DATA SCIENCE PROJECT

ENHANCING CYBERSECURITY RESILIENCE: LEVERAGING MACHINE LEARNING FOR ADVANCED THREAT DETECTION AND RESPONSE

CANDIDATE NUMBER: 277186

MSc Data Science - University of Sussex, Brighton, United Kingdom.

SUPERVISOR: DR. IMRAN KHAN

AUGUST, 2024

DECLARATION

I, Benjamin Ackah, hereby declare that this dissertation titled "Enhancing Cybersecurity Resilience: Leveraging Machine Learning for Advanced Threat Detection and Response" is my own work and that all the sources I have used or quoted have been indicated and acknowledged by means of complete references. This dissertation has not been submitted for any other degree or examination at any other university or institution.

I confirm that I have read and understood the university's guidelines on academic integrity, and I affirm that this work complies with these guidelines.

Signature: Benjamin Ackah

Date: 15 August 2024

DEDICATION

This work is dedicated to the memory of my beloved mother, Mary Kodwo Menlah Tayia. Your love, wisdom, and strength continue to inspire me every day. Though you are no longer with us, your guidance and values have been the foundation of all my achievements. I hope to make you proud.

ACKNOWLEDGEMENT

First and foremost, I would like to express my deepest gratitude to my supervisor, Dr. Imran Khan, for his invaluable guidance, support, and encouragement throughout this research. His expertise and insightful feedback have been instrumental in shaping this dissertation.

I am profoundly grateful to my wife, Paulina Armah, for her unwavering support, patience, and understanding throughout this journey. Your love and encouragement have been my source of strength.

To my wonderful children, Marcus Jeffrey Ackah, Lois Ackah Azane-Raale, and Eki Ogie Osayi, you have been my inspiration and motivation. Your love and smiles have made every challenge worthwhile.

I also extend my heartfelt appreciation to my noble family, whose love and support have been a constant source of comfort. Your belief in me has been a driving force behind my perseverance.

A special thanks to my dear friend Babatunde Oladimaji for your friendship, support, and encouragement during this journey. Your words of wisdom and motivation have been greatly appreciated.

This work would not have been possible without the support and encouragement of all the people mentioned above. I am truly grateful for everything you have done for me.

Thank you.

TABLE OF CONTENT

Declaration ……
Table of Contents

It appears the document you provided is extensive, and the Table of Contents continues beyond what was initially captured. I will now generate the complete Table of Contents with page numbers, excluding the sections you requested to be ignored.

Please give me a moment to extract the full content.

Here is the complete Table of Contents for your dissertation, excluding the sections you requested, and with page numbers added:

Abstract - Page vii

ACRONYMNS AND ABBREVIATION

- **RF**: Random Forest
- **XGB**: Extreme Gradient Boosting
- **DT**: Decision Trees
- **SVM**: Support Vector Machines
- **CICIDS**: Canadian Institute for Cybersecurity Intrusion Detection System
- **NB**: Naive Bayes
- **PCA**: Principal Component Analysis
- **VIF**: Variation Inflation Factor
- **RBF**: Radial Basis Function
- **DNN**: Deep Neural Networks
- **IDS**: Intrusion Detection Systems
- **UEBA**: User and Entity Behaviour Analytics
- **CNN**: Convolutional Neural Networks
- **RNN**: Recurrent Neural Networks
- **FSM**: Finite State Machine
- **OS**: Operating Systems
- **DoS**: Denial-of-Service
- **DDoS**: Distributed Denial-of-Service
- **API**: Application Programming Interface
- **CSV**: Comma-Separated Values
- **SHAP**: SHapley Additive exPlanations
- **LIME**: Local Interpretable Model-agnostic Explanations
- **FL**: Federated Learning
- **SHM**: Structural Health Monitoring
- **IP**: Internet Protocol
- **NSL-KDD**: Knowledge Discovery and Data Mining
- **NIDS**: Network Intrusion Detection Systems
- **Wi-Fi**: Wireless Fidelity
- **IoT**: Internet of Things
- **TPR**: True Positive Rate
- **FPR**: False Positive Rate
- **GBDT**: Gradient Boosted Decision Trees
- **XAI**: Explainable Artificial Intelligence
- **3ICT**: International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies
- **NOMS**: IEEE/IFIP Network Operations and Management Symposium
- **GCAT**: IEEE Global Conference for Advancement in Technology
- **TREC**: Text Retrieval Conference
- **BoW**: Bag-of-Words
- **HIDS**: Host-based Intrusion Detection Systems
- **UNSW-NB15:** University of New South Wales Network-Based 2015

Abstract

Cyber threats pose significant risks to states, organizations, and individuals, leading to data breaches, insecurity, and financial loss. The complexity of these evolving threats necessitates a shift to dynamic defenses and adaptive cybersecurity systems. Machine learning (ML) offers the potential for detecting dynamic cyber threats, identifying abnormalities, and recognizing malicious conduct within networks.

This study investigates ML techniques for enhancing cybersecurity, focusing on network traffic classification. A comparative analysis of supervised learning models, including Random Forest (RF), XGBoost (XGB), Decision Trees (DT), and Support Vector Machines (SVM), was conducted using two benchmark datasets: CICIDS2017 and UNSW-NB15. RF and XGB consistently outperformed other models, achieving high accuracy rates of 98.44% and 98.52%, respectively, on CICIDS2017 and 93.00% each on UNSW-NB15. DT and SVM also showed strong performance, with dataset-specific strengths.

These findings demonstrate the transformative potential of machine learning in cybersecurity, significantly improving threat detection and response. The high accuracy rates and lower false positive rates of RF and XGB contribute to more efficient resource utilization, enhancing the security posture of organizations. Integrating advanced ML techniques promises to strengthen digital ecosystems against sophisticated cyber threats and paves the way for future cybersecurity innovations.

Keywords: Cybersecurity, Machine Learning, Cyber threats, Supervised Learning

CHAPTER ONE
INTRODUCTION

1. Introduction

The proliferation of sophisticated cyber threats has substantially damaged the global economy, resulting in reputational harm and significant financial losses. According to projections from Houghton (2023), the financial impact of cybercrime is anticipated to skyrocket, with projected losses reaching $22.82 trillion annually by 2027—a surge from $3 trillion in 2015, as illustrated in Figure 1.0. This rising trajectory and our dependence on digital systems underscores the pressing need to address vulnerabilities in cybersecurity.

In response to this growing threat, machine learning (ML) has emerged as a cybersecurity weapon, capable of analyzing vast datasets, discerning intricate patterns, and enhancing threat detection and defense methods (Okoli et al., 2024). Integrating ML into cybersecurity transforms threat detection and response, strengthening the ability to swiftly detect, address, and mitigate evolving cyber threats (Nand Kumar et al., 2023).

However, despite the potential of ML, there remains a significant gap in understanding which ML techniques are most effective in different cybersecurity contexts and how to overcome the constraints of current implementations. This research explores the applications and constraints of ML in cybersecurity, emphasizing the development of novel techniques to enhance the security and resilience of digital ecosystems against advanced attacks.

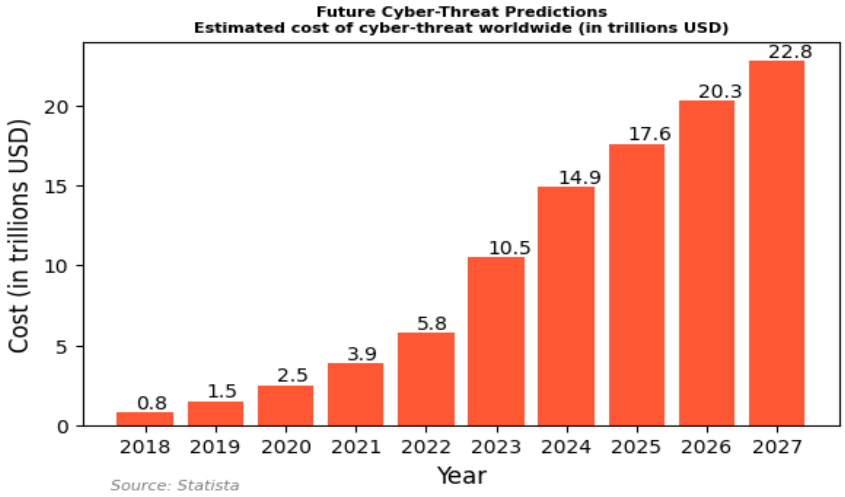


Figure 1.0. Estimated cost of future cyber-threat worldwide (in trillion US dollar) by 2027.

1.2. Research Background

Globally, digital connectivity has become indispensable. This offers individuals immense opportunities for political and socio-economic advancement; however, reliance on digital networks and systems creates systemic vulnerabilities that malicious actors eagerly exploit (Caleb and Thangaraj, 2023). Hackers, including individuals, sophisticated cybercriminal organizations, and state-sponsored groups, persistently target software, network, and system vulnerabilities for financial gain, data theft, and cyberwarfare (Ziolkowski, 2013).

Successful cyber-attacks can have devastating effects, including data breaches, financial harm, vital infrastructure disruption, and even risks to national security (Gulyás and Kiss, 2023). Notable incidents such as the WannaCry ransomware attack (Kao, D.-Y. et al., 2019), the Equifax

data breach (Smith and Mulrain, 2018), the SolarWinds supply chain attack (Coco, Talita, & Tsvetelina, 2022) and the recent cyber-attack on the 2024 Paris Olympics games have highlighted the vulnerabilities inherent in our digital systems, emphasizing the critical importance of implementing strong cybersecurity measures to combat these ever-changing threats.

Cybersecurity encompasses human actions and technologies, designed to safeguard electronic information resources (Zeadally et al., 2020). It covers various areas, including network security, application security, information security, and operational security (Ahsan et al., 2022).

As sophisticated attacks and ever-evolving techniques emerge, traditional security measures are no longer enough. This calls for an advanced mechanism in which Machine Learning (ML) comes into play, offering the potential to revolutionize cybersecurity by automating and enhancing many aspects of threat detection, prevention, and response like never before.

