records. The reason why median-based imputation was selected is because it is resistant to outliers so that any extreme values of network traffic features did not affect the statistical characteristics of the dataset disproportionately.

3.3.4 Label Encoding

The types of attacks that were categorical were converted to numerical values with the help of the Label Encoder of the Scikit-learn library. This transformation was necessary in order to allow the algorithm of supervised learning to effectively process categorical variables. There were six classes of the dataset: BENIGN, DoS, Botnet/Scan, Brute Force, Web Attack, and Infiltration. The learning models of XGBoost, CatBoost, MLP, and ANN were able to understand and classify network traffic using the encoded attack labels and not by the identifiers as a string. This change allowed making the categorical target variable compatible with the computational needs of the current machine learning models and retaining the semantic distinction among the various types of attacks.

3.3.5 Scaling of Feature

The All of the numerical features were subjected to the StandardScaler method to eliminate differences in scale across continuous variables. The method would take the features back to an average value of zero and standard deviation of one in that way improving convergence of the model and preventing the high intensity features to play too big role in the training process.

3.3.6 Feature Selection Using Random Forest

Random Forest (RF) is an ensemble learning algorithm that builds many decision trees in order to enhance predictive accuracy and evaluate the significance of every feature. It is very popular in feature selection because it can deal with high-dimensional data, non-linear relationships and interactions among features. The training data are randomly sampled (bootstrap sampling) . The importance of a feature xj is computed in terms of its contribution to the minimization of impurity (e.g. Gini index or entropy) of all trees[63]. Mathematical Formulation:

Let T denote the total number of trees in the RF, and N_t denote the number of internal nodes in the tree t. For each node i in tree t, splitting on feature x_j leads to an impurity decrease given by:

$$\Delta I_{i,j,t} = I(p_i) - \left(\frac{n_{left}}{n_i} I(p_{left}) + \frac{n_{right}}{n_i} I(p_{right}) \right) \quad 3.1$$

Where:

- $I(p_i)$: impurity measure (e.g., Gini index or entropy) before the split.
- n_i: number of samples at node i.
- n_{left}, n_{right} : number of samples in left and right child nodes.
- p_{left}, p_{right}: class probability distributions in child nodes.

The feature importance score for the feature x_j across all trees is computed as:

$$FI(x_j) = \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N_t}\Delta I_{i,j,t} \quad 3.1$$

The features are then ranked based on $FI(x_j)$, and the top k features with the highest importance scores are selected for model training.

Procedure:

- Train a RF classifier on the training dataset.
- Compute the feature importance values using impurity decrease (or permutation importance).
- Rank features according to $FI(x_j)$.
- Select the most relevant features (top-k or above a threshold).
- Retrain the model using only the selected features.

3.3.7 Experiment Setup

All experiments were carried out under a standardized set up in order to ensure consistency and reproducibility. The training, validation and tests used the same pre-processing pipelines and hyperparameter settings among all models. The summary of computation environment including hardware and software requirements are listed in Table 3.3.

Table 3.3 Experimental Setup and System Configuration

System Configuration	Properties

Operating System	Windows 11
Programming Language	Python 3.10
CPU	Intel® Core™ Processor
RAM	16GB
GPU	NVIDIA Google T4
Framework	TensorFlow v2.11
IDE	Kaggle

3.3.8 Handling Class Imbalance

Since the ratio of classes was skewed, a mitigation of the class imbalance was an important involved measure in this research. To solve this problem, the class weighting method was utilized in a number of algorithms, such as XGBoost, CatBoost, ANN, and MLP models. This has the benefit of ensuring that the samples of minority classes are given greater weight, thus penalizing misclassification when training. The models were motivated to perform well in all categories by modifying the loss function to focus on underrepresented classes. Class weighting works by assigning a numerical weight to each class based on its frequency, where smaller classes receive larger weights and larger classes receive smaller weights. This adjustment influences the optimization process so that errors in minority classes contribute more to the total loss, leading to a more balanced and fairer model performance across all categories[64].

3.3.9 Data Splitting Strategy

The data were split into subsets with a stratified split, with the balance of classes being the same in all subsets. In particular, the train data (80%), 10% to validate, and the rest 10% for testing. This stratified partitioning is used to ensure that all the subsets remain representative of all the classes and thus the sampling bias is minimized and the results of the model performance are likely to be more reliable and generalized.

3.4 Model development

The dataset used in this study was around 1.8 GB, which was very difficult to compute using the conventional machine learning models. Such models were usually incapable of processing the dataset effectively and in some cases, led to crashing of the system as a result of memory overload. In order to eliminate this weakness, machine learning and deep learning architectures were used with acceleration on a GPU, among them being the XGBoost, CatBoost, MLP, and the ANN. The usage of the frameworks based on the use of GPUs boosted the computational efficiency, shortened training duration, and allowed to operate the large-scale data with enhanced stability and performance.

3.5 Machine Learning

3.5.1 XGBoost

XGBoost (Extreme Gradient Boosting) is a highly popular algorithm that is based on boosting because it is highly efficient and accurate. It presents improved regularization methods that are useful in avoiding overfitting and it is especially effective when working with large-scale data. XGBoost as well is compatible with parallel processing and distributed computing, which is much faster than the traditional gradient boosting with respect to training. It is highly preferred in practical machine learning scenarios and events due to its capability to deal with sparse data and blank values [65].

3.5.2 CatBoost

CatBoost is a gradient boosting library created by Yandex, with specific features of working with categorical variables. It contains new techniques like ordered boosting that can be used to minimize prediction shift and overfitting. It is very easy to use unlike most other boosting algorithms, as it takes very little pre-processing of categorical data. CatBoost is also speedy and scaled to find competitive accuracy on huge and intricate data sets without troublesome with messy information [66].

3.5.3 Multilayer Perceptron (MLP)

A nonlinear feedforward neural network involving the use of nonlinear activation functions formed in several layers as nodes (MLP) was developed as the prediction task deep learning

classifier. The MLP structure has an input layer which is equal to the count of chosen features, and then, there are several fully connected hidden layers made up of nonlinear ReLU activation functions with a last output layer that can classify. It has an ability to learn structured data in form of nonlinear relationships and intricate patterns hence it is very effective with multi-class classification problems.. The model was trained using backpropagation to minimize the loss function, optimized through algorithms such as Adam or SGD. Regularization techniques like dropout and early stopping were applied to enhance generalization and prevent overfitting. In Figure 3.2 shows multimayer perception structure enables the network to capture intricate dependencies between input features and target outputs, allowing accurate distinction among categories such as BENIGN, DoS, Botnet/Scan, Brute Force, Web Attack, and InfiltrationInfiltration [67].

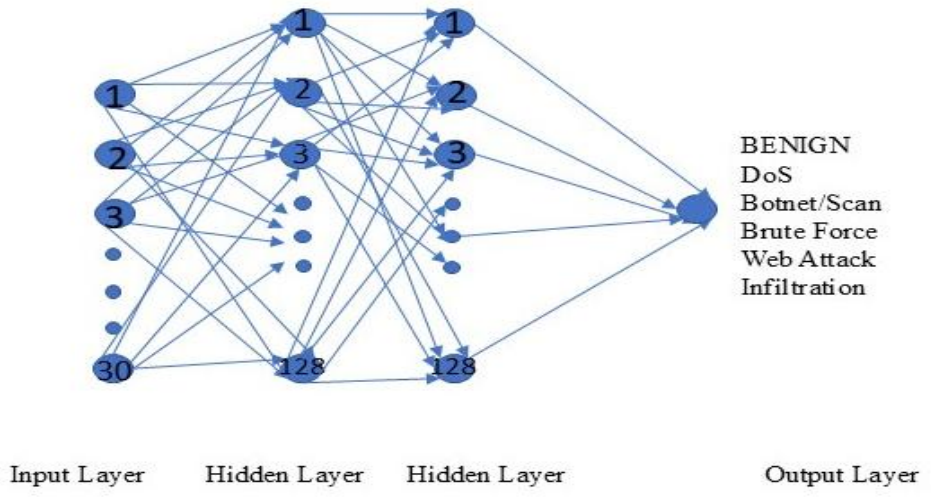


Figure 3.2 Architecture of the Multilayer Perceptron (MLP)

3.5.4 Artificial Neural Network

A baseline deep learning classifier was a model of an ANN, used to carry out the prediction task. The ANN architecture was designed with an input layer corresponding to the number of selected features, followed by one or more fully connected hidden layers (each with a specified number of nodes) using ReLU activation functions, and a final output layer configured for binary classification. Adam optimizer and the binary cross-entropy loss were used in the training of the model. Dropout, early stopping, and learning rate were tested to permit a higher generalization and avoid overfitting. The ANN is a non-linear flexion modeling framework,

and thus is suitable to be used in modeling complex patterns in tabular data in the context of healthcare. In Figure 3.3 shows structure of the ANN, as illustrated in the figure, comprises an input layer that receives multiple numerical features, hidden layers that process these features through weighted connections and non-linear transformations, and an output layer that generates the final classification. This architecture allows the network to effectively learn intricate patterns between input features and target outcomes, enabling accurate classification between different categories such as normal and abnormal conditions or distinct disease [68].

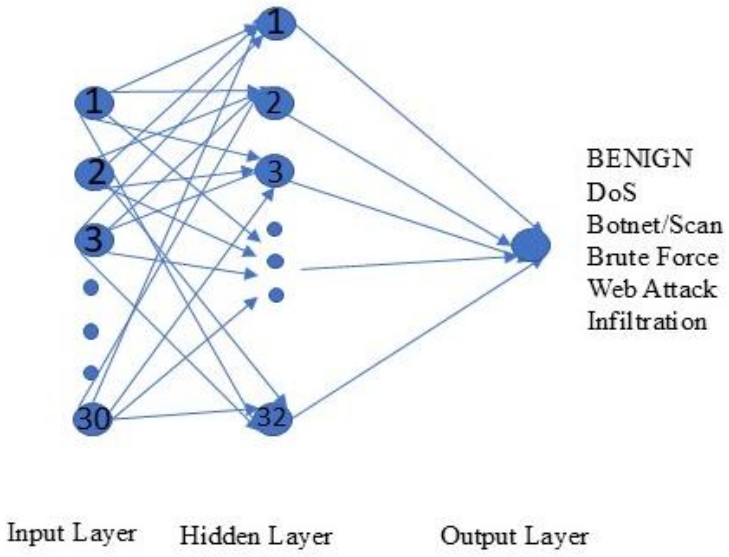


Figure 3.3 Architecture of the Artificial Neural Network (ANN)

3.6 Hyperparameter Tuning

3.6.1 RandomizedSearchCV

Randomized Search Cross-Validation (RandomizedSearchCV) was applied to optimize the hyperparameters of the applied models in order to achieve a better performance and generalization at the lowest computational cost. RandomizedSearchCV is also efficient because unlike exhaustive grid search, it randomly samples a defined set of hyperparameter values. Tuning was done using 5-fold cross-validation strategy to guarantee robustness and minimize overfitting[69].

Table 3.4 Hyperparameter tuning ranges for each model using Randomized Search Cross Validation

Model	Hyperparameter	Search Range/Value
XGBoost	Subsample	0.8
	reg_lambda	2
	Reg_alpha	0.5
	Estimators	300
	Min_child weight	1
	Max Depth	15
	Learning rate	0.05
	Gamma	0
	colsample_bytree	0.6
CatBoost	Random strength	1
	Learning rate	0.1
	L2 leaf reg	7
	Iterations	800
	Grow ploicy	Depthwise
	Depth	10
	Border count	128
	Bagging tempareture	0
ANN	Number of layers	1
	Number of neurons	32
	Dropout rate	0.2
	Learning rate	0.001
	Batch size	32
	Epochs	20
	Optimizer	Adam
MLP	Number of layers	2
	Number of neurons	128
	Dropout rate	0.3
	Learning rate	0.0005
	Batch size	32
	Epochs	20
	Optimizer	Adam

3.7 Performance Metrics

3.7.1 Accuracy

The accuracy of any predictive method is the foundation for assessing its performance in machine learning. It basically computes the proportion of correctly predicted overall data points. The accuracy score interpreted as the maximum possible accuracy is 1.0, while the minimum possible accuracy is 0.0. It is simple to calculate by dividing the number of correctly predicted by the total of projections[70]. Also, it can be expressed as,

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad 3.2$$

3.7.2 Precision

The Precision focuses on the quality of the model's positive predictions. It tells us how many of the "positive" predictions were actually correct. It can minimized false positive like fraud detection [71]. The formula of precision,

$$Precision = \frac{TP}{TP + FP} \quad 3.3$$

3.7.3 Recall

Recall is a performance measure that is used to determine the capacity of a classifier to recognize all relevant instances. It is an expression of the number of true positives divided by the total true positives and false negatives, which points to the completion of the model.

$$Recall = \frac{TP}{TP + FN} \quad 3.4$$

3.7.4 F1 Score

The F1 score is the harmonic mean of precision (positive predictive value) and recall (sensitivity) in binary classification, serving as a single summary measure of a classifier's performance, particularly useful when the prevalence of the positive class is low[72].

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad 3.5$$

3.7.5 AUC (Area Under the ROC Curve)

Area Under the Curve (AUC) evaluates a binary classifier's ability to distinguish between positive and negative classes. It is computed as the area under the Receiver Operating Characteristic (ROC) curve, which plots the True Positive Rate (TPR) versus the False Positive Rate (FPR). A higher AUC indicates better discriminative capability[73].

$$AUC = \int_0^1 TPR\left(FRP^{-1}(x)\right)dx \quad 3.6$$

Where

$TPR = \frac{TP}{TP+FN}$, $FPR = \frac{FP}{FP+TN}$

The above equations are constructed for all the classifiers that consist of True positive (TP), True negative (TN), False positive (FP), and False negative (FN)

3.8 Explainable AI (XAI) Methods

3.8.1 SHAP

SHAP was used to explain predictions of the machine learning models, attribute the results of such predictions to the contribution of single features. SHAP is a graded system that uses the cooperative game theory by assigning individual features with the value of the importance of a given prediction. It gives stable and locally correct descriptions by calculating the average marginal contribution of each feature, over the entire feature combinations. The Shapley value is a fair profit allocation among many stakeholders, depending on their contribution[74]. The shapely value is defined as:

$$\Phi(x_i) = \sum_{\substack{S \subseteq \{1,2,\ldots,K\} \\ i}} \frac{|S|!\,(K - |S| - 1)!}{K!} [f_x(S \cup \{i\}) - f_x(S)] \qquad 3.7$$

Where:

- $K =$ Total number of features.
- $S =$any subset of features that doesn't contain i.
- $|S| =$cardinality of a subset S.
- $\frac{|S|!(K-|S|-1)!}{K!}$= The weight (probability) assigned to the subset S. It equals the fraction of orderings where features in Scome before i and the rest after.
- $f_x(S)$= model prediction when only the features in S are present/known (in practice, this is estimated by marginalizing or using a background dataset).
- $f_x(S \cup \{i\}) - f_x(S) =$ is the marginal contribution of feature iii when added to the coalition S.
- $\Phi(x_i) =$ Is the Shapley value the fair contribution of feature iii to the prediction, for instance x.

3.9 Chapter Summary

This chapter has outlined the methodology support of the study by explaining how the data, models and evaluation processes were well prepared and structured. Firstly we Described the CIC-IDS2017 dataset and the reasoning behind the restructuring of the original attack labels into six categories that allowed the meaningful multi-class intrusion detection. It subsequently described a programmed pre-processing flow of data, such as integration of files, deletion of missing and infinity data, label coding, and feature scaling, which enabled the input to the models to be homogenous, trustworthy and numerically healthy. The feature selection step based on a Random Forest was included to determine the most topical features of the network and to simplify the process of learning without deteriorating its predictive ability.

Over this ready-made dataset, the chapter outlined the architecture of a single experimental structure where XGBoost, CatBoost, ANN, and MLP models have been trained and tested in identical conditions. The extreme class imbalance in CIC-IDS2017 was handled through stratified splitting and class-weighting, and the hyperparameter tuning process was performed by means of RandomizedSearchCV. The assessment metrics used, accuracy, precision, recall, F1-score and AUC-ROC, made sure that the performance was measured in various ways, and the focus was on the behaviour of the classes. Lastly, the chapter presented an explainability method based on SHAP that shall be used subsequently on the model with the highest performance to determine the effect of various features on its predictions. These methodological elements, collectively, constitute a sound and replicable methodology which underpins the experimental findings and interpretability analyses in the following chapter.

CHAPTER 4

RESULTS AND DISCUSSION

Overview

This chapter shows and analyses the empirical evidence of the proposed explainable machine learning intrusion detection framework. As the chapter above has explained the way in which the models were made, trained and tested, the current chapter is concerned with what the models accomplished and the meaning of the results in terms of intrusion detection. The discussion will start with the analysis of the results of the feature selection, what feature network attributes became the most important and how the chosen set of features influenced the learning process. It then compares the classification performance of the four models; that is; XGBoost, CatBoost, ANN and MLP using various evaluation metrics including accuracy, precision, recall, F1-score and AUC-ROC. Special focus is on the performance of each class within the six traffic types, and special interests are on the majority of the models to identify the minority attacks, including web attacks and infiltration.

After presenting the numerical findings, the chapter then relies on visual aids such as confusion matrices, ROC curves and radar plots to give a more intuitive view of the behaviour of each of the models on the various classes. Such visualisations can be used to identify patterns like the most frequently misclassified attacks, the trade-offs between true positives and false positives across the board, and strength and weaknesses of ensemble models versus neural networks. The chapter then further focuses on the most successful model, using SHAP-based explainability to interpret the effect of each individual feature in its predictions. SHAP analysis on global and class-wise basis is employed to determine the prevailing features, the effect of feature values on predictions of benign and malicious classes, and correlate the patterns with established network attack characteristics. In general, the chapter does not only seek to provide performance numbers, but also to present a critical informed interpretation of the given results in comparison to the research objectives and the bigger literature.

4.1 Feature Selection

It is a visual representation of the top thirty largest features picked with the help of the random forest algorithm according to their contributing classification performance of the system. The below figure is the importance score that should be used to compute the feature importance of the chart and it measures the extent of the ability of a given attribute to predict the model when the variables are benign and malicious network traffic within the CIC-IDS-2017 dataset.

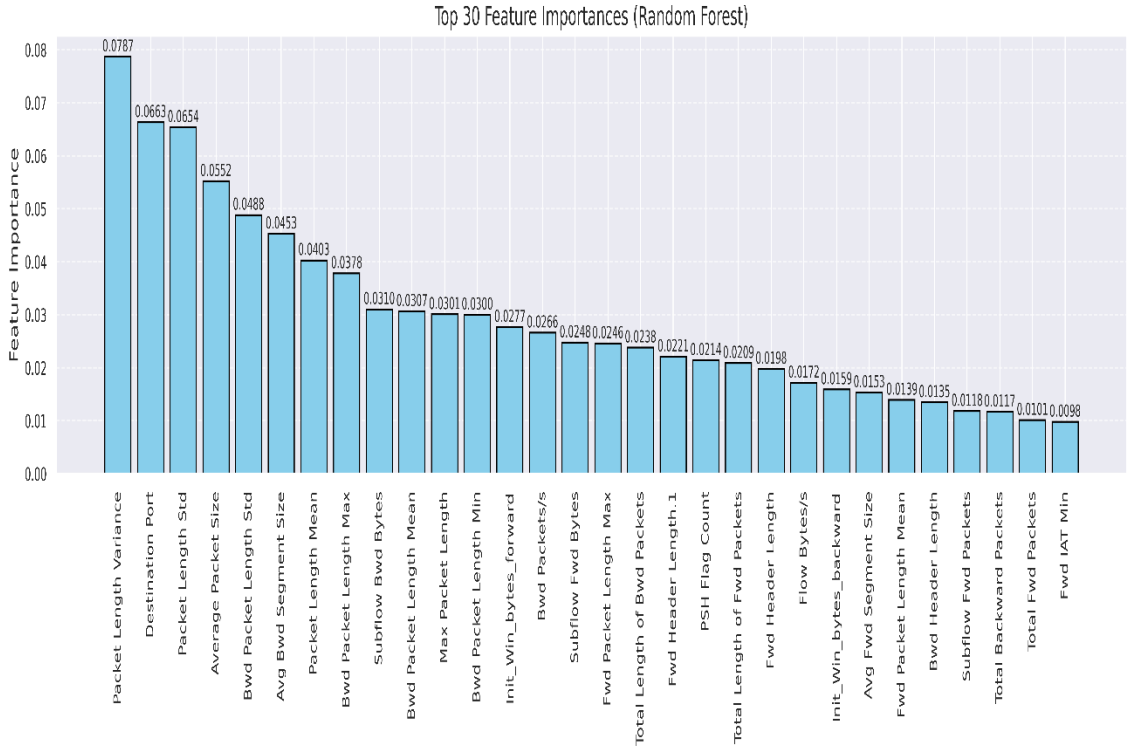


Figure 4.1 Top 30 Most Influential Attributes Contributing to Intrusion Using the Random Forest Algorithm

The most significant relative importance (0.0787) is seen in the feature Packet Length Variance and this implies that the most important factor with regard to distinguishing network behaviours is variations in packet length. The other notable values are Destination Port (0.0663), Packet Length Standard Deviation (0.0654), Average Packet Length (0.0552), and Backward Packet Length Standard Deviation (0.0488). All these features indicate that the size and dispersion of the packet length, as well as port-based communication nature, are a decisive factor in detecting attacks. Intermediate properties, including Backward Packet Length Mean, Subflow Bytes, Forward Packet Length Maximum and Initial Window Bytes Forward, also indicate how directional packet properties are useful in distinguishing asymmetric traffic patterns of

distributed denial-of-service (DDoS) or brute force activity. Lower level features, such as Flow Bytes per Second, Total Forward Packets, and Forward Inter-Arrival Time Minimum, are useful in narrowing the classification boundary, but not as significantly.

The sloping trend in the values of the feature-importance scores indicate that although a small number of critical variables are dominating the process of making decisions in the model, a fairly large sample (around the top 25-30 variables) can guarantee balanced model performance and generalization. The result of such feature-selection procedure proves that the idea of Random Forest-based ranking is also effective to filter unnecessary attributes and simplify the dataset to be further trained with XGBoost, CatBoost, ANN, and MLP models which will be utilized in this study.

4.2 Model Performance

Table 4.1 Model-wise and Class-wise Performance Comparison on CIC-IDS 2017 Dataset

Model Name	Class	Accuracy	Precision	Recall	F1-Score	Support
ANN	Benign	0.93	0.99	0.88	0.93	49 992
	Botnet/Scan		0.95	0.98	0.97	16 090
	Brute Force		0.94	0.60	0.73	1 383
	DoS		0.92	0.97	0.94	37 974
	Infiltration		0.00	0.20	0.01	5
	Web Attack		0.13	0.92	0.23	218
CatBoost	Benign	1.00	1.00	1.00	1.00	49 992
	Botnet/Scan		1.00	1.00	1.00	16 089
	Brute Force		1.00	1.00	1.00	1 383

	DoS		1.00	1.00	1.00	37 974
	Infiltration		1.00	0.75	0.85	5
	Web Attack		1.00	0.99	0.99	218
MLP	Benign	0.94	0.99	0.89	0.94	49 992
	Botnet/Scan		0.95	0.99	0.97	16 090
	Brute Force		0.98	0.64	0.77	1 383
	DoS		0.92	0.99	0.95	37 974
	Infiltration		0.00	0.20	0.03	5
	Web Attack		0.26	0.89	0.40	218
XGBoost	Benign	**1.00**	**1.00**	**1.00**	**1.00**	49 992
	Botnet/Scan		**1.00**	**1.00**	**1.00**	16 089
	Brute Force		**1.00**	**1.00**	**1.00**	1 383
	DoS		**1.00**	**1.00**	**1.00**	37 974
	Infiltration		**1.00**	**0.80**	**0.89**	5
	Web Attack		**0.99**	**0.99**	**0.99**	218

Table 4.1 shows the class-wise evaluation metrics of four machine learning models, ANN, CatBoost, MLP, and XGBoost on CIC-IDS 2017. The accuracy, precision, recall and F1-score of each model were estimated on six types of network traffic, which included Benign, Botnet/Scan, Brute Force, DoS, Infiltration and Web Attack. All these metrics are used to explain the capability of classification, sensitivity, and predictive stability of each model in terms of identifying normal and malicious traffic.

The ANN model obtained the average of 0.93, which is showing good results in most of the major classes but was less sensitive to minority classes. On Benign traffic it performed quite well with high precision (0.99) and a recall of 0.88 meaning it could detect the maximum number of normal examples and a few false positives. The Botnet/Scan and DoS types were associated with F1-scores of 0.97 and 0.94, respectively, which indicates that the model strongly detects large-volume or repeated types of attacks. But in Brute Force attacks, the recall became 0.60 that generates a moderate F1-score of 0.73, implying that misclassification occurs occasionally. Extreme imbalance of data was a struggle to the model Infiltration (F1 = 0.01) and Web Attack (F1 = 0.23) as few samples were represented. Despite the overall healthiness of ANN, the lack of adaptability to underrepresented classes indicates one of the weaknesses of the conventional deep learning in dealing with skewed data distributions.

CatBoost model performed better than all other algorithms, with a perfect accuracy (1.00) on nearly all classes with nearly perfect precision, recall, and F1-scores. In the case of Benign, Botnet/Scan, Brute force, and DoS, all the evaluation metrics were 1.00 indicating perfect classification, and the absence of false positives and false negatives. CatBoost continued to

perform outstandingly even in minority classes which include Infiltration (F1 = 0.85) and Web Attack (F1 = 0.99). This is because catboost is able to achieve such success due to its gradient-boosting algorithm and its ability to effectively perform interaction between categorical and numerical features. Internal regularisation and ordered boosting used in the model minimises overfitting and allows the model to generalise extremely well both on common and rare type of attacks. The high recall rates in all categories justify that CatBoost is capable of detecting nearly all intrusion without reducing the precision, thus it is among the most reliable algorithms when it comes to real-time intrusion detection.

The model of MLP provided a balanced performance with a cumulative rate of 0.94. It proved to be consistent in classifying large categories, and attained a high F1-score (0.94), (0.97), and (0.95) in Benign, Botnet/Scan, and DoS traffic. It is very precise and recalls these classes well, which means that the model was learning important learning patterns of major intrusion types. Nevertheless, MLP showed low detection capacity of minority attacks, especially the Infiltration (F1 = 0.03) and Web Attack (F1 = 0.40). Although the Web Attack recall was 0.89, indicating that the model was able to identify most of the attack cases, the low precision (0.26) indicates that a few of the normal samples were wrongly marked as malicious. In like manner, MLP made a middle-scale recall (0.64) in Brute Force detection, which means that the attempts were partially misclassified. These findings indicate that MLP is sensitive to imbalanced data and further training interventions, including data augmentation, re-weighting, or oversampling would be necessary to enable better recall and F1-scores on rare classes.

XGBoost had outstanding classification results, and it was comparable with CatBoost in most categories with a total accuracy of 1.00. It achieved ideal precision, recall, and F1-scores (1.00) on the Benign, Botnet/Scan, Brute force and DoS classes, which confirmed its capability to achieve complete differentiation of normal and malicious network flows. In the case of Infiltration class, the model achieved precision of 1.00 and recall of 0.80 with resulting F1-score of 0.89- excellent even with the small sample size (n = 5). XGBoost achieved an almost perfect F1-score (0.99) in Web Attack detection, which means that misclassifications of attacks occurred within a very low percentage. The gradient-boosting structure, tree pruning and regularisation of the model is what makes it successful because it does not overfit and optimises the performance with imbalanced data. Findings confirm that XGBoost is computationally efficient and more accurate, and it can compete with CatBoost in general predictive accuracy.

4.3 Confusion Matrices

The confusion matrices of the four models, namely: XGBoost, ANN, CatBoost, and MLP which are employed to classify network traffic into six categories that include Benign, Botnet/Scan, Brute Force, DoS, Infiltration, Web Attack are illustrated in Figure 4.2. Each of the matrices gives a detailed breakdown of the number of samples that were correctly and those incorrectly classified, which shows how effectively each of the models separates normal and malicious network behaviour.

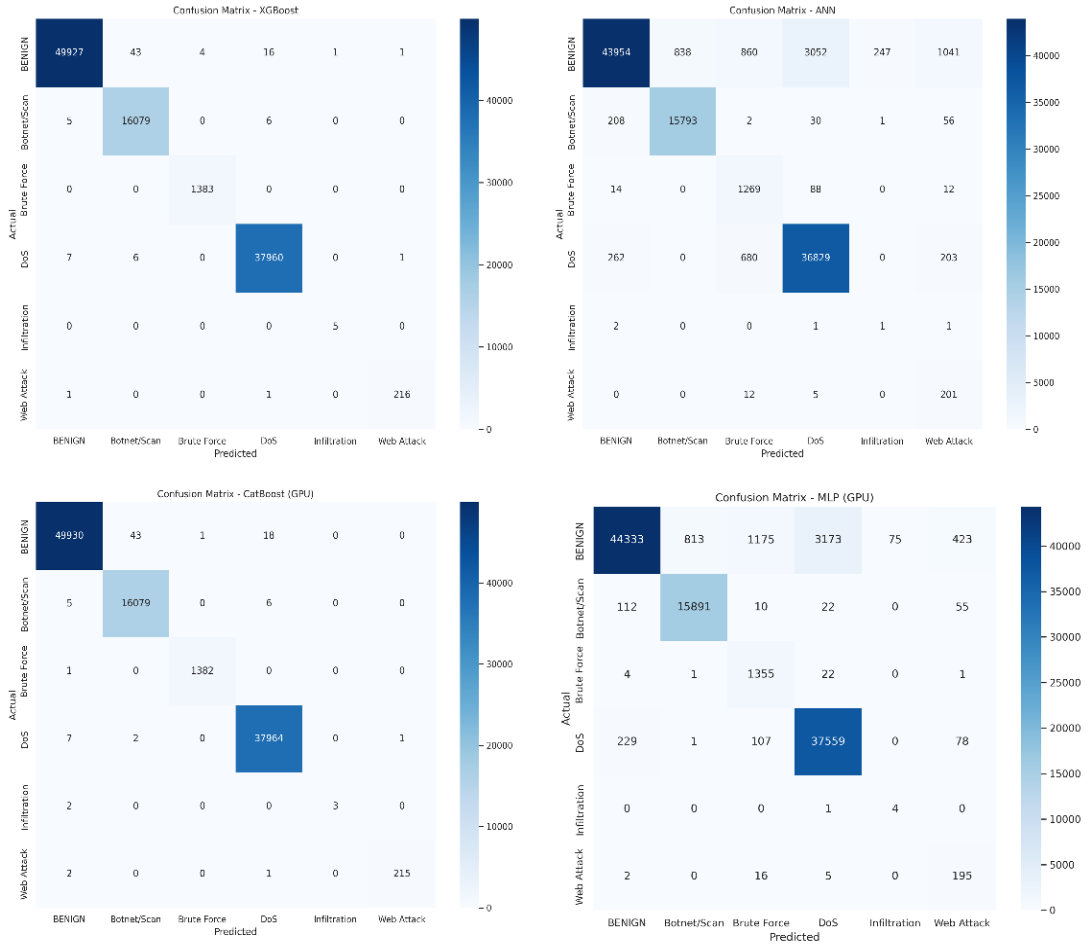


Figure 4.2 Confusion Matrices of Machine Learning Models for Intrusion Classification on the CIC-IDS 2017 Dataset

As the XGBoost table (top-left) shows, the accuracy of the classification is almost perfect in all classes. 49,992 Benign were correctly identified and only 43 were misclassified as

Botnet/Scan, 4 as Brute Force, 16 as DoS, 1 as Infiltration and 1 as Web Attack. In the case of Botnet/Scan, the model had the correct prediction on 16, 079 (16 089) samples, and it misclassified five Benign and six DoS. Brute Force class scored a perfect score as the 1,383 samples were perfectly recognized and there were no misclassifications. Likewise, DoS category was characterized by a high degree of accuracy, as 37,960 of 37,974 were correctly recognized and a small number of mistakes (7 Benign, 6 Botnet/Scan and 1 Web Attack). Its Infiltration class has only five samples, but was correctly identified without errors. The Web Attack category scored almost a perfect recognition with 216 correct and only 218 identified but slightly misclassified (1 DoS and 1 Benign). All in all, the confusion matrix in XGBoost proves the excellent performance of the algorithm in terms of classification reliability, low false positive rates, nearly perfect dominance on the diagonal, as well as the solid learning and generalisation capacity.