Machine learning offers numerous applications in cybersecurity, including network security, malware detection, intrusion detection and prevention, email and web security, user and entity behaviour analytics (UEBA), cyber threat identification, antivirus software, user behaviour modelling, combating AI threats, and threat intelligence gathering. ML techniques are essential for early detection and prediction of attacks, such as spam classification, enhancing cybersecurity measures across these domains (Raheja et al., 2022).

By harnessing ML, organizations can process vast amounts of data, deploy expert intelligence at scale, automate repetitive tasks, and enhance analyst efficiency. ML algorithms can process and analyze enormous volumes of data at speeds far beyond human capability, identifying patterns and anomalies faster, which is invaluable for predictive analytics (Sudhakar et al., 2022). These models can monitor network traffic for signs of unauthorized access, detect known and novel malware, and flag anomalies suggestive of insider threats or compromised accounts (Sabir et al., 2021).

Nevertheless, it's crucial to acknowledge the challenges and limitations of ML, such as the requirement for ample high-quality data, the trade-offs between true and false positives, explainability, repeatability, optimization for specific environments, and fortification against adversarial attacks.

Building upon previous research findings by Dasgupta, Akhtar, and Sen (2022) and Fraley and Cannady (2017), which highlight ML's role in securing cyberspace and identifying sophisticated threats, this research addresses key gaps in current cybersecurity practices. Specifically, it aims to develop and implement ML-based models tailored to diverse organizational contexts, focusing on intrusion detection and threat intelligence analysis. By addressing the challenges and limitations of ML, this study seeks to advance our understanding of its role in cybersecurity and provide practical insights for enhancing cyber defense strategies.

1.3. Motivation for This Research Work

The motivation for this research stems from the escalating importance of cybersecurity in today's interconnected digital environment. As global digital connectivity expands, so does the

threat landscape, with malicious actors targeting vulnerable software, networks, and systems. Recent high-profile incidents like the WannaCry ransomware attack and the SolarWinds supply chain breach underscore the critical need for robust cybersecurity measures adapting to increasingly sophisticated threats.

Traditional security measures, though essential, often struggle to keep pace with the ever-increasing sophistication and sheer volume of cyberattacks. This gap underscores the need for innovative solutions that can markedly improve capabilities in detecting, preventing, and responding to these threats. ML has emerged as a beacon of hope to address these challenges by automating processes, analyzing vast datasets, and swiftly identifying patterns indicative of cyber threats with unprecedented speed and accuracy.

This research aims to harness ML's potential in cybersecurity by developing and deploying specialized models for intrusion detection and threat intelligence. By rigorously evaluating the effectiveness of ML-based intrusion detection methods against traditional approaches, such as Intrusion Detection Systems (IDS) and Explainable AI (XAI), the study seeks to provide empirical evidence of ML's superiority in detecting and mitigating cyber threats.

Ultimately, this research strives to advance the understanding and practical application of ML in cybersecurity, offering actionable recommendations to enhance cyber defense strategies. By bridging theoretical insights with practical implementations, the study seeks to empower organizations to better protect sensitive data and critical digital infrastructures in an increasingly digital-dependent world.

1.4.The Significance of the Study

The significance of this study lies in its contribution to the cybersecurity literature, presenting an integrated framework that combines ML algorithms with established security systems. It identifies areas where ML can augment conventional security approaches by examining current threat scenarios and vulnerabilities. By evaluating the efficacy of various ML-based intrusion detection methods, including the LDSV algorithm, against traditional Intrusion Detection Systems (IDS) and Explainable AI (XAI) approaches, the study offers crucial insights into the effectiveness of these methods. This comparative analysis provides empirical evidence of the superiority of ML-based methods in cybersecurity, addressing key gaps in the literature.

The societal impact of this research is significant, as it enhances organizations' ability to combat cyber threats and protects individuals' sensitive data in our digital era. The study also paves the way for future research in cybersecurity, particularly in exploring the challenges and opportunities of integrating ML into current security systems (Wazid et al., 2022). Issues like data privacy and the need for continuous updates to ML models present avenues for further investigation. Additionally, refining ML-based intrusion detection methods and exploring the potential of specific algorithms, such as Support Vector Machines (SVM), are promising directions for future studies (Niu et al., 2020).

The practical implications of this study are significant, as organizations can implement ML-based intrusion detection systems to enhance security. However, the study also acknowledges

limitations, such as data privacy concerns and the need for ongoing updates to ML models, which must be addressed in future research (Khoje, 2024).
The interdisciplinary nature of this study, integrating machine learning and cybersecurity expertise, is essential for addressing complex cybersecurity challenges. The ethical implications are also emphasized, highlighting the importance of transparency, explainability, and addressing biases in ML algorithms (Al-Mansoori and Salem, 2023). This ethical dimension is pivotal in ensuring that ML systems are ethically designed and deployed.

In conclusion, this study contributes significantly to the cybersecurity literature by presenting an integrated framework that combines machine learning algorithms with established security systems. Its findings have practical implications for organizations, and its interdisciplinary approach and consideration of ethical implications make it a valuable contribution to the field.

1.5. Scope and Limitation

This study will focus on utilizing supervised learning techniques for network intrusion detection, specifically employing Support Vector Machine (SVM), Random Forest (RF), Extreme Gradient Boosting (XGB), and Decision Tree (DT) algorithms. The research will be conducted using two benchmark datasets, UNSW-NB15 and CICIDS2017, to rigorously evaluate the performance and effectiveness of these models in detecting network intrusions.

1.6. Research Questions and Objectives

1. How can machine learning models be utilized to enhance the accuracy of threat detection in network traffic for cybersecurity applications?

   Objective: To investigate the application of machine learning in detecting various cyber threats.

   The complexity of cyber threats requires advanced detection mechanisms. Machine learning models provide solutions for analyzing network traffic to detect anomalies. This research investigates the effectiveness of various machine learning techniques, including supervised and unsupervised learning, in enhancing threat detection accuracy. It will also analyze feature extraction methods to develop a robust framework for real-time threat detection in cybersecurity.

2. Which machine learning models are most effective in identifying and detecting cyber threats in network traffic?

   Objective: To compare the performance of different machine learning models in detecting cyber threats.

   Several machine-learning models, including Random Forest and Support Vector Machines, are proposed for cyber threat detection. This research aims to assess and compare their effectiveness in identifying threats within network traffic. By evaluating metrics such as accuracy, precision, recall, and F1-score, the study will determine which models excel at detecting different types of cyber threats. The findings will highlight each model's strengths and limitations, guiding the selection of the most effective approaches for cybersecurity.

3. How can ML-based threat detection systems be optimized to minimize false positives and false negatives while maintaining high detection accuracy?

   Objective: To optimize ML-based threat detection systems to minimize false positives and false negatives while maintaining high detection accuracy.

   This study aims to optimize ML models to reduce false positives—incorrectly identifying benign activities as threats—and false negatives—failing to detect actual threats. Balancing these is crucial for maintaining high detection accuracy in cybersecurity. By exploring techniques like feature selection, algorithm tuning, and ensemble learning, the research seeks to develop models that detect threats more accurately and minimize unnecessary alerts, thereby enhancing the efficiency and effectiveness of cybersecurity operations.

1.7. Research Organization

For research to be successful, the procedures involved in conducting the study are of utmost importance. This dissertation is organized into six chapters:

- **Introduction:** This chapter provides an overview of the background of the study, including its purpose, objectives, significance, scope and limitations.
- **Literature Review:** This section examines an overview of cybersecurity, cyber threats, and existing research on the topic, highlighting key findings and identifying gaps the current study aims to address, applications of ML within the cybersecurity industry.
- **Methodology:** This chapter details the research design, methods, and procedures to collect and analyze data. data pre-processing, ML models buildings with optimisation, and incorporation of evaluation metrics.
- **Results and Analysis:** This section presents key findings of the study, and analyzes the data collected.
- **Discussion:** This chapter interprets the results, discussing their implications and how they relate to the research questions and existing literature.
- **Conclusion:** This final chapter summarizes the study's findings, discusses their significance, limitations, and suggests directions for future research.

CHAPTER TWO
LITERATURE REVIEW

2. Introduction

Machine learning (ML) has significantly transformed the cybersecurity landscape, providing powerful tools to detect and prevent cyber threats. The widespread adoption of ML in cybersecurity has spurred the publication of hundreds of research papers, each proposing innovative solutions for various cybersecurity challenges.

This literature review synthesizes key research findings on developing and implementing ML-based models in cybersecurity, focusing on intrusion detection, and the challenges associated with transparency, privacy, and model bias. By integrating these insights, the review offers a comprehensive overview of how ML is shaping the future of cybersecurity and identifies areas where further research is needed.

Applications machine learning in intrusion detection cyber threats.

Anomaly detection is a critical component of cybersecurity, involving identifying unusual patterns that may signify a security breach. Ahmed et al. (2016) conducted a comprehensive survey of network anomaly detection techniques, highlighting the importance of these methods in identifying and mitigating potential threats. They also emphasized the challenges associated with anomaly detection, such as the scarcity of labelled data, high-dimensional data, and the need for real-time detection. These challenges have been emphasized in later studies that explore hybrid and ensemble approaches to anomaly detection, combining multiple algorithms to improve accuracy and effectiveness (Lok et al., 2020; Xu et al., 2021). However, the computational intensity and the need for large training datasets remain significant obstacles.

Rahul et al. (2022) introduced a three-stage deep learning framework for detecting network intrusions based on anomalies, underscoring the difficulty and cost of data labelling as significant obstacles. Similarly, Munir et al. (2019) developed an unsupervised anomaly detection system, DeepAnT, which leverages k-means clustering and principal component analysis (PCA) to reduce dimensionality, achieving promising results without needing labelled data.

While hybrid and ensemble methods show promise in enhancing detection accuracy, their scalability and computational demands remain significant challenges, particularly in real-time applications. Additionally, the reliance on labelled data continues to be a bottleneck, suggesting a need for more efficient models capable of operating effectively with minimal labelled data. Future research should focus on developing scalable and efficient models to address these challenges.

Intrusion Detection Systems (IDS) are crucial for detecting and monitoring malicious activities within a network. Studies such as those by Barnard et al. (2022) proposed a two-stage network intrusion detection system using Explainable AI (XAI), which effectively combines supervised and unsupervised methods. Their approach demonstrated high effectiveness in detecting new attacks while maintaining competitive performance on benchmark datasets like NSL-KDD.

Comparative analysis ML models in detecting cyber threats

The evolving nature of cyber threats has drawn researchers to compare the effectiveness of ML models in detecting and mitigating these threats.

Zhao et al. (2017) conducted a comparative study of ML-IDS and traditional IDS on the CICIDS2017 dataset, revealing that ML-IDS outperformed traditional systems in detecting unknown attacks, while traditional IDS performed better for known attacks. This finding highlights the potential of ML-IDS to enhance cybersecurity defenses, particularly against novel threats.

Aldweesh et al. (2020) reviewed deep learning approaches for anomaly-based intrusion detection systems, detailing the strengths and weaknesses of various architectures, datasets, and evaluation metrics. They emphasized the potential of deep learning models to improve IDS accuracy and efficiency by learning complex data patterns. However, they also noted ongoing challenges with the interpretability and explainability of these models, which are crucial for their real-world deployment in cybersecurity.

Moreover, deep learning architectures, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), are frequently used in intrusion detection due to their ability to learn complex patterns from data (Shone et al., 2018). Ensemble methods that combine multiple models, as explored by Ferrag et al. (2020) and Talukder et al. (2024), have also shown impressive accuracy, particularly in handling large and imbalanced datasets.

Recent studies have shown that ML techniques effectively improve cyber threat detection by reducing false positives and negatives.

Bold et al. (2022) assessed machine learning models, including Artificial Neural Networks (ANN) and Random Forests (RF), for ransomware detection, with the ANN model achieving the highest true-negative rate and a precision of 98.65%.

Rajora et al. (2023) combined Extreme Learning Machine (ELM) and Hidden Markov Models (HMM) for network intrusion detection, using the UNSW-NB15 dataset, and reduced the false positive rate by 10%, bringing it below 0.6%. Hari Gonaygunta (2023) employed logistic regression to detect cyber threats and minimize false positives in security operations centres.

Pham et al. (2018) employed ensemble techniques like Bagging and Boosting with tree-based classifiers to enhance IDS performance on the NSL-KDD dataset, improving detection rates and reduce false positives. They also applied feature selection to enhance accuracy and computational efficiency by reducing data dimensionality.

In their 2017 study, Almseidin et al. used a comparative approach to evaluate ML algorithms, including J48, Random Forest, Random Tree, Decision Table, MLP, Naive Bayes, and Bayes Network on the KDD-99 dataset. They assessed classifiers using accuracy, precision, recall, and F1-score, focusing on false negatives and false positives to enhance detection rates. Their results showed that the decision table classifier had the lowest false negative rate, while the random forest classifier achieved the highest average accuracy.

Optimization of ML-Based Threat Detection Systems

Optimizing machine learning models for intrusion detection is vital for reducing false positives and negatives while maintaining high accuracy. This section reviews studies focused on enhancing ML-based systems for improved threat detection.

Shen et al. (2018) proposed an ensemble pruning method optimized with the Bat Algorithm (BA) to enhance classifier performance on the KDD99, NSL, and Kyoto datasets. The study aimed to improve RF and SVM models by reducing false positives and false negatives. The methodology involved implementing BA to iteratively adjust hyperparameters, such as the number of trees in RF or the regularization parameter in SVM, to achieve optimal performance. The study demonstrated that BA significantly improved the accuracy and precision of RF and SVM, surpassing individual Extreme Learning Machines (ELMs) and reducing computational resources in intrusion detection systems (IDS). However, concerns were raised about the scalability and consistency of using metaheuristic algorithms across different datasets.

Aleesa et al. (2021) addressed the challenge of optimizing machine learning models to reduce false positives and negatives while maintaining high detection accuracy in IDS development. They fine-tuned deep learning models to balance detection accuracy and computational efficiency. Their results showed that the DNN model, with its multi-layer architecture, was particularly effective in minimizing false positives and negatives in multi-class classification.

Ustebay et al. (2018) explored feature selection and dimensionality reduction techniques to optimize machine learning-based intrusion detection systems. The study used PCA and Recursive Feature Elimination (RFE) to reduce the number of features in the CICIDS2017 dataset, aiming to enhance model performance and reduce computational complexity. PCA was applied to capture the most significant features, while RFE was used to remove irrelevant or redundant ones. The results showed that reducing the feature space significantly improved the accuracy of Deep Multilayer Perceptron (DMLP) model and reduced false positives. However, the study noted a trade-off between dimensionality reduction and retaining sufficient information to maintain high detection accuracy.

Kumari et al. (2023) proposed a solution that uses Support Vector Machine (SVM) algorithms optimized with Grid Search and cross-validation to enhance detection accuracy on the NSL-KDD dataset. By applying Principal Component Analysis (PCA) for feature selection, they reduced data dimensionality, which improved the performance of their SVM model. Their approach led to a significant reduction in false positives, with the SVM-based system achieving an impressive accuracy of 99.53%.

This literature review highlights the transformative potential of machine learning in cybersecurity, especially in intrusion detection systems. However, challenges like scalability, privacy preservation, bias mitigation, and transparency remain. Addressing these gaps could lead to more robust, adaptive, and trustworthy ML-based cybersecurity systems. This study builds on these insights by comparing the effectiveness of various ML models in detecting cyber threats, aiming to optimize performance while tackling the identified challenges.

2.1.Research Gaps and Opportunities

Despite significant advancements, several research gaps and opportunities persist in applying machine learning (ML) to cybersecurity. Addressing these gaps is crucial for developing more effective, adaptive, and trustworthy ML-based cybersecurity systems.

1. **To investigate the application of machine learning in detecting various cyber threats.**

   Identified Gap: Existing literature, including studies by Ahmed et al. (2016) and Rahul et al. (2022), highlights the transformative potential of machine learning in cybersecurity, particularly in anomaly detection and intrusion detection systems (IDS). However, significant challenges persist, such as the scarcity of labeled data, the high dimensionality of data, and the need for real-time detection. These challenges have not been fully addressed across the diverse range of cyber threats.

   Research Aim: This study aims to investigate the application of machine learning in detecting a wide range of cyber threats, focusing on overcoming these challenges. By exploring various ML techniques and their effectiveness in different cyber threat scenarios, the research will contribute to a deeper understanding of how to enhance the robustness and applicability of ML-based cybersecurity solutions.

2. **To compare the performance of different machine learning models in detecting cyber threats.**

   Identified Gap: While studies like those by Zhao et al. (2017) and Aldweesh et al. (2020) have explored the effectiveness of various ML models in intrusion detection, there remains a gap in the comparative analysis of these models across different environments and threat types. The literature often lacks a thorough comparison of model performance, particularly in data-rich versus data-poor scenarios, and the scalability of these models remains underexplored.

   Research Aim: This research seeks to fill this gap by systematically comparing the performance of different machine learning models in detecting cyber threats. By analyzing their effectiveness in varying conditions, the study will identify the strengths and weaknesses of each model, providing valuable insights for selecting the most appropriate ML techniques in different cybersecurity contexts.

3. **To optimize ML-based threat detection systems to minimize false positives and false negatives while maintaining high detection accuracy.**

   Identified Gap: The literature, including studies by Bold et al. (2022) and Shen et al. (2018), highlights the challenges of optimization of machine learning models for intrusion detection systems, especially in balancing detection accuracy with the rates of false positives and false negatives. While some optimization techniques have been proposed, further research is needed to refine these models for effective deployment in real-world scenarios.

Research Aim: This study aims to optimize ML-based threat detection systems by developing techniques that minimize false positives and false negatives while maintaining high detection accuracy. The research will explore advanced optimization methods to enhance the practical usability and reliability of these systems, ensuring they can effectively support cybersecurity operations with minimal disruptions.

Addressing these gaps is crucial for advancing the field of ML-based cybersecurity. Future research should focus on developing scalable, interpretable, and fair models that can be integrated into existing infrastructures and adapted to evolving threats. By doing so, the potential of ML in transforming cybersecurity can be fully realized.

2.2. Overview of Cybersecurity

In the 1970s, with the creation of ARPANET, the earliest form of the internet, hackers began to explore its weaknesses to launch attacks. This prompted researchers to think of security measures to protect the vulnerabilities in these networks. Notable events included the creation of the first computer virus, Creeper, by Bob Thomas, and its subsequent removal by the first antivirus software, Reaper, developed by Ray Tomlinson (Wikipedia) (Dakota Murphey, 2019). Cybersecurity was first introduced in the early 2000s when the scale of threats began to grow significantly, infiltrating major corporations and government organizations and stealing data. *Definition:* It involves the application of technology, processes, and controls to protect systems, networks, programs, and devices from cyberattacks. This protection aims to lessen the risk of cyberattacks and safeguard against unauthorized exploitation of networks, systems, and technologies (CybHER, 2021) (Codecademy). As a complex field, cybersecurity uses multiple layers and components to protect information, systems, and networks from various threats. Notable key components include network security, application security, data encryption and endpoint security (Darko et al., 2017).

2.2.1. Cybersecurity Threats

According to IBM, 2024, a cyber threat indicates a hacker or malicious actor is attempting to gain unauthorized access to a network to launch a cyberattack to disrupt the network or damage data. These threats include a wide range of attacks from data breaches to computer viruses, denial of service, and numerous other attack vectors.

Common cyber threats include malware, social engineering and phishing, intrusion attempts, denial-of-service (DoS) attacks, Man-in-the-Middle (MITM) attacks and insider threats. Each threat poses unique challenges, but machine learning techniques are the secret weapon that can outsmart and outmanoeuvre these threats, providing a powerful defense against even the most sophisticated attacks.