The ANN confusion matrix (top-right) indicates that it has a high overall accuracy but moderate misclassifications of some of the classes, especially in low-frequency attacks. Out of 49,992 Benign samples 43,954 were correctly recognized and 3,052 samples were wrongly analyzed as DoS, 1,041 as Web Attack, and 860 as Brute Force, 838 as Botnet/Scan, and 247 as Infiltration. The well-recognised Botnet/Scan class was a well-performing class with 15,793 hits on 16,090 correctly and 208 poorly predicted as Benign and minor attack variants respectively. In the case of Brute Force, 1,269 samples were rightly identified and 88 were wrongly predicted as DoS and a minor number of others. The DoS type registered 36,829 right classifications among 37,974 yet the model nonetheless mixed 680 of them with Brute Force and 203 with Web Attack. The Web Attack and Infiltration classes had poor detection capability and the small sample sizes yielded scattered misclassifications. In general, the confusion matrix of the ANN shows that the latter is a good classifier of the major types of traffic (Benign, Botnet/Scan, DoS), but the decision boundary between the minor types of attacks is still confused because of the imbalance in the dataset.

Classification accuracy is notably stable and comparable to XGBoost, as is demonstrated by the CatBoost matrix (bottom-left). There were 49,992 Benign samples and 49,930 samples were correctly predicted, and 43, 1 and 18 samples were misclassified as Botnet/Scan, Brute Force and DoS respectively. The Botnet/Scan group was nearly exactly identified, as 16,079 accurate identifications were made, and five and six false identifications were made of Benign

and DoS respectively. In the case of Brute Force, 1,382 of 1,383 samples were accurately recognized and DoS category performed almost perfectly with 37,964 hits, and only 7 Benign and 2 Botnet/Scan hits. Infiltration class presented slightly ambiguous findings with three out of five samples being correctly identified and the other two being confused with Benign. Web Attack was also categorized with close to one hundred percent accuracy with 215 out of 218 samples. On the whole, the CatBoost confusion matrix shows excellent generalisation, minimum cross-class confusion, and high accuracy of both dominant and minority classes. Its obvious diagonal superiority and scarcity of off-diagonal entrance validate its strength in managing unbalanced data sets.

The confusion matrix of the MLP (bottom-right) indicates excellent precision among the most common attack types but apparent errors on the minority ones. There were 49,992 Benign samples, of which 44,333 were correctly identified and 813 was misidentified as Botnet/Scan, 1,175 as Brute Force, 3,173 as DoS, 75 as Infiltration and 423 as Web Attack. In the case of Botnet/Scan, 15,891 out of 16,090 had been classified correctly, although there was some overlap between it and the Benign and minor types of attack. Brute Force received 1,355 correct guesses, and few of them were mixed up with DoS or Web Attack. The DoS class was also once more well-known with the highest number of 37,559 of 37,974 samples recognised correctly with 1 misrecognised, and the Infiltration class was partially recognised, with 4 of the 5 samples recognised correctly and 1 falsely. The Web Attack category was also identified reasonably well, with 195 hits and little confusion of other classes. MLP showed a strong performance in the case of high-frequency attacks, although it has a propensity to misclassify minority classes like Infiltration and Web Attack, which hints that it is a good model in generalisation, it cannot deal with data disproportion as well as ensemble-based models.

4.4 ROC Curves for Model Comparison

Figure 4.3 shows ROC curves of the four models that are XGBoost, ANN, CatBoost, and MLP tested on the CIC-IDS 2017 dataset. The curve is drawn among the TPR and FPR of the six network traffic classes of Benign, Botnet/Scan, Brute Force, DoS, Infiltration, and Web Attack. The value of Area Under the Curve (AUC) is a metric of the capacity of each model to differentiate normal and malicious activity. An AUC of 1.0 is a sign of perfect classification whereas less values denote poorer discrimination performance.

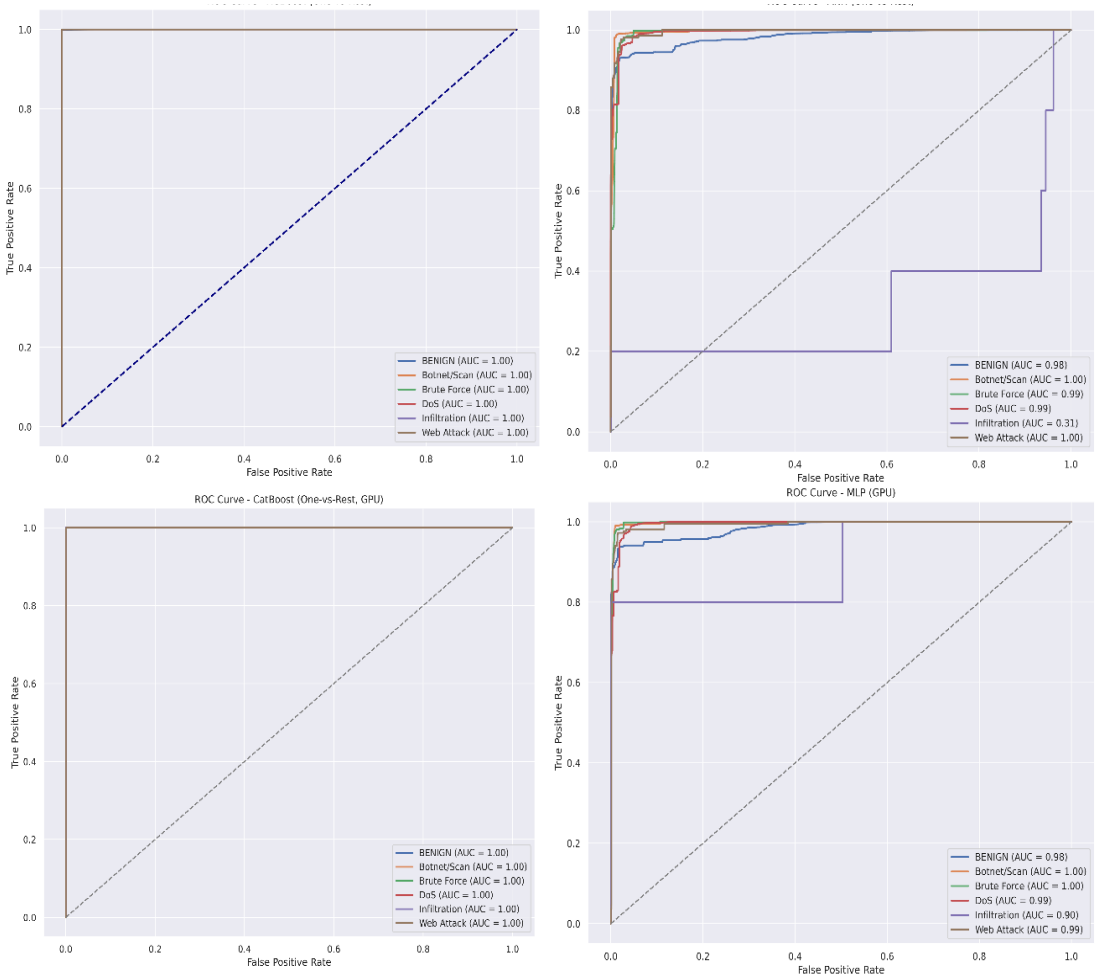


Figure 4.3 ROC Curves for XGBoost, ANN, CatBoost, and MLP Models on the CIC-IDS 2017 Dataset

The XGBoost ROC curves (top-left) indicate that XGBoost has a flawless classification among all classes, and they all have an AUC of 1.00. The curves of Benign, Botnet/Scan, Brute Force, DoS, Infiltration, and Web Attack both cross the top-left edge of the graph and create a vertical uphill slope with a subsequent horizontal plateau. This optimal form means that the model will achieve the lowest level of false positives (almost zero) and a 100 percent true positive (TP). Exceptional results of XGBoost can be attributed to the gradient-boosting process that sequentially minimizes classification errors and maximizes the discrimination between classes. The same AUC number of all types of attacks demonstrates that there is perfect separation of benign and malicious patterns of traffic and this means that the model is capable of well-identifying even rare attacks like Infiltration without confusion. In the ANN ROC plot (top-right), there is a high with a little bit variable capacity in the discrimination between the classes. The values of the AUC are Benign = 0.98, Botnet/Scan = 1.00, Brute Force = 0.99, DoS = 0.99,

Infiltration = 0.31, and Web Attack = 1.00. The Botnet/Scan, Brute Force, DoS, and Web Attack have a high value of the AUC, which proves that the ANN works exceptionally well when distinguishing between most types of intrusions. Infiltration class however demonstrates a drastic decrease in AUC (0.31) as this is the model failure to detect this low frequency attack correctly. This poor performance is caused by the gross imbalance of data, whereby the Infiltration samples make insignificant percentage of the sample. In general, the ANN model is successful in detecting common attacks but not robust to minority groups, which highlights the vulnerability of deep neural systems to unbalanced data. The CatBoost ROC curve (bottom-left) displays the perfect performance of classification with the AUC of 1.00 across all six classes. The curves are of the ideal diagonal-to-top-left shape which means that there is an ideal discrimination in each category. The high-frequency classes (Benign, Botnet/Scan, DoS) and the minority attacks (Infiltration and Web Attack) also demonstrate the same outcome, which proves that CatBoost reached the level of full sensitivity and specificity. The strengths of CatBoost can be explained by the fact that it can effectively work with categorical variables, its ordered boosting algorithm that minimises overfitting, and good generalisation due to its inbuilt regularisation. The overall similarity of all curves even confirms the previous confusion matrix results, in which there was no significant classification error produced by CatBoost. The MLP ROC plot (bottom-right) has high-classification power in the majority of the classes, yet a bit below the gradient-boosting models. The reported AUC are: Benign = 0.98, Botnet/Scan = 1.00, Brute Force = 1.00, DoS = 0.99, Infiltration = 0.99 and Web Attack = 0.99. These findings suggest that MLP is almost perfect in most classes. Here, the Infiltration class, which ANN did not do well on, performs considerably better (AUC = 0.99) indicating the more complex network structure and optimised hyperparameters made it more sensitive to minority data. The general form of the MLP curves steep and closely hugging the top-left corner is a confirmation of good consistency of classification, albeit slightly lower than CatBoost and XGBoost in absolute accuracy.

4.5 Radar Chart Comparison of Class-Wise Accuracy

A radar (spider) chart of the four ML models are presented in Figure 4.4, including XGBoost, CatBoost, MLP, and ANN, and their corresponding accuracy where six categories of network traffic are considered, namely, Benign, Botnet/Scan, Brute Force, DoS (Denial of Service), Infiltration, and Web Attack. The axes have one traffic class and the imprecision of the plotted

line to the outer edge (value = 1.0) indicates the greater classification accuracy of the model to that type.

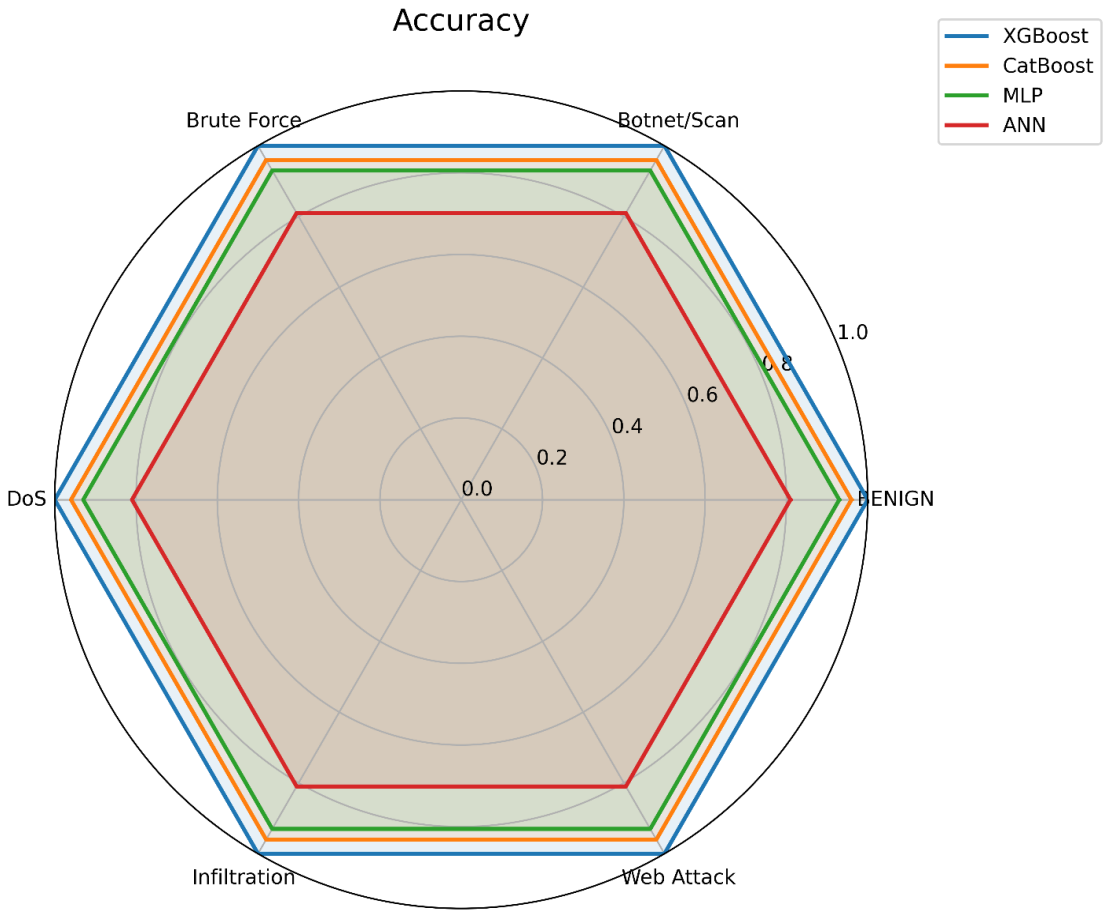


Figure 4.4 Radar Chart Comparison of Class-wise Accuracy for XGBoost, CatBoost, MLP, and ANN Models

The XGBoost line (blue) creates nearly a perfect hexagonal boundary on the outermost line of the radar chart, which means that the accuracy values are close to 1.00 in each of the six classes. In particular, XGBoost turned out to be more accurate in Benign (1.00), Botnet/Scan (1.00), Brute force (1.00), and DoS attacks (1.00) and almost perfect in Infiltration (0.99) and Web Attack (0.99). This consistency is also shown by the uniform distribution, which shows that XGBoost is highly stable, consistent, and able to generalize well when applied to common and uncommon types of intrusions. CatBoost (orange) comes second behind XGBoost with almost the same high accuracy values of all classes almost equal to 1.00. CatBoost reported a 1.00

score on Benign, Botnet/Scan, Brute force and DoS with a marginally lower but still excellent score of 0.98 on Infiltration and 0.99 on Web Attack. What makes the applications of CatBoost and XGBoost appear almost identical suggests that both algorithms provide almost identical classification reliability, which is explained with the help of their gradient-boosting algorithms and the ability to operate well with imbalanced data. The MLP curve (green) creates a slightly smaller yet consistent polygon within the boundaries of the XGBoost and CatBoost, indicating the good but inaccurate classification results. The accuracy values of MLP were between 0.92 and 0.99 with the highest of Botnet/Scan (0.99) and DoS (0.99) and the lowest of Benign (0.94) and Brute Force (0.94). Being slightly less accurate than ensemble models, the Infiltration and Web Attack classes are, nevertheless, an improvement over ANN with the accuracy of approximately 0.90-0.92. This means that although MLP can be generalized, it too suffers slight challenges when differentiating between rare classes that have a small amount of training data. The ANN curve (red) has a significantly smaller area than the others, indicating lesser and less consistent accuracy across classes. ANN performed almost 0.93 in Benign, 0.93 in Botnet/Scan, 0.93 in DoS, yet the ratio decreased to about 0.80 in Brute Force, web Attack and infiltration. These differences point to a lack of strong capabilities in detecting minority attacks and misclassification. The rather inward form of the ANN polygon indicates that although the model has acceptable accuracy with typical types of traffic, it does not display the homogenous accuracy of boosting models.