- *Malware* - short for "malicious software", is intentionally written to harm a computer system or its users. It is commonly used to steal personal, financial, or business information and can even wipe files critical to the operating system (CyberArk & IBM, 2024). Common types of malwares include ransomware, trojan horses, worms, and spyware.
- *Social engineering and phishing* - Social engineering involves manipulating individuals to take actions that lead to the exposure of confidential information, financial harm, or other security breaches. Examples include domain name spoofing (DNS).

However, phishing tricks individuals using deceptive emails, attachments, and text messages into divulging personal data or login details, downloading malware, transferring money to criminals, or engaging in other activities that may result in cybercrime (Smith, 2020). Spear phishing, whale phishing and business email compromise (BEC) are examples of phishing.

- *Man-In-The-Middle* (*MITM*) *Attack* – As the name suggests, this attack secretly relays messages between two users either to eavesdrop or to impersonate one of the parties to steal personal information, such as login credentials, account details and credit card numbers.
- *Denial-Of-Service (DoS) Attack* – A Denial of Service (DoS) attack overwhelms a website, application, or network, severely degrading its performance or making it inaccessible to legitimate users. This attack floods the target with excessive fraudulent traffic or requests, exhausting its resources and preventing normal operations. This can significantly affect an organization by causing service disruptions, financial losses and data corruption.
- *Zero-day Exploit* – This attack is a vulnerability that takes advantage of an unknown, unaddressed, or unpatched security flaw in computer software, hardware, or firmware. These exploits become well-known vulnerabilities when they are discovered and attacked for the first time (Kumar & Animesh, 2014).

2.3. Applications of Machine Learning in Cybersecurity

Machine learning (ML) and cybersecurity convergence have garnered considerable attention in contemporary discourse. As the landscape of cyber threats becomes progressively intricate and pervasive, conventional rule-based security protocols must be revised to provide comprehensive protection. Machine learning offers adaptive, scalable, and efficient solutions for detecting and mitigating these threats. Researchers have explored various ML techniques to enhance cybersecurity mechanisms, addressing issues such as intrusion detection, malware classification, phishing detection, and anomaly detection.

2.3.1. Malware Detection And Analysis

In cybersecurity, the ongoing battle against malware is a significant challenge. Host-based Intrusion Detection Systems (HIDS), including signature-based antivirus software, play a crucial role in detecting threats at the individual device level. As malware evolves to exploit vulnerabilities across various operating systems (OS) in computers, smart devices, and extensive networks, driven by advances in information technologies (Gorment et al., 2023), effective defense strategies are essential. Techniques such as feature engineering, classification algorithms, and model evaluation metrics are employed to differentiate between benign and malicious software, helping to combat these evolving threats.

According to Kolosnjaji et al. (2018) and Talukder and Talukder (2020), feature extraction and representation are essential for applying machine learning to malware detection. Key features include static attributes like binary file headers and imported libraries, as well as dynamic characteristics such as system call patterns, network traffic, API calls, opcode sequences, byte-level n-grams, and code structure (Sikorski & Honig, 2012).

Cakir (2018) used a shallow deep learning-based feature extraction method with the Gradient Boosting algorithm for malware detection, achieving up to 96% accuracy. Similarly, Khammas et al. (2015) found that combining PCA feature selection with SVM classification yielded the

highest accuracy with the fewest features, outperforming other feature selection-classifier combinations in static malware detection via n-gram analysis.
In a more recent study, Singh et al. (2022) enhanced an SVM-based ML model for malware detection through data preprocessing techniques, using transformation, outlier identification, filling, and smoothing to the CLaMP dataset. This preprocessing improved the model's accuracy by 7.95% for the SVM linear kernel and 3.19% for the polynomial kernel.
According to Kaspersky (2023), Android's open-source nature makes it vulnerable to malware attacks, exploiting its flexibility and openness. Unlike iOS, Android's modifiability can introduce security vulnerabilities, creating an environment conducive to hacking. This increases the likelihood of successful malware attacks on Android devices. However, Dai, Guqian et al. (2015) developed an SVM-based detection scheme for Android apps that leverages risky permission combinations and vulnerable API calls as features. This innovative approach demonstrated effectiveness in preliminary tests, identifying and mitigating malware threats in Android applications. Similarly, Westyarian et al. (2015) proposed a method for detecting malware on Android smartphones using API classes. They classified 412 applications (205 benign and 207 malicious) using RF, J48, and SVM models, achieving an average classification precision of 91.9%, showcasing the effectiveness of their approach in distinguishing between benign and malicious applications. Additionally, Yerima et al. (2016) developed proactive machine-learning approaches based on Bayesian classification to uncover unknown Android malware through static analysis, demonstrating high detection accuracy. In separate studies, Rana et al. (2018) and Mariam et al. (2017) also proposed using RF and SVM for malware detection on Android phones, with both classifiers achieving high accuracy, and RF slightly outperforming SVM. This consistency in results highlights the effectiveness of these ML algorithms in detecting malware on Android devices.
Deep learning approaches, particularly CNNs and RNNs, have also been employed for malware detection and analysis. These models can learn complex patterns and features from raw data, such as binary code or network traffic, without the need for extensive feature engineering (Vinayakumar et al., 2019).

2.3.2. Intrusion Detection

Network Intrusion Detection (NID) systems detect harmful activities in a network activity, leading to confidentiality, integrity, or availability of systems. Many intrusion detection systems leverage machine learning techniques to identify deviations from normal behaviour due to their adaptability to new and unknown attacks (Ford & Siraj, 2014).

Yuan et al. (2017) employ deep learning models for detecting Distributed Denial of Service (DDoS) attacks. Their approach leverages deep neural networks (DNNs) to analyze network traffic patterns and identify anomalies suggestive of DDoS activities. The architecture entails training a DNN on labelled data comprising normal and attack traffic, empowering the model to classify new instances based on acquired characteristics. Meanwhile, Ashing et al. (2018) highlighted using autoencoders for network anomaly detection. Autoencoders, an unsupervised learning model, compress input data into a latent space representation and reconstruct the input from this representation. Anomalies are detected based on the reconstruction error, expressed

as $E = \|x - \hat{x}\|^2$, where x represents the input data and $\hat{x}$, the reconstructed data and a high reconstruction error signifies a potential anomaly.

Tang et al. (2018) delves into the use of Deep Recurrent Neural Networks (RNNs) for intrusion detection in Software-Defined Networking (SDN) environments. RNNs, especially Long Short-Term Memory (LSTM) networks, excel in capturing temporal dependencies in sequential data, making them well-suited for analyzing network traffic over time. Consequently, the model's ability to maintain long-term dependencies and recognize patterns over sequences enhances its effectiveness in intrusion detection.

In contrast, Chowdhury et al. (2017) proposed a hybrid intrusion detection method that combines clustered traffic profiles with deep neural networks. Initially, the method clusters network traffic data to identify typical patterns and then employs a DNN to classify traffic based on these patterns. This hybrid approach enhances detection accuracy by leveraging unsupervised clustering and supervised deep learning techniques.

Furthermore, Nguyen, Phuoc-Cuong et al. (2021) present a significant advancement in Intrusion Detection System (IDS) technology with a lightweight and effective feature selection algorithm. This algorithm combines the strengths of Random Forest and AdaBoost to address network feature diversity, enhancing detection accuracy and efficiency.

In a related vein, Alom et al. (2019) provided a comprehensive survey on deep learning theory and architectures, highlighting their relevance across various domains, including cybersecurity. The survey emphasizes the significance of model architecture and training strategies in achieving high performance in intrusion detection tasks.

2.3.3. Phishing Detection

Phishing is a dangerous attack targeting individuals, organizations, and nations through social engineering tactics, including scam emails, text messages, and malicious websites designed to steal sensitive information. Machine learning techniques, especially Support Vector Machines (SVMs), have proven effective in detecting phishing.

Malar et al. (2020) created an anti-phishing system using lexical features and host properties, trained on over 10,000 URLs and achieved over 90% accuracy with various SVM kernels. Zouina and Outtaj (2017) developed a lightweight phishing detection system using SVM and a similarity index, reaching 95.80% accuracy. The use of Hamming distance, $d(s_1, s_2) = \sum_{i=1}^{n} \left(s_1[i] \neq s_2[i] \right)$, where, $s_1[i]$ and $s_2[i]$ represent the ith feature in the two URLs or feature vectors, significantly improved the recognition rate. This system requires only six URL features, which makes it suitable for resource-constrained devices like smartphones, unlike more complex systems.

Niu et al. (2017) demonstrated the effectiveness of Support Vector Machine (SVM) in classification tasks but noted that default kernel parameters can limit its accuracy, especially in phishing email detection. To address this, they proposed a hybrid classifier, Cuckoo Search SVM (CS-SVM), which integrates Cuckoo Search with SVM to optimize Radial Basis Function (RBF) parameters. The results show that CS-SVM outperforms default SVM, achieving a high accuracy of 99.52% in phishing email detection. Wang Ming-zheng (2011) and Bireswar Banik, Abhijit Sarma (2018) independently proposed Support Vector Machine

(SVM) models for detecting phishing websites, relying solely on URL features. Notably, the latter achieved a high accuracy rate of 96.35%.
Hybrid models enhance phishing detection accuracy by combining multiple machine-learning techniques. Altwaijry et al. (2024) demonstrated that a hybrid model using CNN and Bi-GRU achieved 100% precision, 99.68% accuracy, 99.66% F1 score, and 99.32% recall for phishing detection. Salahdine et al. (2021) proposed a technique with SVM, LR, and ANN, achieving fast and accurate phishing detection. Subashini and Narmatha (2023) introduced an ensemble model combining ANN, Random Forest Classifier (RFC), and SVM, reaching a 97.3% detection rate for phishing websites. Beh and Lim (2024) showed that their hybrid model, combining URL-based and content-based methods, achieved 95.3% accuracy, with a False Positive Rate of 5.4% and a False Negative Rate of 3.9%. Their ensemble model also performed well, with 94.7% accuracy, 5.9% False Positive Rate, and 4.7% False Negative Rate.

Deep learning has also proven effective for phishing detection. Kaushik (2023) and Adebowale (2019) used CNN and LSTM networks, achieving accuracy rates of over 95% and 93.28%, respectively. Zhang et al. (2018) introduced a hybrid model combining an autoencoder with CNN, achieving a mean accuracy of 97.68%. Patil (2023) explored various ML and deep learning algorithms, with XGBoost reaching 86.8% accuracy. Atawneh and Aljehani (2023) used public datasets and deep learning models, achieving 99.61% accuracy with a BERT-LSTM model.

2.3.4. Spam Detection

Spam detection has evolved from basic user-defined filters to sophisticated machine learning algorithms, addressing spam in emails, blogs, search engines, tweets, and videos. Initially, spam filters used user-defined rules, easily bypassed by spammers through content obfuscation. Spam detection typically involves classifying emails as either spam or legitimate, treated as a binary classification problem.

According to Kumar et al. (2020), spam refers to unsolicited commercial messages sent in bulk to numerous recipients. Such messages often aim to deceive users into divulging personal information or promoting advertisements. For instance, a recent spam email in my inbox reads, "Congratulations – You've won $67,540! Please enter your bank credentials to claim this amount."
Recent advancements in machine learning have significantly improved the accuracy and efficiency of spam detection methods.
The Naive Bayes classifier stands out as a widely adopted algorithm for spam filtering, leveraging probabilities to determine whether a message is legitimate. These classifiers often rely on a bag-of-words (BoW) model, which simplifies the representation of words in messages and is widely used in document classification. Naive Bayes operates on Bayes' theorem and assumes that the presence of a particular feature in a class is independent of the presence of any other feature. The formula for calculating the probability that an email E belongs to class C (spam or not spam) is: $P(C|E) = \frac{P(C)\prod_{i=1}^{n} P(E_i|C)}{P(E)}$, where $P(C)$ is the prior probability of

class C, $P(E_i|C)$ is the likelihood of feature E_i given class C, and $P(E)$ is the evidence or total probability of the email E.

Wei, 2018 explores the Naive Bayes Classifier's role in email servers, focusing on its mathematical process and effectiveness in spam filtering. The study underscores its simplicity and efficacy in real-world applications. Sharma et al. (2014) assesses the efficacy of Naive Bayes and MLP classifiers in detecting spam emails using the Trec07 dataset. Their study introduces an anti-spam filtering technique using data mining and machine learning methodologies to differentiate between spam and legitimate (ham) emails within the dataset. The research highlights the effectiveness of Bayesian Network-based solutions. Maram, S.V. (2021) demonstrated that the Naive Bayes algorithm achieved 98.13% accuracy in classifying SMS messages as spam or ham using the SMS spam collection dataset, proving effective in detecting and filtering SMS spam. Similarly, Paper et al. (2015) proposed a Multi Naive Bayes Classifier for classifying malicious emails, which outperformed other classifiers in accuracy, false positive rate, precision, and recall, showcasing its potential as a reliable tool for email classification.

Despite the Naïve Bayes classifier being the most efficient for spam detection, other ML algorithms such as SVMs, random forest, etc., have proved effective. Zahra S. Torabi et al. (2015) explored using Support Vector Machines (SVM) for spam detection and classification. They improved SVM's performance by combining it with other algorithms, dimension reduction techniques, and kernel functions, highlighting its effectiveness in pattern recognition and classification tasks that require precise categorization into distinct classes. Sonare et al. (2023) developed the Multi-Layer Perceptron (MLP) algorithm on email datasets to detect and classify emails as spam or non-spam, achieving approximately 98% accuracy. In their study, Ketcham et al. (2023) proposed a Random Forest-based classification method to distinguish spam from non-spam messages by removing duplicates and applying quantitative transformations. Using a machine learning approach, they evaluated various models and found that Random Forest achieved the highest accuracy of 97% in classifying text messages as spam or non-spam.

In recent years, social media platforms, including X (formerly Twitter), Facebook, Instagram, and others, have become conduits for spammers to launch the attacks. According to Statista (https://www.resolver.com/blog/navigating-the-surge-of-spam-on-social-media/), Resolver AI systems detected over 10 million individual spam messages across 110 brand accounts on social media between January 1st and August 1st, 2023. This amounts to approximately 47,000 spam messages per day during that period. Sumathi and Raja (2023) and Sahane (2024) investigated the surge in spam on social media platforms and highlighted two primary detection approaches: machine learning (ML) and expert-based detection. While expert-based detection is time-consuming and heavily reliant on specialized knowledge, the authors preferred ML-based detection for Online Social Networks (OSNs). They utilized various ML algorithms, including Logistic Regression, Decision Trees, and Support Vector Machines, to address imbalanced data distributions and proposed a Voting Classifier achieving a high accuracy of 97.96%. Meanwhile, Sumathi and Raja (2023) developed a dedicated website for spam detection.

In contrast, Zhang et al. (2020) introduced a different approach with their Improved Incremental Fuzzy-Kernel-Regularized Extreme Learning Machine (I2FELM) algorithm. This

method effectively detects Twitter spam in unbalanced datasets using a minimal number of features. Validation results confirmed the algorithm's accuracy in identifying Twitter spam.

2.3.5. Network Anomaly Detection

The rise in network attacks threatens system security and availability, making network anomaly detection systems crucial for monitoring deviations from established protocols. These systems are widely used to identify unknown attacks and malicious activities (Nawir et al., 2019; Stefano Leggio, 2023).

Casas et al. (2016) proposed using decision trees for detecting and classifying network traffic anomalies in cellular networks, comparing them with SVM and neural networks. They evaluated their approach with synthetic data from cellular ISP benchmarks and recommended a multi-detector strategy to improve performance. Bernieri et al. (2019) assessed various machine-learning models for anomaly detection in industrial control networks using data from the Secure Water Treatment (SWaT) testbed. They explored supervised and unsupervised methods, highlighting their strengths and limitations. Nawir et al. (2019) used supervised ML for network anomaly detection to reduce communication costs and bandwidth. Their experiments with the UNSW-NB15 dataset identified AODE as the best algorithm, achieving 97.26% accuracy with a 7-second processing time. They also explored distributed algorithms, which, while addressing centralization issues, had slightly lower accuracy and longer processing times.

2.3.6. Bot and Botnet Attacks Detection

A bot (robot) attack involves automated scripts to disrupt websites, steal data, make fraudulent purchases, or perform other malicious actions. These attacks can target websites, servers, and APIs. On the other hand, botnets (robot networks) are hijacked computer devices used to execute various scams and cyberattacks such as DDoS attacks, spamming, malware campaigns, and breaking down large networks (Ismail et al., 2020). While bot attacks aim to steal sensitive information and damage infrastructure, botnets exploit your devices to scam others or cause disruptions without your consent. Bot attacks can devastate businesses, causing significant downtime, lost revenue, and reputational damage.

Daya et al., (2020) employed BotChase, a robust system that transforms network flows into an aggregated graph model, to detect bot attacks. BotChase uses two machine learning phases, Self-Organizing Maps and Deep Learning, to differentiate bots from benign hosts, achieving high true and low false positives. It detects bots using different protocols, resists unknown attacks, and performs well in cross-network training and inference, with improved results when combined with F-Norm.

Soe et al. (2020) proposed a machine learning-based botnet attack detection framework with a sequential architecture, achieving about 99% detection performance using an artificial neural network (ANN), J48 decision tree, and Naïve Bayes classifiers. They used an efficient feature selection approach to create a high performing; lightweight system capable of detecting new attack types. Alissa et al. (2022) introduced decision trees, XGBoost, and DNN models for classifying botnet attacks in IoT environments with the UNSW-NB15 dataset. The decision tree model had the highest accuracy at 94%, followed by XGBoost at 88% and random forest

at 87.89%, with the DNN model showing precision rates above 90%. Al-Othman et al. (2020) developed a machine-learning approach for detecting IoT botnet attacks, where RF and J48 classifiers outperformed MLP networks, achieving accuracies of 0.96 for main classifications and 0.93 for subcategories, with a maximum micro-FN rate of 0.076. Their approach effectively distinguishes between normal and malicious traffic.

2.3.7. User and Entity Behaviour Analytics (UEBA)

UEBA leverages machine learning techniques to establish baselines for normal user and entity behaviour within a corporate network, identifying deviations that may indicate malicious activities or security breaches. Unsupervised learning techniques, such as clustering and anomaly detection, are well-suited for UEBA tasks, as they can learn normal behaviour patterns and identify deviations without relying on predefined rules or signatures (Husák et al., 2021). In their study, Shashanka et al. (2016) introduced an intelligence platform designed for User and Entity Behavior Analytics (UEBA) in cybersecurity. This platform uses machine learning algorithms, specifically those based on Singular Value Decomposition (SVD), to autonomously identify anomaly behaviours and notify analysts accordingly. Suspicious activities, which may indicate potential malicious intent, are highlighted alongside pertinent contextual details to facilitate deeper investigation by analysts. Salitin et al. (2016) investigated User Entity Behavior Analytics (UEBA) as a cybersecurity process for detecting real-time network attacks, including zero-day attacks. They evaluated 15 UEBA technologies, highlighting their strengths, weaknesses, and effectiveness in detecting real-time attacks. Muhammad et al. (2022) and R, Rengarajan & S Babu. (2021) proposed a UEBA-based framework to detect anomalies and insider threats by classifying user profiles as normal or aberrant. The framework uses IP addresses, location data, and user organization details for enhanced accuracy. The study emphasizes using data science and analytical methods to create data visualizations for effective anomaly identification.

The detection process in UEBA can be described using anomaly detection in multi-dimensional spaces. Let x_i represent the behaviour vector for entity *i* over a given period. The mean vector μ and the covariance matrix Σ define the baseline behaviour.