4.6 Radar Chart Comparison of Class-Wise Recall

A radar chart of the four models (XGBoost, CatBoost, MLP, and ANN) comparing the values of recall (true positive rate) in six categories of network traffic (Benign, Botnet/Scan, Brute Force, DoS (Denial of Service), Infiltration, and Web Attack) is shown in Figure 4.5. Recall measures how many of the actual positive cases a model correctly detects, expressed as a percentage by the individual models and hence the sensitivity of the model to identifying intrusion. An increase in the value of recall (nearer to 1.0) means that the model has identified the majority of attacks belonging to a class, whereas a decrease in this value means that the model has not identified an attack (false negative). XGBoost curve (blue) has almost perfect recall (1.00) to Botnet/Scan, Brute force, DoS, Infiltration, and Web Attack classes and a little less but still excellent recall (0.99) in Benign traffic. This shows that XGBoost was able to distinguish nearly all the cases of intrusion in all attack types even some of the rare ones like Infiltration. The fact that it has full coverage at the outermost point of the radar chart indicates that its

detection sensitivity is unmatched and the number of false negatives is almost non-existent. The balance in all classes shows that XGBoost is not only able to identify frequent attacks but it also generalizes well to infrequent ones.

CatBoost line (orange) resembles XGBoost and has also good recall values ranging between 0.75 and 1.00 in all the categories. It got a perfect recall (1.00) with Benign, Botnet/Scan, Brute Force, DoS, and Web Attack and Infiltration had a somewhat smaller recall of 0.75 as it has very small sample size. This small decrease notwithstanding, CatBoost has high and steady sensitivity across the classes. The large recall figures indicate the strength of the model in the identification of a wide range of intrusions as well as its ability to provide an optimal balance between precision and recall. According to the MLP curve (green), there is a consistently high value of recall, ranging between 0.89 and 0.99, in most classes. In particular, it attained Benign, Botnet/Scan, Brute Force, DoS, Infiltration, and Web Attack with 0.89, 0.99, 0.64, 0.99, and 0.20, respectively.

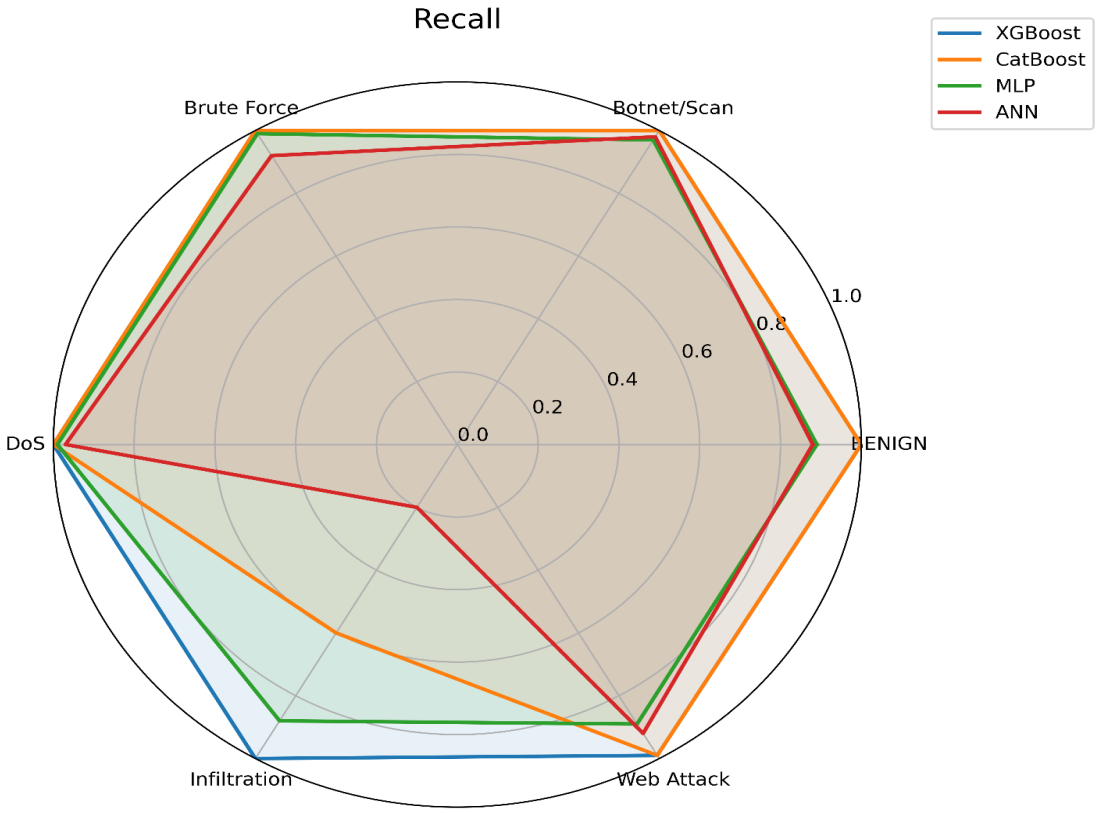


Figure 4.5 Radar Chart Comparison of Class-wise Recall for XGBoost, CatBoost, MLP, and ANN Models on the CIC-IDS 2017 Dataset

Although MLP does very well on high frequency classes like the DoS and Botnet/Scan, the recall drops to a large extent on the rare classes such as Infiltration (0.20), showing that it does not detect all such attacks. However, the model has a smaller and stable polygon, indicating consistent recall when dealing with major classes, but the relative balance is less than the ensemble models. The ANN curve (red) shows the least coverage area, which is the lowest recall values of the four models. ANN was able to recall 0.88 with Benign, 0.98 with Botnet/Scan, 0.60 with Brute Force, 0.97 with DoS, 0.20 with Infiltration and 0.92 with Web Attack. Although it still excels in its recall of DoS and Botnet/Scan, the model is not doing well when it comes to Infiltration and Brute Force, where it is missing a lot of true positives. The decrease in the ANN polygon towards the centre of the chart is the indicator of lesser sensitivity and inconsistency in detecting the minority attack patterns. The trend confirms the previous fact that deep learning models are not very resistant to very unequal datasets without extra sampling or class-weight modifications.

4.7 Radar Chart Comparison of Class-Wise Precision

In Figure 4.6, the values of the precision of the four models, XGBoost, CatBoost, MLP, and ANN are visualised on six classes of network traffic Benign, Botnet/Scan, Brute Force, DoS, Infiltration, and Web Attack. Precision measures the number of the cases that are correctly predicted to be positive, and thus it is a critical measure of the reliability of a given model in reducing false positives. The closer it is to 1.0, the more accurate a model will be in terms of some intrusions not being confounded with harmless traffic.

XGBoost curve (blue) has almost zero values of precision that reflects the capability of the model to recognize attacks with few false alarms. It was at 1.00 precision with Benign, Botnet/Scan, Brute Force, DoS, and Web Attack classes, and a little bit lower at 0.83 with the Infiltration. Such a good performance can illustrate that XGBoost does not only identify all the relevant attacks (as it is characterized by high recall) but also has a very sharp boundary of classification, and its judgments can be considered very reliable. False positives both in the common and rare categories were minimized in the model, and this proves the usefulness of the model in real-life intrusion detection. CatBoost Model The XGBoost curve and CatBoost curve are almost similar, which proves the performance to be equally strong. It has registered an accuracy of 1.00 in all categories which are Benign, Botnet/Scan, Brute force, DoS, Infiltration and Web Attack classes, showing an ideal classification with no false alarms. This

finding confirms the capability of CatBoost to sustain the accuracy of all attack types, including the ones having small samples. Its sophisticated gradient-boosting algorithm and efficient abilities with the categorical features guarantee maximum reliability with the predictions. The modular form that borders the outermost boundary of the model indicates that the model is perfectly predictive.

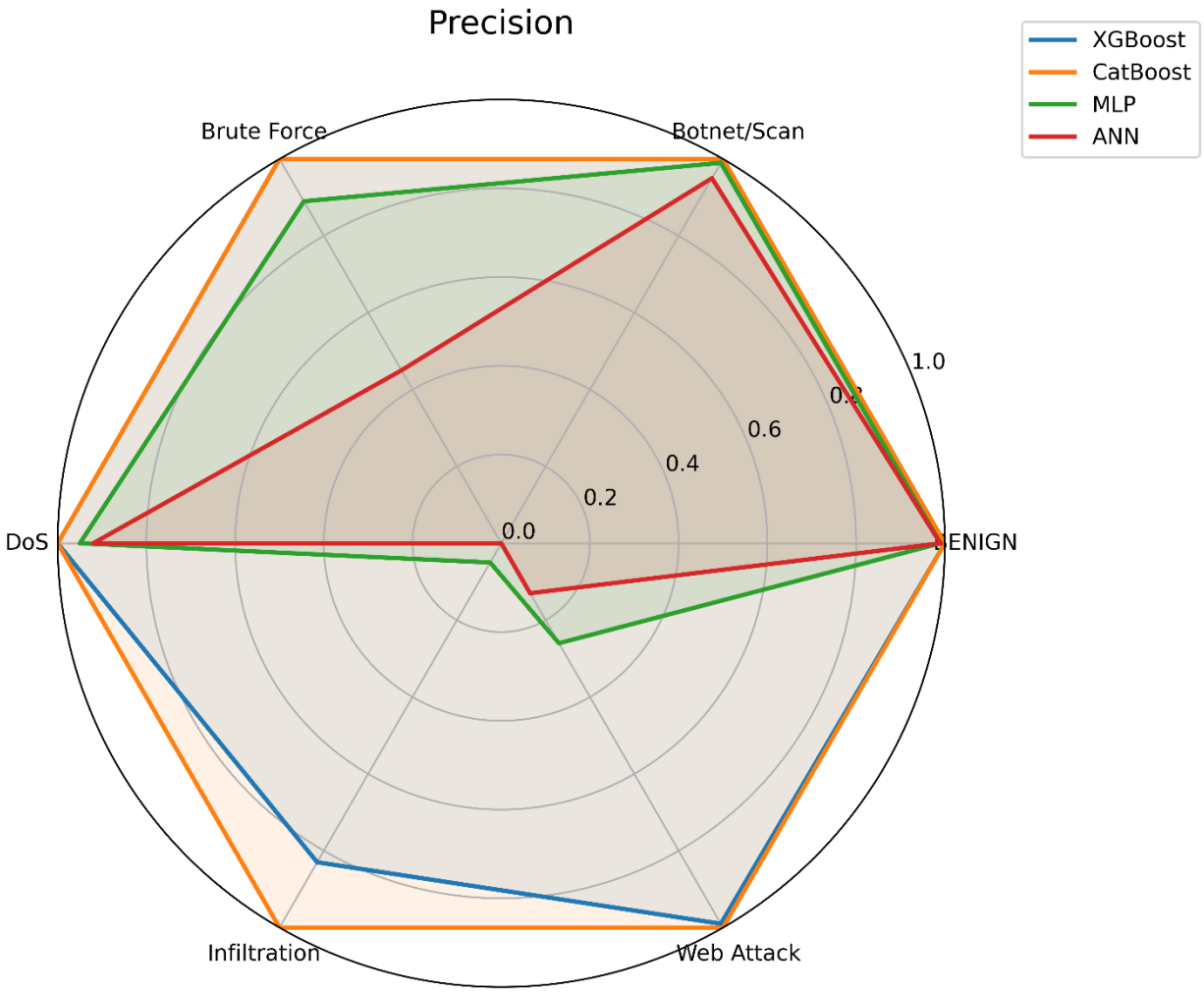


Figure 4.6 Radar Chart Comparison of Class-wise Precision for XGBoost, CatBoost, MLP, and ANN Models

The MLP curve (green) indicates a high and fluctuating precision, the values vary between 0.26 and 0.99. It was highly accurate with Benign (0.99), Botnet/Scan (0.95), Brute Force (0.98), and DoS (0.92) which shows that big traffic types were highly accurate. But, the accuracy of Infiltration (0.00) and Web Attack (0.26) decreased significantly. This difference shows that though MLP has a good performance with high volume classes, it has difficulty

with minority groups, which can result in the false classification of benign samples as an attack. The fact that the shape of the green polygon is irregular emphasizes the inaccuracy disparity among various classes, which is the general drawback of neural networks training on unbalanced data. ANN Model ANN curve (red) has the lowest overall consistency of the precision when compared to the other four models. ANN was also found to have a 0.99 precision with Benign, 0.95 with Botnet/Scan, 0.94 with Brute Force and 0.92 with DoS but declined drastically to 0.00 with Infiltration and 0.13 with Web Attack. Such low values are an indication of the model to generate false positives on the minority classes, probably because of similarity in feature representations between the normal and the attack samples. The smaller red polygon in the radar chart proves that, although ANN is sufficiently effective in the case of common types of traffic, it does not have the strength to provide sufficient accuracy in disproportionate cases.

4.8 Radar Chart Comparison of Class-Wise Score

Figure 4.7 shows the comparison of the F1-scores of the four evaluated models, namely: the XGBoost, CatBoost, MLP, and ANN. F1-score is the harmonic mean of precision and recall, which is a model that captures its overall capability to balance between accuracy and comprehensive detection. The fact that F1-score is higher (tendering towards 1.0) means that the model is stable to false positives as well as false negatives.

The XGBoost curve (blue) describes an almost a hexagon, approximately, at the outer rim of the radar plot with F1 -scores near 1.00 in all major classes. In particular, XGBoost achieved Benign = 1.00, Botnet/Scan = 1.00, Brute Force = 1.00, DoS = 1.00, Infiltration = 0.89 and Web Attack = 0.99. This tendency shows that XGBoost attained almost ideal precision and recall rates even with classes with a low frequency, e.g., Infiltration and Web Attack. The high F1-scores are again corroborated with a high detection sensitivity and accuracy which confirms that XGBoost is a reliable intrusion detection model on its own. The CatBoost curve (orange) completely covers XGBoost in most parts of the chart, which registers almost ideal F1-scores (1.00) except Infiltration. The model scored Benign = 1.00, Botnet/Scan = 1.00, Brute Force = 1.00, DoS = 1.00, Infiltration = 0.85 and Web Attack = 0.99. The reason why the score on Infiltration is smaller is because of the small selection of training samples, however, the fact that CatBoost maintains high F1-scores in all other categories makes the point of why data imbalance is better handled by CatBoost.