The Mahalanobis distance, which measures the number of standard deviations x_i is from the mean $\boldsymbol{\mu}$, is given by: $D_M(x_i) = \sqrt{(x_i - \mu)^T \sum^{-1} (x_i - \mu)}$

An alert is triggered if $D_M(x_i)$ a predefined threshold τ, calibrated to balance false positives and false negatives, enhancing the system's sensitivity and specificity (Ker, 2010).

CHAPTER THREE
METHODOLOGY

3. Introduction

Investigation of machine learning-based models for intrusion detection and threat intelligence analysis is paramount and contributes significantly to cybersecurity fortification. This research aims:

- To investigate the application of machine learning in detecting various cyber threats.
- To compare the performance of different machine learning models in detecting cyber threats.
- To investigate how machine learning algorithms can be applied to reduce the latency in responding to detected cyber threats.

3.1.Scope of the methodology

This section outlines the systematic framework of the study to achieve the objective of enhancing cybersecurity through the application of machine learning.

This includes data preprocessing techniques–scaling, feature extraction, and dimension reduction, machine learning models – RF, SVM, XGB, and DT, for intrusion detection, model training and evaluation using metrics such as accuracy, precision, recall and F1-score, and ethical considerations. These elements contribute to the goals of investigating, comparing, and optimizing machine learning models for cybersecurity applications.

The schematic block diagram of our proposed paradigm is illustrated in Figure 2.0.

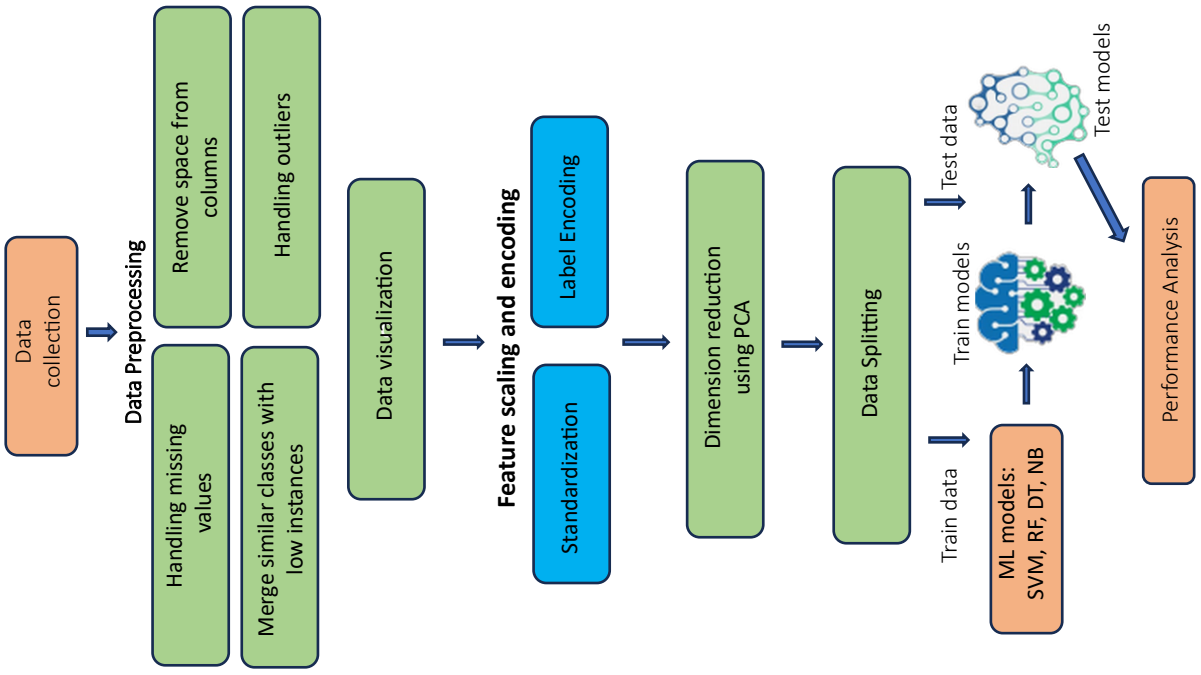


Figure 2.0. Systematic framework of Data Preprocessing with PCA for intrusion detection.

3.2.Research Approach

This research employs a data-driven approach to investigate the application of machine learning in cybersecurity enhancement, specifically focusing network intrusion detection.

However, due to the complexity of cybersecurity challenges, a mixed-method approach (quantitative and qualitative) would be used in this research (Mukhopadhyay & Jain, 2024). This would enable the examination of large datasets and the quantification of model performance using statistical metrics.
Qualitative insights from domain expertise such as CyberHawk, a cybersecurity provider, provide a nuanced understanding of the cybersecurity landscape. Incorporating these insights alongside quantitative performance metrics provides a holistic view of cybersecurity dynamics. This combined approach strengthens the validity and reliability of our research findings.

3.3.Framework

The conceptual framework of this study is based on the hypothesis that machine learn models, when trained on high-dimensional and large-scale datasets, can significantly enhance threat detection accuracy in cybersecurity systems. The study focuses on classification algorithms, aiming to detect and categorize cybersecurity threats.

3.4.Data Collection

The effectiveness of machine learning (ML) in cybersecurity heavily relies on the quality and relevance of the data used to train the model. However, one of the key challenges in this field is obtaining suitable and reliable datasets. Although the recent surge in interest in ML has led to the availability of more open datasets, these datasets still present significant limitations.
In this research, we utilized two benchmark datasets, namely UNSW-NB15 and CICIDS2017. These datasets are realistic to the IDS environments and have up-to-date attack categories to detect attacks.

3.4.1. UNSW-NB15 Dataset

This project utilizes the "UNSW-NB15" dataset, a pivotal network intrusion detection research resource. Created in 2015 by the IXIA PerfectStorm tool in the Cyber Range Lab of the University of New South Wales (UNSW), Canberra, this dataset blends real modern normal activities with synthetic contemporary attack behaviours. Its primary aim is to evaluate network intrusion detection systems (NIDS), making it essential for cybersecurity researchers and practitioners.
Available on the UNSW website and academic repositories like Kaggle, the dataset represents network traffic in a simulated environment that closely mimics real-world conditions. Permission was obtained following the university's research guidelines to access and utilize the dataset, which underwent rigorous verification to ensure its validity and reliability.
The UNSW-NB15 dataset comprises 257,673 rows and 49 columns in a CSV file. Each row represents a network traffic session, detailing attributes such as 'attack_cat', 'proto', 'service', and 'state'. It includes nine types of attacks: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms as indicated in Figure 1.0.

Attack Categories	Counts	Percentage (%)
Normal	93000	36.09%
Generic	58871	22.85%
Exploits	44525	17.28%
Fuzzers	24246	9.41%
Dos	16353	6.35%
Reconnaissance	13987	5.45%
Analysis	2677	1.03%
Backdoor	2329	0.90%
Shellcode	1511	0.11%
Worms	175	0.08%
Total	257,674	100

Table 1.0. The attack categories in UNSW-NB15 dataset.

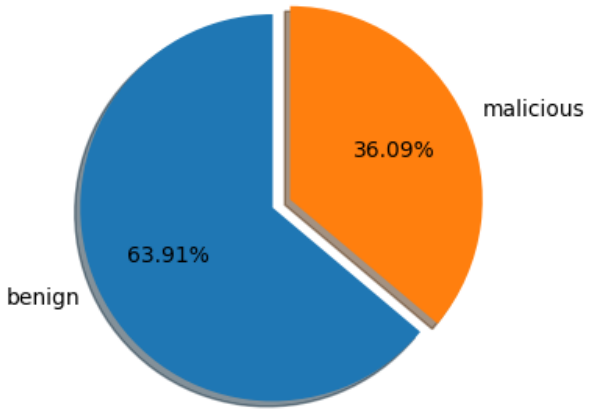


Figure 3.0. Pie-chart of attacks and non-attacks in UNWS-NB15 dataset.

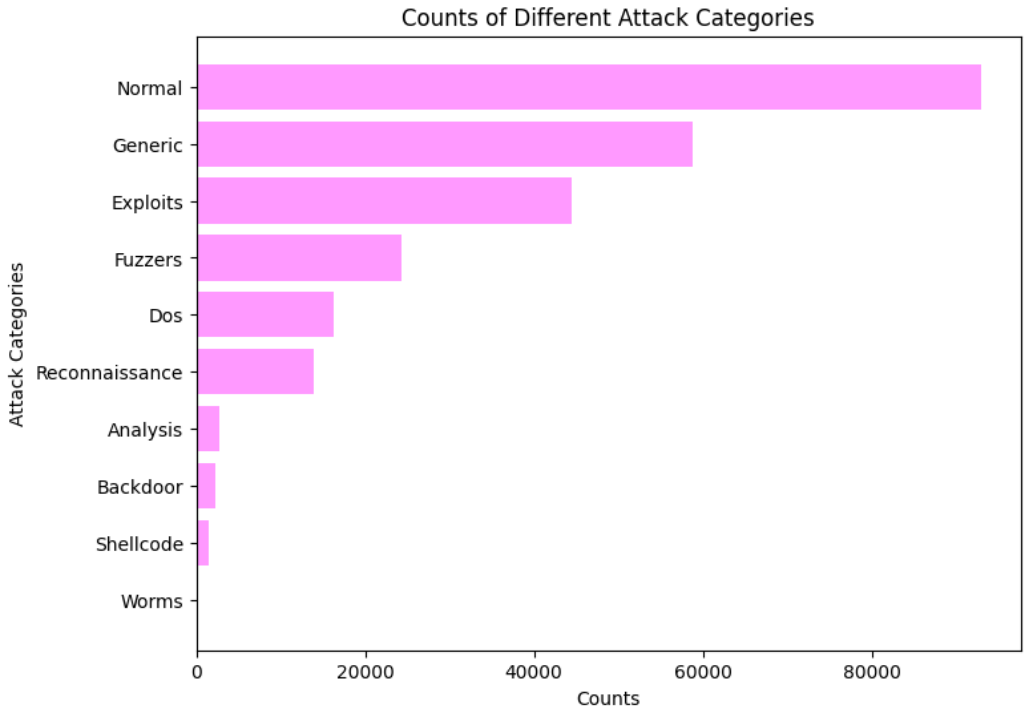


Figure 4.0. The horizontal bar plot of the attack categories of UNWS-NB15 dataset.

3.4.2. CICIDS2017 Dataset

Another benchmark dataset used in this study is the CICIDS2017 dataset, developed by the Canadian Institute for Cybersecurity (CIC) at the University of New Brunswick. This dataset is a significant resource for network intrusion detection research. Created using CICFlowMeter, it provides diverse network traffic data for evaluating intrusion detection systems (IDS). This dataset includes a comprehensive collection of benign and malicious network traffic, covering various cybersecurity threats such as Denial-of-Service (DoS), Distributed DoS (DDoS), brute-force attacks, Heartbleed, Botnet, and network infiltration.
Available through academic repositories and platforms like Kaggle, it was obtained following strict university guidelines and underwent rigorous validation to ensure data integrity and reliability. Structured as a CSV file, CICIDS2017 contains detailed attributes for each network traffic session. For this research, we utilized 10% of the samples from each class to manage time complexity. Our experimental dataset comprises 251,496 instances with 79 distinct features, including 14 attack classes, as detailed in Table 2.0.

Attack Categories	Counts	Percentage (%)
Benign	202974	80.78%
DDoS	17849	7.10%
PortScan	17495	6.96%
DoS Hulk	10335	4.11%
FTP-Patator	555	0.22%
DoS GoldenEye	448	0.18%
SSH-Patator	395	0.16%
Bot	331	0.13%
DoS slowloris	277	0.11%
Web Attack Brute Force	265	0.11%
DoS Slowhttptest	232	0.09%
Web Attack XSS	114	0.05%
Web Attack Sql Injection	4	0.002%
Infiltration	2	0.001%
Heartbleed	1	0.0004%
Total	251277	100

Table 2.0. The attack categories in CICIDS2017 dataset.

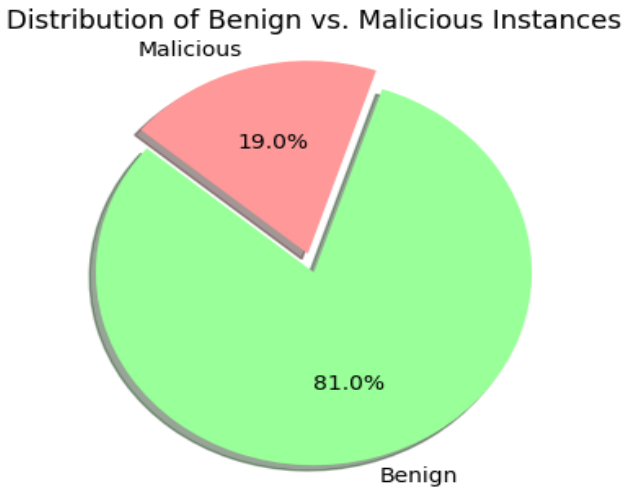


Figure 5.0. Pie-chart of attacks and non-attacks in CICIDS2017 dataset.

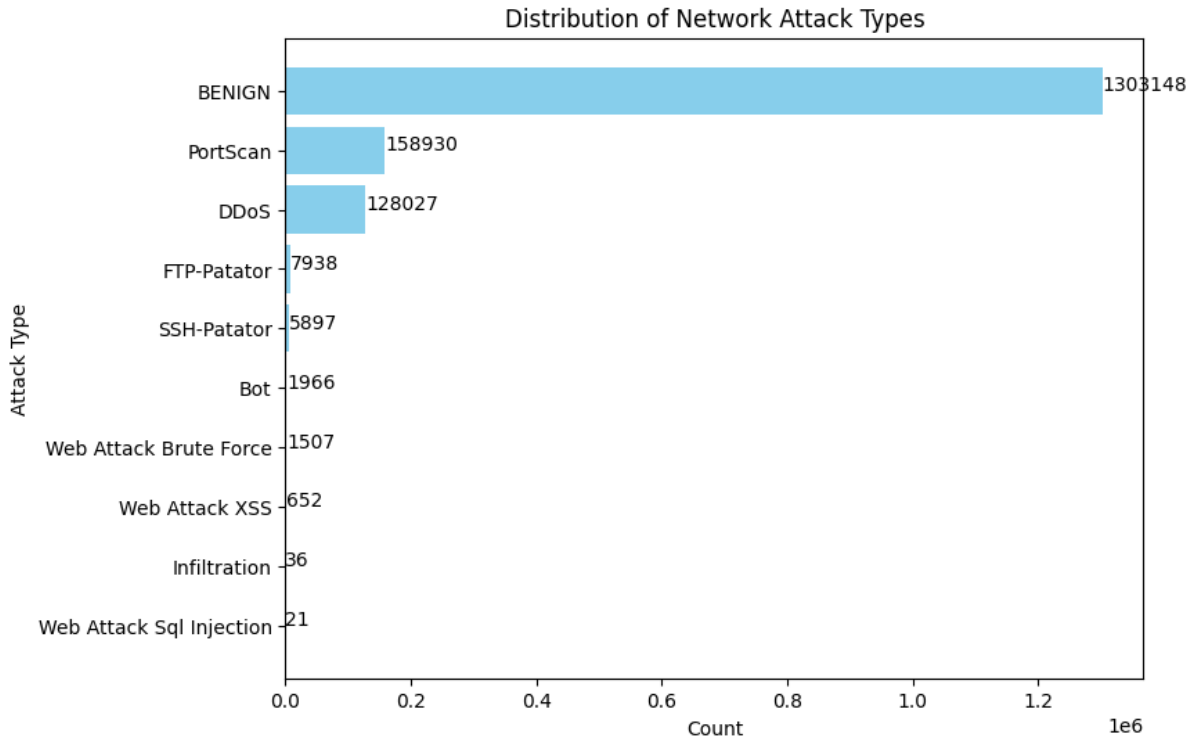


Figure 6.0. The horizontal bar plot of the attack categories of CICIDS2017 dataset.

3.5.Descriptive Statistics of The Datasets.

Descriptive statistics provide a summary of the dataset and helps to simplify high data volume to more understand form. It provides insight into the trends and patterns and distribution of the dataset, central tendencies and dispersions. It helps in identifying outliers, bias and other key characteristics of the dataset.

In our study, we used descriptive statistics in Table 3.0 and Table 4.0, to identify outliers. Values significantly outside this range—specifically those more than 1.5 times the IQR above the 75th percentile or below the 25th percentile—are typically considered outliers. Such outliers can overfit the models.

	Destination Port	Flow Duration	Total Fwd Packets	Total Backward Packets	Total Length of Fwd Packets	Total Length of Bwd Packets	Fwd Packet Length Max	Fwd Packet Length Min	Fwd Packet Length Mean	Fwd Packet Length Std	...	act_dat
count	251277.000000	2.512770e+05	251277.000000	251277.000000	2.512770e+05	2.512770e+05	251277.000000	251277.000000	251277.000000	251277.000000	...	251
mean	8133.449966	1.289382e+07	9.597556	10.851228	5.465844e+02	1.688547e+04	210.522973	19.695969	60.693135	70.321485	...	
std	18191.686543	3.162017e+07	758.726533	1030.102309	6.402296e+03	2.246447e+06	799.814732	67.276846	207.697886	319.977916	...	
min	0.000000	-1.000000e+00	1.000000	0.000000	0.000000e+00	0.000000e+00	0.000000	0.000000	0.000000	0.000000	...	
25%	53.000000	1.510000e+02	1.000000	1.000000	6.000000e+00	6.000000e+00	6.000000	0.000000	6.000000	0.000000	...	
50%	80.000000	3.116000e+04	2.000000	2.000000	5.700000e+01	1.170000e+02	35.000000	6.000000	32.000000	0.000000	...	
75%	443.000000	1.833664e+06	4.000000	4.000000	1.320000e+02	3.980000e+02	56.000000	36.000000	48.000000	16.263456	...	
max	65534.000000	1.200000e+08	206687.000000	281741.000000	1.645814e+06	6.070000e+08	23360.000000	1983.000000	5238.769231	5796.500690	...	198

8 rows × 78 columns

Figure 3.0. Descriptive statistics of features of CICIDS2017 dataset.

```
merged_data.describe().round(2)
```

	dur	rate	sttl	dttl	sload	sloss	dloss	sjit	djit	dtcpb	smean	trans_depth	ct_state_ttl	ct_f
count	257673.00	257673.00	257673.00	257673.00	2.576730e+05	257673.00	257673.00	257673.00	257673.00	2.576730e+05	257673.00	257673.00	257673.00	
mean	1.25	91253.91	180.00	84.75	7.060869e+07	4.89	6.74	5419.37	582.25	1.002295e+09	137.64	0.10	1.32	
std	5.97	160344.64	102.49	112.76	1.857313e+08	65.57	53.70	49034.50	3930.15	1.363877e+09	205.90	0.71	0.99	
min	0.00	0.00	0.00	0.00	0.000000e+00	0.00	0.00	0.00	0.00	0.000000e+00	24.00	0.00	0.00	
25%	0.00	30.79	62.00	0.00	1.231800e+04	0.00	0.00	0.00	0.00	0.000000e+00	57.00	0.00	1.00	
50%	0.00	2955.66	254.00	29.00	7.439423e+05	0.00	0.00	0.67	0.00	0.000000e+00	73.00	0.00	1.00	
75%	0.69	125000.00	254.00	252.00	8.000000e+07	3.00	2.00	2787.37	119.71	1.992752e+09	100.00	0.00	2.00	
max	60.00	1000000.00	255.00	254.00	5.988000e+09	5319.00	5507.00	1483830.92	463199.24	4.294882e+09	1504.00	172.00	6.00	

Figure 4.0. Descriptive statistics of features of CICIDS2017 dataset.