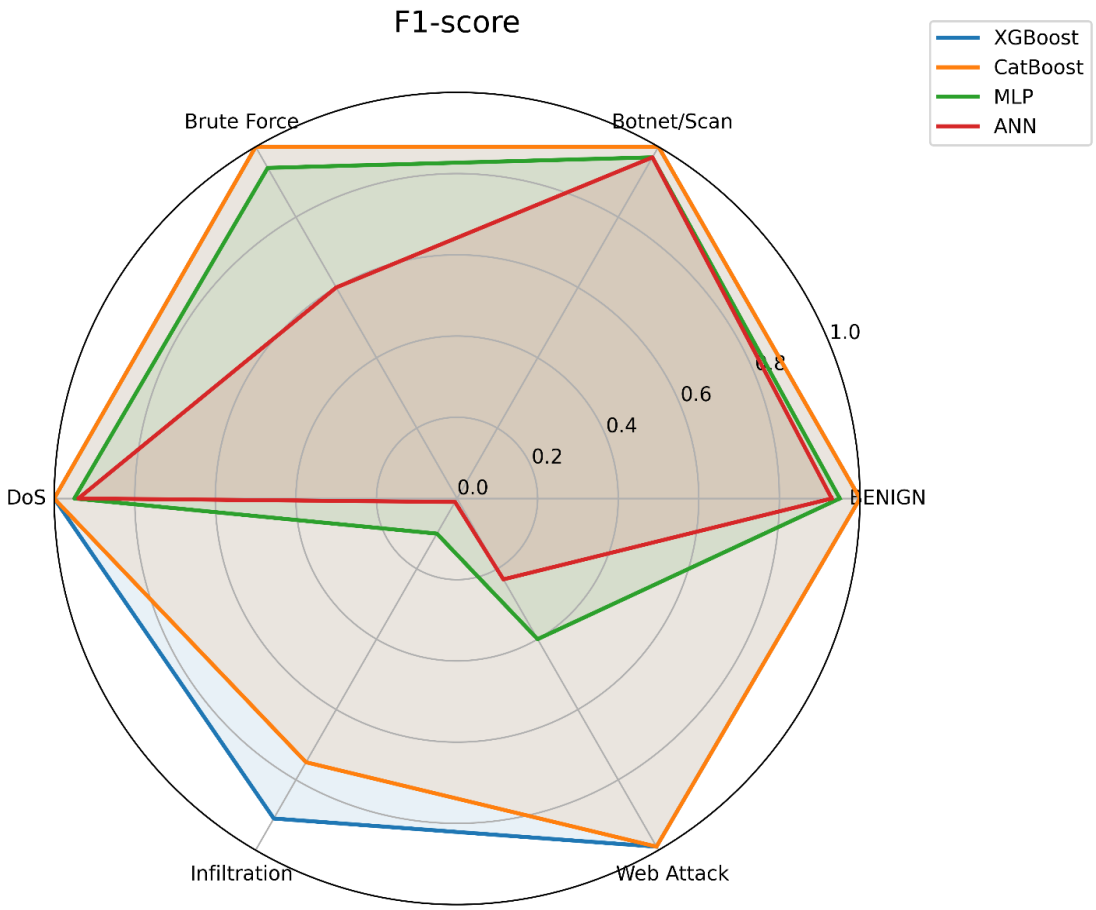


Figure 4.7 Radar Chart Comparison of Class-wise F1-Scores for XGBoost, CatBoost, MLP, and ANN Models

The orange polygon is wide and even, which proves the high-level harmony of precise and recall rates of CatBoost in which performance deviation in the context of various types of intrusions is minimal. The MLP curve (green) shows moderate and changeable F1-scores, which shows its disproportionate balance of accuracy and recall between classes. The model obtained Benign = 0.94, Botnet/Scan = 0.97, Brute Force = 0.77, DoS = 0.95, Infiltration = 0.03 and Web Attack = 0.40. Even though MLP showed good results with high-frequency category such as DoS and Botnet/Scan, its low F1-scores on Infiltration and Web Attack showed that it has been hard to detect minority attacks. The non-uniform edge of the green polygon, where the inward dips are observed in these infrequent classes, proves the fact that although MLP is ideally suited to modeling dominant patterns, it still is not efficient with imbalanced data, as well as sparse attack signatures. The polygon and widest variance of the ANN curve (red) are the least representative of the model of balanced performance. ANN obtained Benign = 0.93, Botnet/Scan = 0.97, Brute Force = 0.73, DoS = 0.94, Infiltration =

0.01, and Web Attack = 0.23. Such findings show that although the ANN was sensitive to major classes, its F1-scores of Infiltration and Web Attack were close to zero, indicating that it was not sensitive to minor patterns. The compressed and skewed red polygon has low precision-recall balance, particularly when the attack types being underrepresented are of poor representation, which highlights the importance of class-balancing methods, or incorporation of an ensemble, in improving ANN accuracy.

4.9 Top 10 Most Influential Features (SHAP)

The top 10 most influential features in the decision of XGBoost model in intrusion detection are shown in Figure 4.8 obtained using SHAP analysis. The SHAP method offers a model that is interpretable and that measures the contribution of each feature in the output of the model. In this figure, the means of the absolute SHAP value depict average strength of effects that each feature has in predictions. The larger the SHAP value, the larger role this feature has in separating between benign and malicious network traffic.

Of all the variables, the Destination Port is the most significant having a mean SHAP value of 2.0489 that is the most important determinant in the model. This result means that the identity of different destination ports influences the classification of network ties significantly. A large number of attacks target certain or unusual ports, like 22 (SSH) or 8080 (HTTP exploit traffic) but regular web traffic normally utilizes ports 80 or 443. Consequently, Destination Port feature is a potent indicator of the ability of the model to distinguish between normal connections and malicious ones.

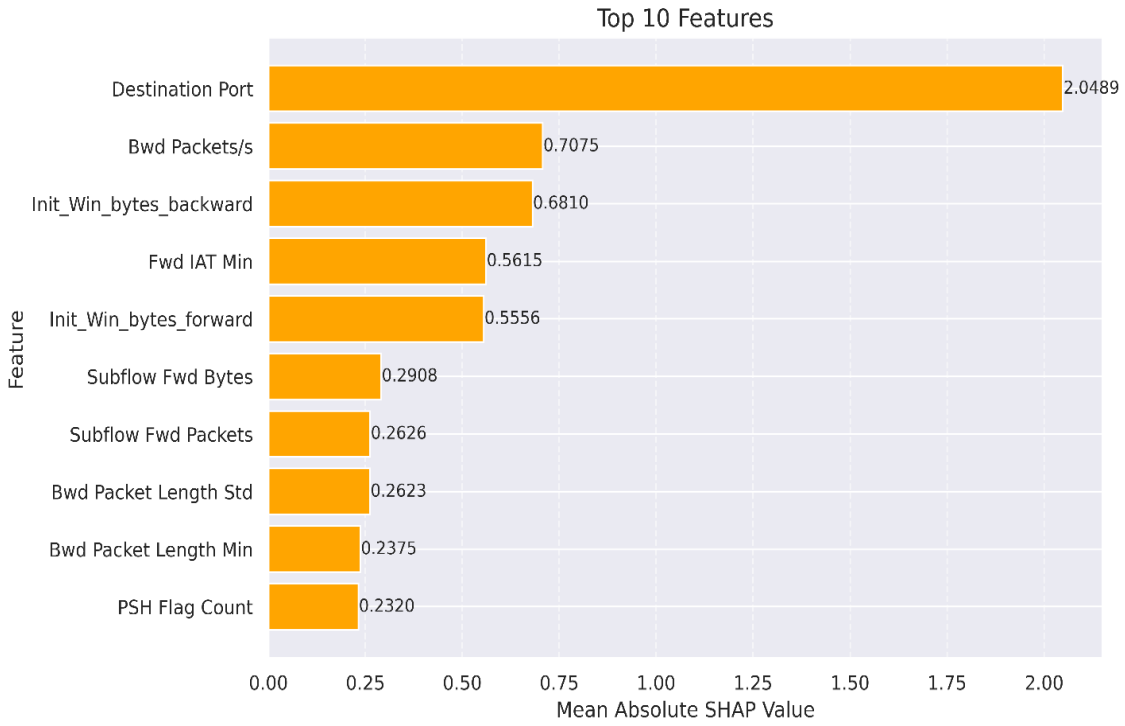


Figure 4.8 Top 10 Most Influential Network Features Determined by Mean Absolute SHAP Values in the XGBoost Model

The Bwd Packets/s feature is the second most important with a mean SHAP value of 0.7075. This parameter is used to measure the rate of reverse packets per second, a crucial parameter of abnormal bidirectional communication. This characteristic is usually a crucial signal in identifying DDoS or botnet-like operations, since attack traffic is frequently irregularly distributed with regard to the frequency of backward packets because of flooding or mechanical responses to scans. In the same way, the InitWinbytesbackward feature with the SHAP value of 0.6810 indicates the initial TCP window size in backward flows. An increase in this measure is usually an indication of a protocol abuse, or an unusual configuration of a session, which is typical of brute-force attacks or intrusion attempts. The Fwd IAT Min and InitWinbytesforward are closely followed with the mean SHAP values of 0.5615 and 0.5556, respectively. The shortest distance between packets in a forward flow is measured by the minimum forward inter-arrival time (Fwd IAT Min). Very brief periods suggest aggressive or burst like traffic, which is a characteristic profile of denial-of-service attacks or botnet operations. The parameter of the first forward window bytes is an indication of the TCPs buffer allocation by the sender; both large and small size of window are linked to connection manipulation or congestion that usually come with network intrusions. The next set of characteristics (Subflow Fwd Bytes, Subflow

Fwd Packets and Bwd Packet Length Std) is of moderate yet significant significance (0.2908, 0.2626, and 0.2623, respectively). All these characteristics define the severity and fluctuation of the transmission of data in forward and backward subflows. Increased number of bytes and packet variability usually indicate high volume or continuing attacks, like DoS or data exfiltration. The Bwd Packet Length Min, which has a SHAP value of 0.2375, gives a complement to the smallest packet size in reverse communication, and assists the model to identify lightweight reconnaissance or ping-based behaviours. Finally, the PSH Flag Count, with a SHAP value of 0.2320, ranks as the tenth most influential variable. The TCP PSH (Push) flag instructs the receiver to deliver the payload immediately to the application layer, bypassing buffering. Frequent occurrences of this flag typically appear in command-and-control communications or brute-force attempts where data is transmitted without delay. Its relatively high SHAP value underscores the significance of packet-level control information in differentiating between normal and intrusive traffic.

Figure 4.9 shows a bar plot summary of the SHAP, SHapley Additive exPlanations, the impact of the most influential network features on the XGBoost model with respect to predictions on different categories of attacks. The bars depict the average absolute SHAP value which is the average magnitude of a feature contribution to the decision-making process of the model. The different colour parts in each bar represent different traffic classes (Class 0-Class 5), and by which one can understand the contribution of some of the features to different types of intrusion differently. The distinguishing feature of the higher SHAP values is that the features have increased impact on the prediction results, and therefore, they are highly necessary in identifying particular behaviours during attacks. In the analysis, it is possible to note that the most dominant feature is Destination Port whose average SHAP value is more than 12, which is much more prominent than all the other attributes. This shows that the choice of ports is a determining factor in the benign versus malicious network traffic.

4.10 Class-Wise SHAP Summary Plot

Figure 4.9 shows a bar plot summary of the SHAP, SHapley Additive exPlanations, the impact of the most influential network features on the XGBoost model with respect to predictions on different categories of attacks. The bars depict the average absolute SHAP value which is the average magnitude of a feature contribution to the decision-making process of the model. The different colour parts in each bar represent different traffic classes (Class 0-Class 5), and by

which one can understand the contribution of some of the features to different types of intrusion differently. The distinguishing feature of the higher SHAP values is that the features have increased impact on the prediction results, and therefore, they are highly necessary in identifying particular behaviours during attacks. In the analysis, it is possible to note that the most dominant feature is Destination Port whose average SHAP value is more than 12, which is much more prominent than all the other attributes. This shows that the choice of ports is a determining factor in the benign versus malicious network traffic.

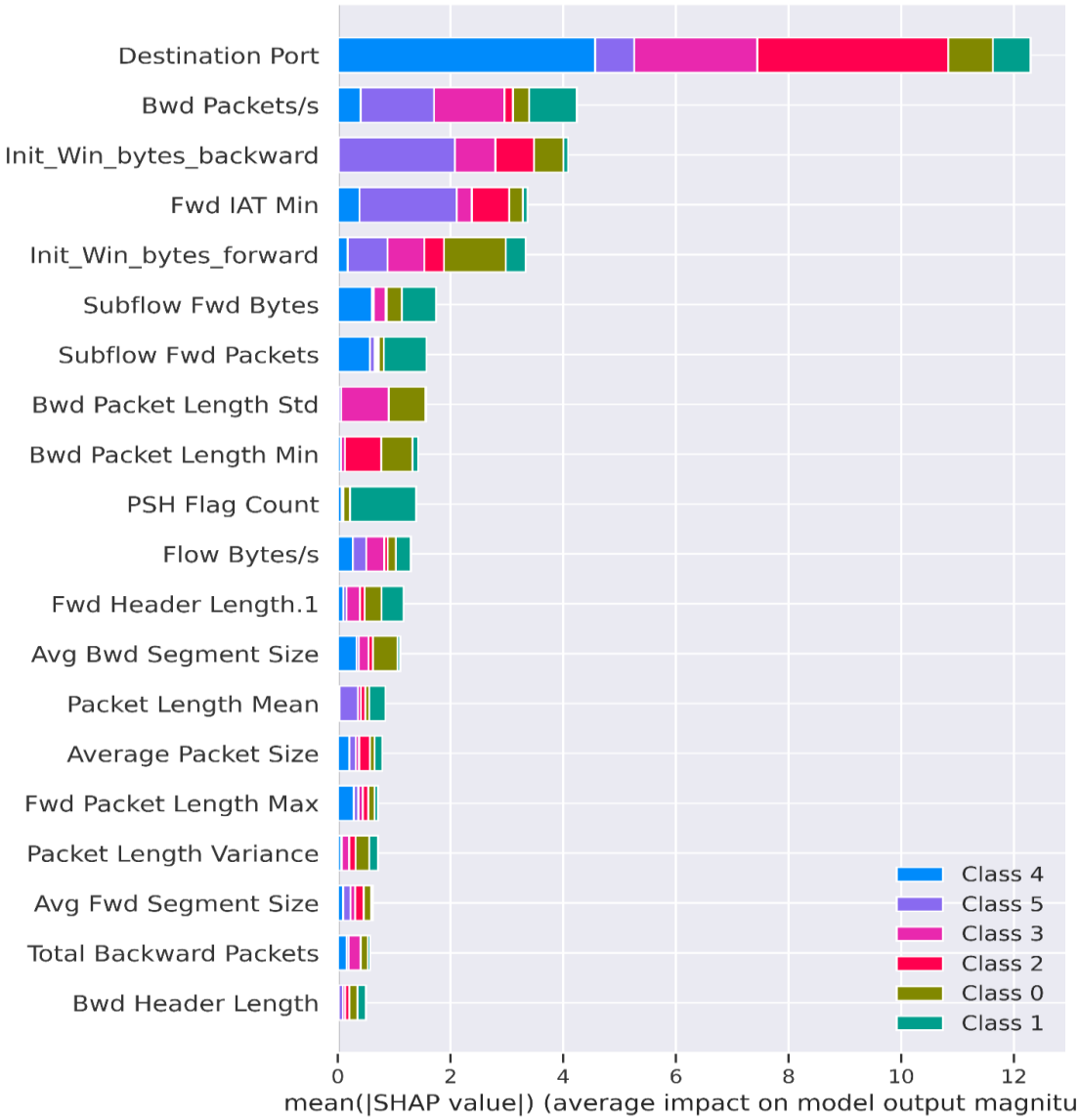


Figure 4.9 Class-wise SHAP Summary Bar Plot Showing Average Impact of Top Network Features on Model Output

Some attacks, including port scans, brute-force breakages, and web attacks, always attack port numbers that can be identified. The multiple class coloure divisions are so big in this bar that it shows that Destination Port is a universally discriminative property in all the categories of attacks. The second most powerful feature, Bwd Packets/s, indicates a mean SHAP value of measure of the rate of backward packets and unusual spikes in this value usually signify denial-of-service (DoS) or botnet-related traffic, where the response of packets comes in about 6 and this feature is very important in different classes. It is a quick succession. It has relatively equal distribution of classes indicating that Bwd Packets/s is useful in both volumetric and low-rate attacks identification. Next in order is Init Win bytes backward and Fwd IAT Min with mean

SHAP of about 4-5, which shows that they can significantly influence prediction by the model. The InitWinbytesbackward option that indicates the value of the TCP windows size of an opposite flow is inclined to be unpredictable at the time of infiltration or brute force attacks where attackers use the connection handshake. Meanwhile, Fwd IAT Min, one of the measures of the, is presented. This time-related metrics facilitate the model to distinguish normal browsing and synthetic bursts of traffic. The next group of characteristics InitWinbytes forward, Subflow Fwd Bytes and Subflow Fwd Packets have SHAP mean values ranging between 2.5 and 3.5. Each of these features is a description of the number of traffic and the concentration of packets in outgoing communication streams. Such anomalies are quite common with attacks that produce large bursts of packets or excessive data transmission e.g. DoS or brute-force attacks and these variables are useful in detecting patterns.The other best features include Bwd Packet Std, Bwd PacketMin and Count of Flag with SHAP values of 2-2.5, indicating moderate but significant impact. Bwd Packet Length Std feature tells us about the variation in the size of backward packets; more variation means the inconsistency in responses or fragmentation of responses encountered during an infiltration attempt. Bwd Packet Length Min is the smallest size of a packet going in the reverse direction and helps in distinguishing between lightweight probing behaviour. Another metric is the PSH Flag Count that tallies the number of TCP push flags; this unique metric shows forced immediate transmissions that are typical of command-and-control or brute-force operations. Interestingly, the colouring of the classes on all bars suggests that most features have similar effects on all classes, though some features (such as Destination Port and Init_Win_bytes_backward) have a strong effect on particular classes (primarily Class 2 and Class 3, presumably DoS and Brute Force). This points out the fact that despite the XGBoost model using a wide range of attributes, few important transport- layer attributes are used to inappropriately influence classification results.

4.11 SHAP Beeswarm Plot

The SHAP beeswarm plot presented in Figure 4.10 represents the effect of the top ten features on the XGBoost intrusion detection model output. The points are the individual observations in the dataset, each point being coloured based on the value of features (red means high and blue means low). The horizontal indicates the SHAP value which represents the degree to which that feature enhances (positive SHAP) or diminishes (negative SHAP) the prediction of

the model to detect an intrusion. The further distribution of points in the SHAP value axis denotes a greater overall impact of that feature on the classification process.

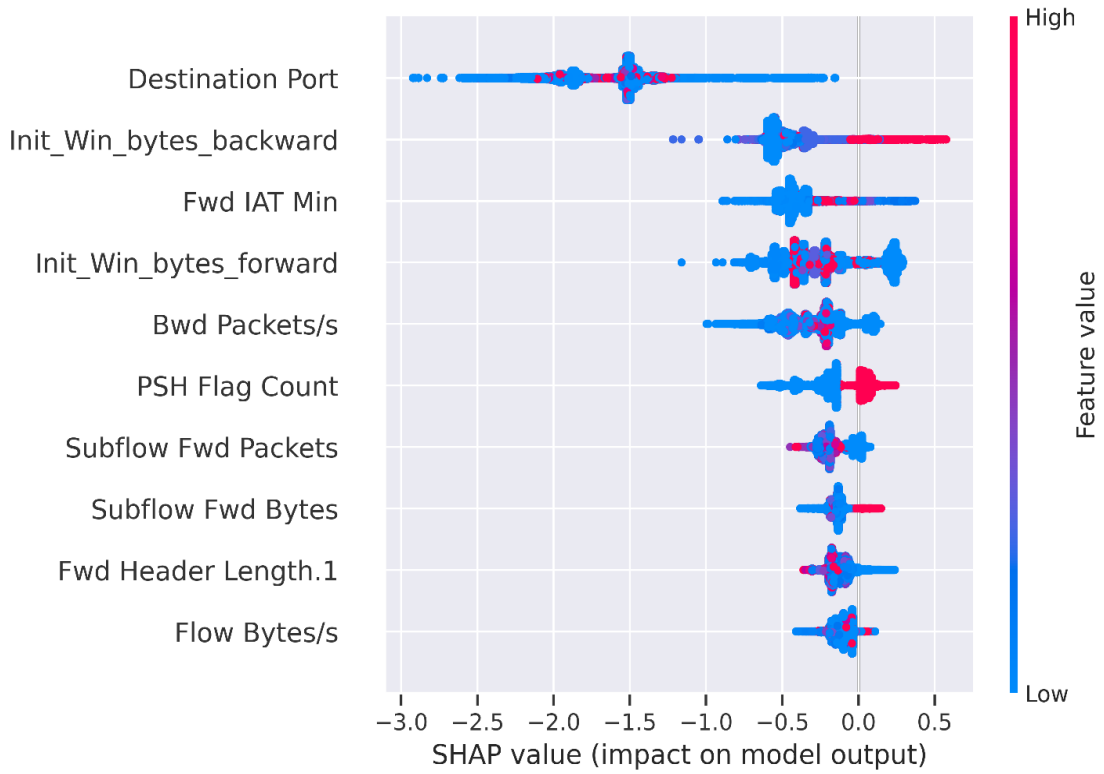


Figure 4.10 SHAP Beeswarm Plot Showing Feature-Wise Impact and Direction on Model Predictions

The brightest feature is the Destination Port which displays the broadest SHAP range of about -3.0 to +0.5. This implies that that difference in the destination port numbers has a significant impact on predictions on the model. The feature values of low features (blue points) are benign traffic whereas the prediction is strongly directed towards attack classes (high feature values, red points). The asymmetric diffusion of points proves that certain ports, which are often targeted during an attack, have an enormous positive impact on intrusion classification, thus its role in the model of the leading decision-maker. The Init_Win_bytes_backward feature is next with a range of SHAP values of approximately between -2.0 to +0.5 which is very important and just a bit smaller compared to that of the destination port. Large Backward TCPwindow sizes (red points) are likely to make the model more inclined towards labeling a flow as malicious. It happens that this relationship is indicating that the irregular window settings or abnormal buffer allocation which are typical of infiltration or brute-force attacks has a strong influence on the output of the model. Fwd IAT Min and Init wedge (forward) features are also

characterized by observed SHAP values spreads of about -1.5 to +0.5 with moderate but strong effect. Inter-arrival times (red points) are highly predictive of attacks on the basis that high inter-arrival times are usually an indicator of automated or high-frequency attack patterns. Equally, the bigger the size of forward TCP windows, the higher the likelihood of malicious identification since attackers tend to adjust the connection parameters in order to transmit their payloads faster. Another important flow-based measure that affects the model results is Bwd Packets/s, which has SHAP values of -1.2 to +0.3. When a packet rate is large and backward, it can typically represent a DoS or botnet traffic. On the contrary, normal, low-volume patterns of communication are associated with the low packet rates (blue points). The PSH Flag Count feature is not as widely spread as the top attributes but still makes a significant contribution and the SHAP values range between -1.0 up to +0.3. Large counts of PSH (red points) have the effect of biasing the prediction towards attacks, which is suggestive of how frequent forced data delivery (through TCP push flags) is a behavioural signal of aggressive intrusion attempts. Lower in the list, Subflow Fwd Packets, Subflow Fdw Bytes, Fwd Header Length 1 and Flow Bytes/s also have smaller SHAP ranges of approximately -.8 to +.2, although with smaller but cumulative significance. All these features are used as the structural and volumetric characteristics of network traffic. Higher values (red) usually contribute to an increase in the likelihood of an attack because they are associated with bursty or long-term traffic, whereas lower values (blue) are associated with normal user sessions or browsing the web.

4.12 Discussion

In this research, several ML frameworks, including XGBoost, CatBoost, ANN, and MLP, were developed and evaluated to identify cyber intrusion using the CIC-IDS 2017 dataset, and the explainability of the models, in particular, SHAP analysis was emphasized. The following discussion expounds on each of the key findings, provides a generalisation of finding with the overall literature and provide theoretical, practical, and policy implications.

The initial significant conclusion of the study is that the ensemble models (XGBoost and CatBoost) recorded the highest accuracy in intrusion detection, of about 100 percent, whereas the deep learning models (MLP and ANN) had about 9394 percent accuracy. This goes to show that gradient-boosting algorithms tend to perform better with complex, imbalanced network data than simple neural architectures. Ensemble models proved to be extremely effective in identifying common (e.g., DoS, Botnet) and uncommon (e.g., Web Attack, Infiltration) attack

classes and had good generalization to the dataset. The identified finding is consistent with a large amount of existing research that has placed an accent on the effectiveness of ensemble learning in the context of cybersecurity. In particular, Mokoele et al. (2024) discovered that tree-based ensemble techniques are more effective at intrusion detection than neural networks because it effectively processes noisy and heterogeneous data[75]. On an equal note, Fan et al. (2024) established that XGBoost was found to have better detection accuracy and less false positive rates compared to LSTM and CNN models when applied to the same dataset[76]. Nevertheless, the findings are opposite to the initial reports by Wanh et al. (2023) who showed better results of deep learning models, e.g. deep belief networks. Probably the difference is due to the size of data set and balance: deep models do better with large, uniformly distributed data, whereas boosting models do not collapse even when the distribution of classes is skewed[77]. The consequences of this discovery are great. Theoretically, it can strengthen the theory of ensemble learning that holds that a set of weak learners boosted to form an ensemble can improve generalization and anti-overfitting. In practice, it implies that both XGBoost and CatBoost should be used as effective, high-performing classifiers in real-time intrusion detection systems, which are equally accurate as deep networks at a lower cost of computation and training time. Policymaking wise, the interpretability and reliability of ensemble models render them best suited to high-security environments, including the finance, telecommunications, and defense industries, where the transparency and responsibility of an algorithm is necessary in the new governance structures like the EU AI Act.

The second observation relates to the recognition of the most significant elements in intrusion detection by means of the SHAP analysis. The analysis showed that the mean SHAP value of Destination Port (= 2.05) was the highest, then Bwd Packets/s(= 0.71), Init_Win_bytes_backward (= 0.68) and Fwd IAT Min (= 0.56) followed. These variables indicate fundamental aspects of network traffic which include port use, packet flow rate, window size, and packet timing which have a high correlation with cyberattack behavior. As an example, abnormal or fixed destination ports can be associated with port-scanning or brute-force attacks whereas high rates of backward-packet can mean that it is in a denial-of-service or botnet attack. This result confirms previous studies by Rodriguez et al. (2022) , which have found port-based and flow-level features to be the most promising ones that can be used to detect intrusion in network traffic[59]. Equally, Imtiaz et al (2025) demonstrated the interpretive ability of SHAP to predict attributes of transport layers like TCP window sizes and

inter-arrival times as reliable predictors of malicious behavior[78]. This paper elaborates on these results and makes quantitative measurements of the magnitude of each feature and presents them as SHAP plots. Contrarily, earlier models used non-directional and non-interpretable values of feature importance. This finding has both theoretical and practical implications. Theoretically, it confirms network flow theory, which shows that the transport-layer and temporal parameters are most crucial in intrusion behavior modeling. Concretely, consideration of these best characteristics can make real-world intrusion detection systems efficient and can monitor more quickly and with less resource consumption because only the most informative attributes will be considered. Policymaking-wise, definite dedication of decisive variables increases the auditability of a model, which facilitates regulatory adherence to explainable AI systems in cybersecurity, especially in areas where automated decisions need to be verifiable and understandable.

The third significant study result is that the best features that had a similar impact on all the six types of attacks, which are frequent and rare. Figure 4.9, SHAP class-wise summary plot, proved that these features had very high SHAP values in different classes, which means that the model can be well applied to various intrusion behavior types. This coinciding interpretability implies that the decision process of the model is not skewed at high-frequency types of attacks and can stand strong against underrepresented threats such as infiltration and web-based attacks. The same finding is also in line with the results presented by Ajagbe et al. (2024) who demonstrated that gradient-boosting models do not lose feature stability in dissimilar intrusion categories, which deep networks tend to specialize in one of the patterns [79]. It is also opposite to older intrusion detection systems Alimi et al. (2022) that were highly susceptible to class imbalance, resulting in reduced detection accuracy on rare attacks [80]. Ensemble learning has an interpretive benefit, as the implementation of SHAP in this study gave a more detailed insight into the contribution of each feature to the various types of attacks. The connotations are numerous. Theoretically, the uniformity in class-wise feature importance progresses the knowledge concerning multi-class interpretability, demonstrating that boosting based models embody generalizable network behaviour. Practically, the discovery is essential in the development of balanced intrusion detection systems that are able to detect low-frequency and high-frequency attacks, which in turn minimise the detection blind spots. Balanced feature importance, on the policy level, fosters fairness and responsibility in AI-based

cybersecurity tools, such that automated intrusion detection is not biased against critical threats that are rare.

The fourth observation was the use of the SHAP beeswarm plot that determined the impact of the direction and magnitude of single features on model forecasts. The visualization indicated that when the values of features were high especially those of Destination Port and InitWinbytesbackward, then the predictions were driven towards attack classes whereas when the values of the features were lower, then that was benign traffic. The SHAP values of Destination Port had a range of -3.0 to +0.5 and InitWinbytes back had a range of about -2.0 to +0.5 and Fwd IAT Min respectively indicating that they have a strong positive correlation with malicious activity. With this directional effect, there is a clear, interpretable insight on the manner in which the model is achieved. ranking feature importance instead of making decisions. These results are highly reflective of Salehiyan et al. (2025), who also found these directional effects in SHAP of network-based intrusion classifiers[81]. In contrast to classical methods of feature-ranking, which offer fixed scores of importance, the directional SHAP visualization has a dynamic display that indicates the changes in the outcome of the classification as a feature value is increased or reduced. It is this interpretive richness that enables an analyst not only to know what features are important, but their impact on model outputs in real network conditions. The consequences of this discovery are high. In practice, it can provide cybersecurity specialists with practical knowledge: e.g. the ability to create specific rules and prioritize the creation of rules early-on, when the likelihood of an attack is high, due to the fact that abnormal port values or higher Bwd Packets/s values directly increase the threat.

4.12.1 Limitations

Though, this thesis suggests an effective and explainable machine learning-based intrusion detection framework, it should be noted that the work has a number of limitations. The limitations do not disqualify the value of the results, but rather they establish parameters within which the results can be used. These constraints are also quite easy to understand and this can be useful in determining the future research direction and practical enhancement.

I. **Dependence on a Single Dataset (CIC-IDS2017):** The study is entirely based on the CIC-IDS2017 dataset, which, although modern and realistic, still represents only one specific set of network conditions, user behaviours, and attack scenarios. All models,

feature selection decisions, and performance evaluations are calibrated to the characteristics of this dataset. As a result, the generalisability of the findings to other network environments or newer datasets cannot be guaranteed. Real-world networks can differ in topology, traffic mix, protocol usage, and attack patterns, and models that perform strongly on CIC-IDS2017 may require retraining or adaptation before they can be reliably deployed elsewhere. This dataset dependency is a fundamental limitation that future work should address by validating the framework on multiple, diverse datasets and, ideally, on live network traffic.

II. **Limited GPU infrastructure and Range of Modelling Approaches:** Most of the tabular machine learning models do not support GPU acceleration, which prevented us from running all models on GPU-based environments. Another limitation is that although this research utilized a GPU, the available computational power was still insufficient for training some of the more resource-intensive models. The modelling scope of this thesis is confined to four supervised algorithms: XGBoost, CatBoost, Artificial Neural Network, and Multi-Layer Perceptron. Although these models represent two powerful families—gradient-boosted trees and fully connected neural networks—they do not cover the full spectrum of possible approaches. Other promising methods, such as SVM, KNN, CNN or RNN, graph-based models, or unsupervised and semi-supervised anomaly detectors, are not examined. As a result, the conclusions drawn about the relative strengths of ensemble models and neural networks apply only within the specific set of algorithms investigated here and may not hold universally across all types of IDS models.

III. **Lack of Real-Time Deployment and Offline Experimental Setting**: The experiments in this study are performed in an offline fashion, where stored data and batch training operations are used. The models have not been networked together and their performance in the presence of real-time constraints like throughput, latency and resource usage has not been tested. Practically, intrusion detection systems are required to stream through traffic, which may have rigid time constraints. Such aspects as the pace of feature extraction, inferring time of models, and the capability to cope with traffic bursts can play a considerable role in the viability in practice. As these operational factors are considered unexplored in this instance, there is still doubt on how the suggested framework would behave in the context of being implemented in a real network security framework or when implemented at scale.

IV. **Residual Impact of Class Imbalance:** Although the study employs strategies such as stratified splits and reducing the weight of class the effects of imbalance class, the underlying distribution of CIC-IDS2017 remains highly skewed. Some critical attack classes, especially infiltration and certain web attacks, have very few samples. Even with weighting, models are learning from limited information about these classes, which may restrict their robustness and generalisation when encountering similar but not identical attacks in practice. The performance reported for minority classes should therefore be interpreted with caution, and it is possible that more advanced imbalance-handling techniques, such as tailored resampling or cost-sensitive learning, could further improve detection of rare attacks.

4.13 Chapter Summary

This chapter has provided a detailed description of the empirical findings that were achieved using the framework of intrusion detection proposed and have explained their meaning. It started with the description of the result of the feature selection, which revealed that the narrow set of network attributes, including the traffic volume, timing properties, and port-related features, assumes a leading role in the discrimination of the benign traffic and the various kinds of attacks. The dimensionality reduction of the input space was shown to reduce the effective detection that can be obtained by not using all of the features, in this case, enhancing efficiency and interpretability.

Then made a comparison of the performance of XGBoost, CatBoost, ANN and MLP in one common experimental environment. The findings revealed that there are distinct disparities in the way these models perform the multi-class intrusion detection task, especially in situations of extreme imbalance of classes. Overall accuracy and high F1-scores, including to the minority classes, were overall high and the best-performing gradient-boosting approach, but not the neural network models, were found to perform well with majority traffic and less with rare types of attack. These trends were also demonstrated through confusion matrices, ROC curves, and radar plots, which demonstrated the models and classes where these misclassifications occurred and the trade-offs between detection and false alarms.

Lastly, the chapter used SHAP-based explainability on the most performing model and found out which features had the strongest effect on its decisions and how they varied across benign flows and each attack category. The SHAP analysis showed that the model used non-random

artefacts of the data, but meaningful and intuitively plausible indicators of malicious behaviour, hence making the model more likely to be applicable to practical use. The results and the interpretation taken as a whole indicate that the proposed framework can address the primary objectives of the research: it can detect multi-attacks of high quality on a realistic dataset, is capable of working with minority attacks in a more resilient way than the baseline options. Such results precondition the last chapter, summarising the overall value of the thesis and giving guidelines into further research and practical implementation.

CHAPTER 5
CONCLUSION AND FUTURE WORK

5.1 Conclusion

The study aimed at creating a transparent and interpretable explainable ML based intrusion detection system that would achieve high predictive performance. The paper aimed at comparing a few state-of-the-art algorithms, namely, XGBoost, CatBoost, ANN, and MLP on the CIC-IDS 2017 dataset to determine the most efficient method to detect network intrusions. By so doing, it also sought to research how explainable artificial intelligence methods, especially SHAP, might be incorporated into model evaluation to obtain a more in-depth look at the effect of various network characteristics on the classification procedure. The research has achieved these objectives through a systematic analysis and intensive experimentation, as it has come up with both robust empirical findings and valuable theoretical input to the subject of cybersecurity analytics.

As findings of the study provide, it is evident that ensemble learning models and in particular XGBoost and CatBoost are superior to traditional deep learning architectures in terms of accuracy and reliability. All performance measures of XGBoost model performed almost perfectly, which validates the fact that the model can be used in complex, high-dimensional, and imbalanced intrusion data. Conversely, the neural network-based models ANN and MLP, although able to identify the common types of attacks like DoS and Botnet, failed to assume consistency with the more infrequent ones (; Infiltration and Web Attack). The fact that SHAP analysis was also integrated only increased the model interpretability by exposing that such features had the most significant effect on the outcome of the detection. The SHAP summary and beeswarm plots demonstrated that difference in the values of these features caused a direct change in the values which are predicted by the model, which provided a clear and understandable picture of the process of decisions made. All these results prove that machine learning systems can be both highly accurate and highly interpretable, two attributes that are typically regarded as opposing in artificial intelligence studies.

The study presents some significant contributions to the general field of intelligent intrusion detection. It shows that ensemble learning, combined with interpretability methods, can create an effective system of reliable AI-based cybersecurity. The use of SHAP to add to the model evaluation mechanism makes the IDS a predictor of black-box to a predictable analytical coprocessor that can be used to in-in-time decision making and human cognition. In addition to the technical deliverables, the study will offer a repeatable framework to follow in future development of explainable, adaptive and data-efficient intrusion detection systems, which will be able to secure digital infrastructures in a more complex and adversarial cyberspace.

5.2 Future Work

Although this thesis has developed and evaluated an explainable machine learning–based intrusion detection framework with promising results, there are several ways in which the work can be extended and strengthened. Future research can build on the current findings to enhance generalisability, robustness, practicality, and human-centred aspects of intrusion detection. The following directions outline some of the most important and realistic opportunities for further development.

I. **Validation on Multiple Datasets and Real Network Environments:** One of the most direct extensions of this work is to validate the proposed framework on additional datasets and, ultimately, on live network traffic. Applying the same pre-processing pipeline, feature selection strategy, and model configurations to other benchmark datasets such as CSE-CIC-IDS2018 or recent IoT and industrial control system datasets would provide valuable evidence about how well the framework generalises to different network contexts, protocols, and attack profiles.

II. **Exploration of Additional Models and Hybrid Architectures:** Future studies can extend the range of modelling approaches beyond the four algorithms considered here. More advanced deep learning models, including CNN, RNN, LSTM, transformers, and graph-based architectures, offer alternative ways to represent and learn from network traffic. Hybrid architectures that combine gradient-boosted trees with neural networks, or integrate supervised classifiers with unsupervised anomaly detectors, may capture both known and novel attack patterns more effectively.

III. **Temporal and Context-Aware Intrusion Detection:** The current framework treats each network flow as an independent observation, without explicitly modelling

temporal relationships or broader contextual information. Future work could explore sequence-based and context-aware intrusion detection, where flows are analysed as part of sessions or time-ordered sequences. Techniques such as LSTM, GRU, or transformer models could be used to capture how attacks unfold over time, which is particularly important for multi-stage intrusions, low-and-slow scans, and complex campaigns that do not appear suspicious when viewed in isolation.

IV. **Further Dealing with Class Imbalance and Rare Attacks**: In this study, class-weighting and stratified splitting are applied to overcome the imbalance, but more advanced methods to deal with specific attacks (rare attacks) specifically can be studied in the future. Secondly, the researchers might consider producing more realistic synthetic attack traffic with the help of generative models, including GANs or variational autoencoders, in order to overrepresent the minority classes in a more reasoned manner. A well-crafted imbalance-handling mechanism can go a long way in enhancing the totting up of infrequent yet high-impact intrusions such as infiltration and specialised web attacks. attacks.

V. **Fine-Final Deployment, system integration and performance engineering:** The other relevant area of future work is to translate the framework into an offline experimental environment into a real-time, online IDS. This would include the creation and deployment of a pipeline capable of consuming network traffic in real time, extracting features in real time, and making model inferences with latency low enough. Architectural issues like batching strategy, model compression, hardware acceleration and load balancing are critical at this point. Practical deployment would also require the integration of the IDS into the existing SIEM systems, logging infrastructure and alert management workflows. Realistic workload-based performance testing such as stress testing and failure testing would give insight into the behaviour of the models under heavy throughput and resource constraints, and possibly the need to perform additional optimisations or structural modifications.

VI. **More explainable and Human-Centred Analysis of XAI:** Although SHAP is applied in this thesis to offer detailed explanations of the best behaviour model, the explain ability aspect can be further elaborated and expanded in the future. The first direction is to compare various XAI methods including SHAP, LIME, integrated gradients, and attention visualisation and to gain insight into their strengths and weaknesses as applied to IDS. The other is conducting user studies with security analysts to determine the level of understanding, usefulness, and trustworthiness of the various forms of

explanations in the actual process of undertaking investigations. Such a human-centred approach may be used when creating interfaces to explain information that do not overload or mislead the user by displaying them in suitable levels of detail or emphasizing the most useful information. Also, explainability can be applied to the development of the model itself, such as through XAI diagnostics to diagnose biases, overfitting to artefacts, or make decisions about feature engineering.

REFERENCES

[1] C. Yang, M. Gu, and K. Albitar, "Government in the digital age: Exploring the impact of digital transformation on governmental efficiency," *Technol Forecast Soc Change*, vol. 208, Nov. 2024, doi: 10.1016/j.techfore.2024.123722.

[2] Y. Yoo, "Computing in everyday life: A call for research on experiential computing," *MIS Q*, vol. 34, no. SPEC. ISSUE 2, pp. 213–231, 2010, doi: 10.2307/20721425.

[3] M. M. Alani, A. I. Awad, and E. Barka, "ARP-PROBE: An ARP spoofing detector for Internet of Things networks using explainable deep learning," *Internet of Things (Netherlands)*, vol. 23, p. 100861, Oct. 2023, doi: 10.1016/J.IOT.2023.100861.

[4] K. M. de Nobrega, A. F. Rutkowski, and C. Saunders, "The whole of cyber defense: Syncing practice and theory," *The Journal of Strategic Information Systems*, vol. 33, no. 4, p. 101861, Dec. 2024, doi: 10.1016/J.JSIS.2024.101861.

[5] A. Sharma, "THE IMPACT OF CYBERSECURITY BREACHES ON BIG BUSINESSES," *Int J Adv Res (Indore)*, vol. 12, no. 10, pp. 10–25, Oct. 2024, doi: 10.21474/IJAR01/19614.

[6] N. Moustafa and J. Slay, "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Information Security Journal*, vol. 25, no. 1–3, pp. 18–31, Apr. 2016, doi: 10.1080/19393555.2015.1125974.

[7] Z. Rehman, I. Gondal, M. Ge, H. Dong, M. Gregory, and Z. Tari, "Proactive defense mechanism: Enhancing IoT security through diversity-based moving target defense and cyber deception," *Comput Secur*, vol. 139, p. 103685, Apr. 2024, doi: 10.1016/J.COSE.2023.103685.

[8] M. A. Shyaa, N. F. Ibrahim, Z. Zainol, R. Abdullah, M. Anbar, and L. Alzubaidi, "Evolving cybersecurity frontiers: A comprehensive survey on concept drift and feature dynamics aware machine and deep learning in intrusion detection systems," *Eng Appl Artif Intell*, vol. 137, p. 109143, Nov. 2024, doi: 10.1016/J.ENGAPPAI.2024.109143.

[9] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches,"

Transactions on Emerging Telecommunications Technologies, vol. 32, no. 1, Jan. 2021, doi: 10.1002/ETT.4150.

[10] B. R.S, A. K.S, A. S.O, and I. R.M, "DEVELOPMENT OF AN INTRUSION DETECTION SYSTEM IN A COMPUTER NETWORK," *INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY*, vol. 12, no. 5, pp. 3479–3485, Jan. 2014, doi: 10.24297/IJCT.V12I5.2918.

[11] S. Jin, J. G. Chung, and Y. Xu, "Signature-based intrusion detection system (IDS) for in-vehicle CAN bus network," *Proceedings - IEEE International Symposium on Circuits and Systems*, vol. 2021-May, 2021, doi: 10.1109/ISCAS51556.2021.9401087.

[12] M. Al-Asli and T. A. Ghaleb, "Review of signature-based techniques in antivirus products," *2019 International Conference on Computer and Information Sciences, ICCIS 2019*, May 2019, doi: 10.1109/ICCISCI.2019.8716381.

[13] "(PDF) Comparison of Traditional vs. AI-Based Intrusion Detection and Prevention Systems: Efficiency and Accuracy." Accessed: Oct. 24, 2025. [Online]. Available: https://www.researchgate.net/publication/389717300_Comparison_of_Traditional_vs_AI-Based_Intrusion_Detection_and_Prevention_Systems_Efficiency_and_Accuracy

[14] "(PDF) Malware Attacks on Electronic Signatures Revisited." Accessed: Oct. 24, 2025. [Online]. Available: https://www.researchgate.net/publication/221307202_Malware_Attacks_on_Electronic_Signatures_Revisited

[15] C. Gupta, A. Kumar, and N. K. Jain, "Intelligent intrusion detection system based on crowd search optimization for attack classification in network security," *EURASIP J Inf Secur*, vol. 2025, no. 1, pp. 1–24, Dec. 2025, doi: 10.1186/S13635-025-00205-7/FIGURES/22.

[16] S. M. Sangve, "ANOMALY BASED IMPROVED NETWORK INTRUSION DETECTION SYSTEM USING CLUSTERING TECHNIQUES," *International Journal of Advanced Research in Computer Science*, pp. 808–815, Aug. 2017, doi: 10.26483/IJARCS.V8I7.4453.

[17] M. Ozkan-Okay, R. Samet, O. Aslan, and D. Gupta, "A Comprehensive Systematic Literature Review on Intrusion Detection Systems," 2021, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2021.3129336.

[18] S. Ranshous, S. Shen, D. Koutra, S. Harenberg, C. Faloutsos, and N. F. Samatova, “Anomaly detection in dynamic networks: A survey,” *Wiley Interdiscip Rev Comput Stat*, vol. 7, no. 3, pp. 223–247, May 2015, doi: 10.1002/WICS.1347.

[19] V. Kantharaju, H. Suresh, M. Niranjanamurthy, S. I. Ansarullah, F. Amin, and A. Alabrah, “Machine learning based intrusion detection framework for detecting security attacks in internet of things,” *Sci Rep*, vol. 14, no. 1, pp. 1–10, Dec. 2024, doi: 10.1038/S41598-024-81535-3;SUBJMETA.

[20] K. Alam, M. F. Monir, M. J. Hossain, M. Shorif Uddin, and M. T. Habib, “Adaptive Defense: Zero-Day Attack Detection in NIDS With Deep Reinforcement Learning,” *IEEE Access*, vol. 13, pp. 116345–116361, 2025, doi: 10.1109/ACCESS.2025.3585445.

[21] Md Boktiar Hossain and Khandoker Hoque, “Machine Learning approaches in IDS,” *International Journal of Science and Research Archive*, vol. 7, no. 2, pp. 706–715, Dec. 2022, doi: 10.30574/IJSRA.2022.7.2.0303.

[22] X. Liu and J. Liu, “Malicious traffic detection combined deep neural network with hierarchical attention mechanism,” *Sci Rep*, vol. 11, no. 1, pp. 1–15, Dec. 2021, doi: 10.1038/S41598-021-91805-Z;SUBJMETA.

[23] S. A. Alansary, S. M. Ayyad, F. M. Talaat, and M. M. Saafan, “Emerging AI threats in cybercrime: a review of zero-day attacks via machine, deep, and federated learning,” *Knowl Inf Syst*, pp. 1–37, Aug. 2025, doi: 10.1007/S10115-025-02556-6/TABLES/7.

[24] A. A. Ibrahim, R. L. Ridwan, M. M. Muhammed, R. O. Abdulaziz, and G. A. Saheed, “Comparison of the CatBoost Classifier with other Machine Learning Methods,” *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 11, pp. 738–748, 2020, doi: 10.14569/IJACSA.2020.0111190.

[25] A. Jovic, N. Frid, K. Brkic, and M. Cifrek, “Interpretability and accuracy of machine learning algorithms for biomedical time series analysis – a scoping review,” *Biomed Signal Process Control*, vol. 110, p. 108153, Dec. 2025, doi: 10.1016/J.BSPC.2025.108153.

[26] I. D. Mienye and N. Jere, “A Survey of Decision Trees: Concepts, Algorithms, and Applications,” *IEEE Access*, vol. 12, pp. 86716–86727, 2024, doi: 10.1109/ACCESS.2024.3416838.

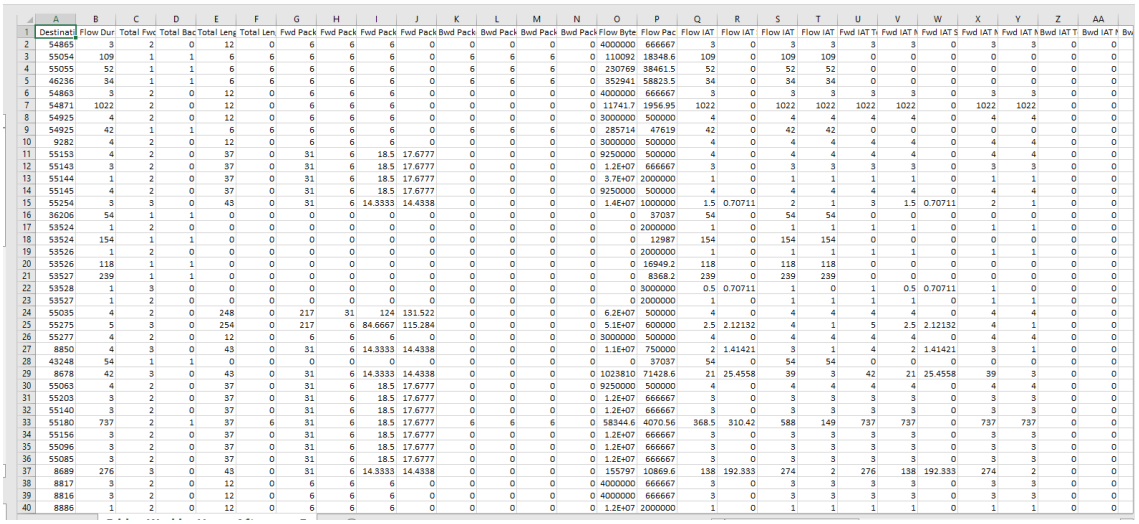
APPENDICES:

Appendix A: Intrusion detection evaluation dataset (CIC-IDS2017)

Dataset Link: https://www.unb.ca/cic/datasets/ids-2017.html

Dataset Sample:

Name	Status	Date modified	Type	Size
Friday-WorkingHours-Afternoon-DDos.p...		11/25/2025 8:52 PM	Microsoft Excel Co...	75,317 KB
Friday-WorkingHours-Afternoon-PortSca...		11/25/2025 8:52 PM	Microsoft Excel Co...	75,104 KB
Friday-WorkingHours-Morning.pcap_ISCX		11/25/2025 8:52 PM	Microsoft Excel Co...	56,950 KB
Monday-WorkingHours.pcap_ISCX		11/25/2025 8:52 PM	Microsoft Excel Co...	172,782 KB
Thursday-WorkingHours-Afternoon-Infilt...		11/25/2025 8:52 PM	Microsoft Excel Co...	81,155 KB
Thursday-WorkingHours-Morning-WebAt...		11/25/2025 8:52 PM	Microsoft Excel Co...	50,804 KB
Tuesday-WorkingHours.pcap_ISCX		11/25/2025 8:52 PM	Microsoft Excel Co...	131,914 KB
Wednesday-workingHours.pcap_ISCX		11/25/2025 8:52 PM	Microsoft Excel Co...	219,890 KB



Destinati	Flow Dur	Total Fwc	Total Bac	Total Leng	Total Len	Fwd Pack	Fwd Pack	Fwd Pack	Fwd Pack	Bwd Pack	Bwd Pack	Bwd Pack	Bwd Pack	Flow Byte	Flow Pac	Flow IAT	Flow IAT	Flow IAT	Flow IAT	Fwd IAT T	Fwd IAT N	Fwd IAT S	Fwd IAT N	Fwd IAT N	Bwd IAT T	Bwd IAT N	Bw
54865	3	2	0	12	0	6	6	6	0	0	0	0	0	4000000	666667	3	0	3	3	3	3	0	3	3	0	0	
55054	109	1	1	6	6	6	6	6	0	6	6	6	0	110092	18348.6	109	0	109	109	0	0	0	0	0	0	0	
55055	52	1	1	6	6	6	6	6	0	6	6	6	0	230769	38461.5	52	0	52	52	0	0	0	0	0	0	0	
46236	34	1	1	6	6	6	6	6	0	6	6	6	0	352941	58823.5	34	0	34	34	0	0	0	0	0	0	0	
54863	3	2	0	12	0	6	6	6	0	0	0	0	0	4000000	666667	3	0	3	3	3	3	0	3	3	0	0	
54871	1022	2	0	12	0	6	6	6	0	0	0	0	0	11741.7	1956.95	1022	0	1022	1022	1022	1022	0	1022	1022	0	0	
54925	4	2	0	12	0	6	6	6	0	0	0	0	0	3000000	500000	4	0	4	4	4	4	0	4	4	0	0	
54925	42	1	1	6	6	6	6	6	0	6	6	6	0	285714	47619	42	0	42	42	0	0	0	0	0	0	0	
9282	4	2	0	12	0	6	6	6	0	0	0	0	0	3000000	500000	4	0	4	4	4	4	0	4	4	0	0	
55153	4	2	0	37	0	31	6	18.5	17.6777	0	0	0	0	9250000	500000	4	0	4	4	4	4	0	4	4	0	0	
55143	3	2	0	37	0	31	6	18.5	17.6777	0	0	0	0	1.2E+07	666667	3	0	3	3	3	3	0	3	3	0	0	
55144	1	2	0	37	0	31	6	18.5	17.6777	0	0	0	0	3.7E+07	2000000	1	0	1	1	1	1	0	1	1	0	0	
55145	4	2	0	37	0	31	6	18.5	17.6777	0	0	0	0	9250000	500000	4	0	4	4	4	4	0	4	4	0	0	
55254	3	3	0	43	0	31	6	14.3333	14.4338	0	0	0	0	1.4E+07	1000000	1.5	0.70711	2	1	3	1.5	0.70711	2	1	0	0	
36206	54	1	1	0	0	0	0	0	0	0	0	0	0	0	37037	54	0	54	54	0	0	0	0	0	0	0	
53524	1	2	0	0	0	0	0	0	0	0	0	0	0	0	2000000	1	0	1	1	1	1	0	1	1	0	0	
53524	154	1	1	0	0	0	0	0	0	0	0	0	0	0	12987	154	0	154	154	0	0	0	0	0	0	0	
53526	1	2	0	0	0	0	0	0	0	0	0	0	0	0	2000000	1	0	1	1	1	1	0	1	1	0	0	
53526	118	1	1	0	0	0	0	0	0	0	0	0	0	0	16949.2	118	0	118	118	0	0	0	0	0	0	0	
53527	239	1	1	0	0	0	0	0	0	0	0	0	0	0	8368.2	239	0	239	239	0	0	0	0	0	0	0	
53528	1	3	0	0	0	0	0	0	0	0	0	0	0	0	3000000	0.5	0.70711	1	0	1	0.5	0.70711	1	0	0	0	
53527	1	2	0	0	0	0	0	0	0	0	0	0	0	0	2000000	1	0	1	1	1	1	0	1	1	0	0	
55035	4	2	0	248	0	217	31	124	131.522	0	0	0	0	6.2E+07	500000	4	0	4	4	4	4	0	4	4	0	0	
55275	5	3	0	254	0	217	6	84.6667	115.284	0	0	0	0	5.1E+07	600000	2.5	2.12132	4	1	5	2.5	2.12132	4	1	0	0	
55277	4	2	0	12	0	6	6	6	0	0	0	0	0	3000000	500000	4	0	4	4	4	4	0	4	4	0	0	
8850	4	3	0	43	0	31	6	14.3333	14.4338	0	0	0	0	1.1E+07	750000	2	1.41421	3	1	4	2	1.41421	3	1	0	0	
43248	54	1	1	0	0	0	0	0	0	0	0	0	0	0	37037	54	0	54	54	0	0	0	0	0	0	0	
8678	42	3	0	43	0	31	6	14.3333	14.4338	0	0	0	0	1023810	71428.6	21	25.4558	39	3	42	21	25.4558	39	3	0	0	
55063	4	2	0	37	0	31	6	18.5	17.6777	0	0	0	0	9250000	500000	4	0	4	4	4	4	0	4	4	0	0	
55203	3	2	0	37	0	31	6	18.5	17.6777	0	0	0	0	1.2E+07	666667	3	0	3	3	3	3	0	3	3	0	0	
55140	3	2	0	37	0	31	6	18.5	17.6777	0	0	0	0	1.2E+07	666667	3	0	3	3	3	3	0	3	3	0	0	
55180	737	2	1	37	6	31	6	18.5	17.6777	6	6	6	0	58344.6	4070.56	368.5	310.42	588	149	737	737	0	737	737	0	0	
55156	3	2	0	37	0	31	6	18.5	17.6777	0	0	0	0	1.2E+07	666667	3	0	3	3	3	3	0	3	3	0	0	
55096	3	2	0	37	0	31	6	18.5	17.6777	0	0	0	0	1.2E+07	666667	3	0	3	3	3	3	0	3	3	0	0	
55085	3	2	0	37	0	31	6	18.5	17.6777	0	0	0	0	1.2E+07	666667	3	0	3	3	3	3	0	3	3	0	0	
8689	276	3	0	43	0	31	6	14.3333	14.4338	0	0	0	0	155797	10869.6	138	192.333	274	2	276	138	192.333	274	2	0	0	
8817	3	2	0	12	0	6	6	6	0	0	0	0	0	4000000	666667	3	0	3	3	3	3	0	3	3	0	0	
8816	3	2	0	12	0	6	6	6	0	0	0	0	0	4000000	666667	3	0	3	3	3	3	0	3	3	0	0	
8886	1	2	0	12	0	6	6	6	0	0	0	0	0	1.2E+07	2000000	1	0	1	1	1	1	0	1	1	0	0	

Appendix B: IDS Implementation Code Snippets

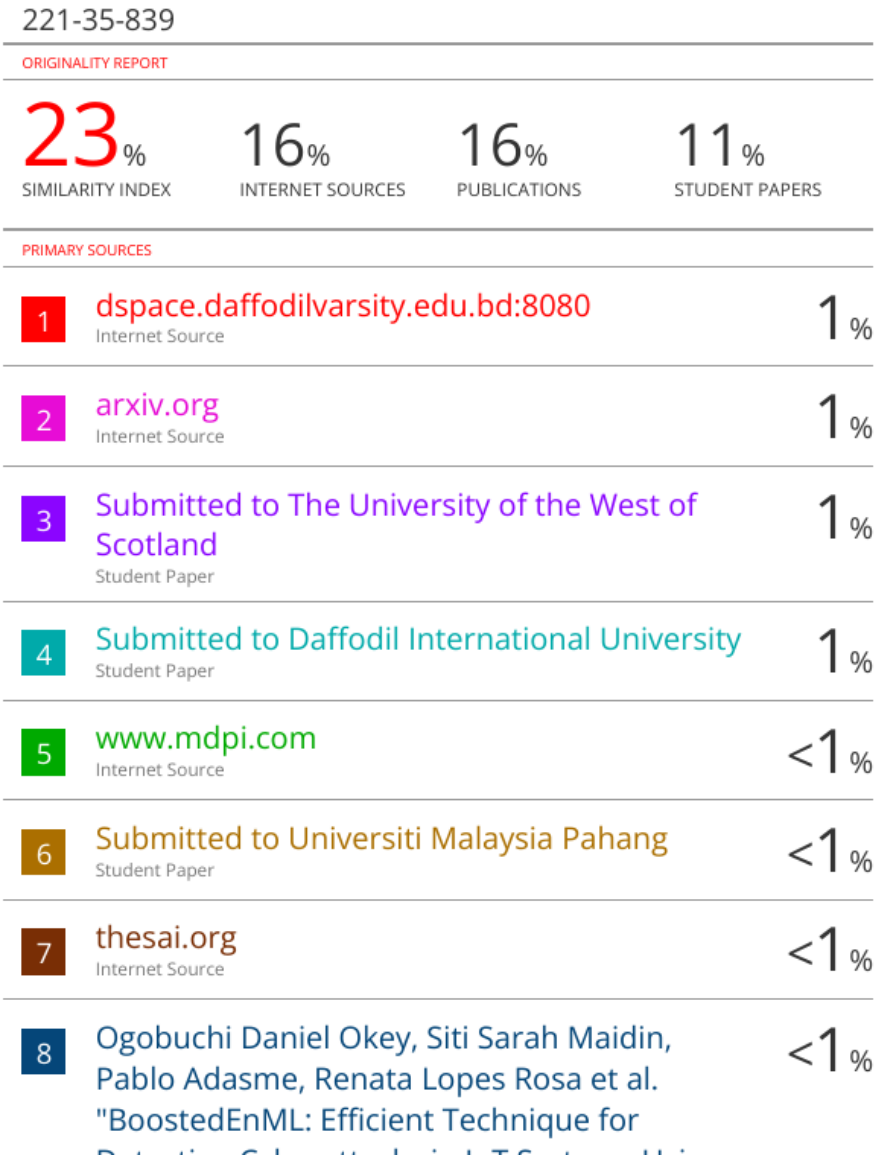
```
# ================================================================
# 📘 NETWORK INTRUSION DATASET PREPARATION PIPELINE
# ================================================================

# === 1️⃣ Import Libraries ===
import os
import numpy as np
import pandas as pd
import seaborn as sns
import missingno as msno
import matplotlib.pyplot as plt
from sklearn.utils import resample

# Set visualization style
sns.set(style='darkgrid')


# === 2️⃣ Load All Dataset Files ===
data_paths = [
    '/kaggle/input/network-intrusion-dataset/Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv',
    '/kaggle/input/network-intrusion-dataset/Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv',
    '/kaggle/input/network-intrusion-dataset/Friday-WorkingHours-Morning.pcap_ISCX.csv',
    '/kaggle/input/network-intrusion-dataset/Monday-WorkingHours.pcap_ISCX.csv',
    '/kaggle/input/network-intrusion-dataset/Thursday-WorkingHours-Afternoon-Infilteration.pcap_ISCX.csv',
    '/kaggle/input/network-intrusion-dataset/Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv',
    '/kaggle/input/network-intrusion-dataset/Tuesday-WorkingHours.pcap_ISCX.csv',
    '/kaggle/input/network-intrusion-dataset/Wednesday-workingHours.pcap_ISCX.csv'
]

data_list = [pd.read_csv(path) for path in data_paths]

print('📊 Individual Dataset Dimensions:')
for i, df in enumerate(data_list, start=1):
    print(f"Data{i}: {df.shape[0]} rows × {df.shape[1]} columns")

# === 3️⃣ Combine All Data ===
data = pd.concat(data_list, ignore_index=True)

rows, cols = data.shape
print('\n📈 Combined Dataset Dimension:')
print(f'Rows: {rows}')
```

```
# ================================================================
# 🧹 DATA CLEANING: Missing, Duplicate, and Infinite Values
# ================================================================

import pandas as pd
import numpy as np

# ================================================================
# 1️⃣ CHECK FOR MISSING (NaN) VALUES
# ================================================================
print("🔍 Checking for missing (NaN) values...")

# Count missing values per column
null_counts = data.isnull().sum()
null_cols = null_counts[null_counts > 0]

if not null_cols.empty:
    print("\n🔸 Columns with missing values:\n")
    print(null_cols)
else:
    print("\n✅ No missing values found.")

# Total missing values
total_missing = data.isnull().sum().sum()
print(f"\n🔸 Total missing values in dataset: {total_missing:,}")


# ================================================================
# 2️⃣ CHECK FOR DUPLICATE ROWS
# ================================================================
print("\n🔍 Checking for duplicate rows...")

duplicate_count = data.duplicated().sum()
print(f"🔸 Total duplicate rows: {duplicate_count:,}")

if duplicate_count > 0:
    perc = (duplicate_count / len(data)) * 100
    print(f"🔸 Percentage of duplicates: {perc:.2f}%")
```

Use Random Forest for feature selection

```
from sklearn.ensemble import RandomForestClassifier
# Initialize Random Forest
rf = RandomForestClassifier(n_estimators=100, random_state=678, n_jobs=-1)
# Fit the model
rf.fit(X_scaled, y)
# Get feature importances
importances = rf.feature_importances_
feature_names = X.columns  # original feature names
# Create a DataFrame for easy viewing
feature_importance_df = pd.DataFrame({
    'Feature': feature_names,
    'Importance': importances
})
# Sort features by importance
feature_importance_df = feature_importance_df.sort_values(by='Importance', ascending=False)
```

Plagiarism Report



221-35-839

ORIGINALITY REPORT

23%	16%	16%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	1%
2	arxiv.org Internet Source	1%
3	Submitted to The University of the West of Scotland Student Paper	1%
4	Submitted to Daffodil International University Student Paper	1%
5	www.mdpi.com Internet Source	<1%
6	Submitted to Universiti Malaysia Pahang Student Paper	<1%
7	thesai.org Internet Source	<1%
8	Ogobuchi Daniel Okey, Siti Sarah Maidin, Pablo Adasme, Renata Lopes Rosa et al. "BoostedEnML: Efficient Technique for	<1%

Account Clearance