3.6. Data Preprocessing

Data quality is commendable for ML models' training and evaluation. To ensure the quality of a dataset, it undergoes pre-processing by handling missing values, outliers, data scaling, and feature engineering. These approaches transform the data into a format that maximizes the performance and interpretability of machine learning models.

The comprehensive preprocessing of UNSW-NB15 and CICIDS2017 datasets ensures that the models are well-equipped to handle the complexities of network traffic classification and anomaly detection in cybersecurity applications.

3.6.1. Feature Engineering

Feature engineering involves converting raw data into features that accurately represent the underlying problem for a predictive model. In this study, the UNSW-NB15 and CICIDS2017 datasets underwent rigorous feature engineering to enhance their suitability for machine learning models. These datasets include various types of network traffic data, categorical data, missing values, outliers, and imbalanced datasets. Refining these aspects is crucial for improving the performance and interpretability of any machine learning model.

3.6.2. Handling of Missing Values

With only 2.0% and 3.0% of the data missing in the UNSW-NB15 and CICIDS2017 datasets, respectively, we chose to apply listwise deletion. This method removed 5,153 rows from the UNSW-NB15 dataset and 5,025 rows from the CICIDS2017 dataset, where data was absent. By doing so, we minimized the impact on the overall datasets while ensuring the completeness of the remaining variables in our analysis.

3.6.3. Mitigating Outliers

In statistics, outliers are data points that lie outside the overall distribution of a dataset. Outliers can degrade the performance of models by introducing noise, leading to poor accuracy and generalization. They can cause models to overfit, as the model may learn the noise rather than the underlying patterns. Several models, including SVMs and Decision Trees, are prone to outliers and can significantly impact their performances.

In this study, the outliers of the datasets were found using the interquartile range (IQR) approach, a robust statistical technique well-suited for non-normally distributed features.
The mathematical approach to the interquartile range is given by: $IQR = Q_3 - Q_1$

where:

Q_1 is the first quartile (0.25 for a sorted dataset)

Q_3 is the third quartile (0.75 for a sorted dataset)

To mitigate the outliers, the logarithm transformation was applied to the entire dataset to ensure consistency in data transformation and avoid introducing any bias or inconsistencies, thereby achieving a more normal distribution of data. However, to cater for zero or negative values, the dataset was shifted before applying the logarithm transformation, as the logarithm is undefined for non-positive numbers.

3.6.4. Feature Encoding

Machine learning models thrive on numerical data. Feature encoding plays a role in converting/transforming categorical data into numerical format easily understood by machine learning algorithms. In the CICIDS2017 dataset, '*label*' variable is encoded into numerical format using Label Encoding. Each unique value in a categorical feature is assigned a unique integer as indicated in Figure 5.0.

Label	Encoded Label
BENIGN	0
DDoS	1
PortScan	2
DoS Hulk	3
FTP-Patator	4
DoS GoldenEye	5
SSH-Patator	6
Bot	7
DoS slowloris	8
Web Attack Brute Force	9
DoS Slowhttptest	10
Web Attack XSS	11
Web Attack Sql Injection	12
Infiltration	13
Heartbleed	14

Table 5.0. Label encoding of categorical data in CICIDS2017 Dataset.

3.6.5. Correlation Matrix

Using a correlation heatmap for feature selection effectively identified and mitigated multicollinearity, ensuring our model's robustness and interpretability. High correlations between predictors lead to multicollinearity, inflating parameter estimate variances and making the model unstable.
The Pearson correlation coefficient measures the linear relationship between two variables, ranging from -1 (strong negative correlation) to 1 (strong positive correlation).

In the heatmaps illustrated in Figure 7.0 below, we identified feature pairs with correlation coefficients above a threshold of 0.8 in the UNSW-NB15 dataset. For each pair, we retained the more important feature and removed the other. This visualization in allowed us to make informed decisions, enhancing our model's reliability and simplifying it by reducing the number of predictors.

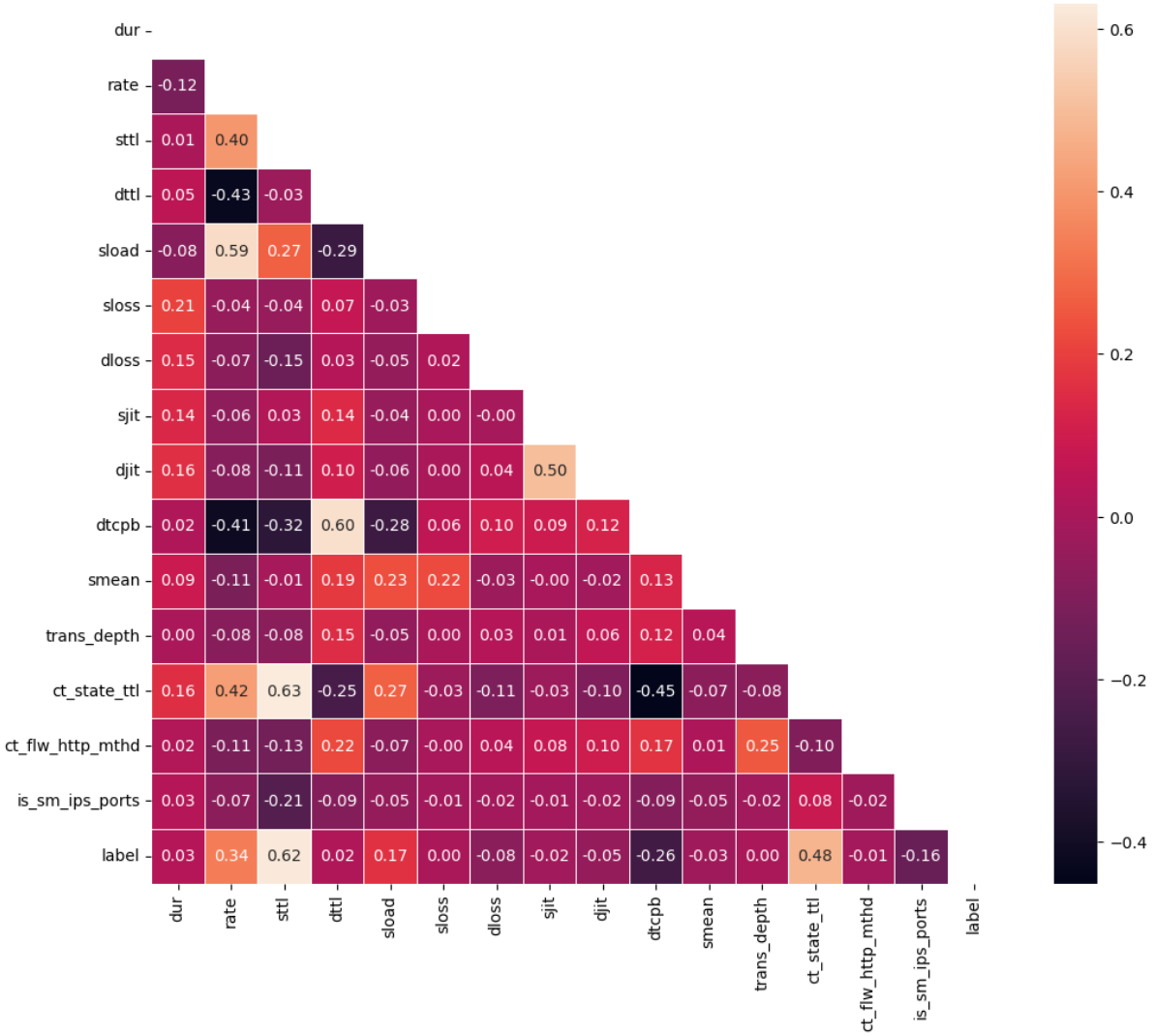


Figure 7.0. Correlation heatmap of the features of UNSW-NB15 dataset.

3.6.6. Feature Scaling

The dominance of one feature in the dataset impacts machine learning models and their performance. One effective way to handle this is by scaling the dataset. In this report, all features except the label were scaled using the '*StandardScaler*' from the *'scikit-learn'* library. This transformation ensures that the features have zero mean and unit standard deviation, preventing any feature from dominating others due to scale differences.

Mathematically, the standard scaler is expressed as: $z = \frac{x - \mu}{\sigma}$,

where: x = feature, μ = mean of feature x, σ = standard deviation

3.6.7. Variation Inflation Factor (VIF)

To identify the degree of multicollinearity among features in the CICIDS2017 dataset, we utilized the Variance Inflation Factor (VIF) to reduce the variance. High variance inflation can result in unstable and unreliable estimates, which can adversely affect the model's performance and interpretability.

For predictors in our dataset, we calculated the VIF using the formula: $VIF_i = \frac{1}{1-R_i^2}$,

Where:

R_i^2 is the unadjusted coefficient of determination for regressing ith variable on the remaining variables.

Using the following thresholds:

$VIF < 10$: Indicates moderate multicollinearity, which is generally acceptable.

$VIF \geq 10$: Suggests high multicollinearity, which could be problematic.

Predictors with VIF values greater than or equal to 10 were flagged and iteratively removed from the model. After each removal, VIF values were recalculated for the remaining predictors. This process was repeated until all remaining predictors had VIF values below the chosen threshold.

3.6.8. Feature extraction using PCA

A high-dimensional dataset can increase the complexity of a machine learning model, often resulting in overfitting and poor performance. To address this in our study, we applied Principal Component Analysis (PCA) to reduce the dimensionality of the dataset. This approach helped decrease computation time and facilitated exploratory data analysis by enabling the visualization of high-dimensional data. Additionally, PCA was instrumental in eliminating redundant features while generating new features that retained as much variance as possible from the original dataset.

This transformation was achieved by calculating eigenvectors and eigenvalues from the covariance matrix of the original data, allowing us to capture the most significant information in fewer dimensions.

To reduce the number of features from n samples to k features, involves the following:

The covariance Matrix CM is expressed as: $C_M = \frac{1}{n}\sum_{i=1}^{n}(x_i)(x_i)^T$

where:

n = instances number

$x_i = n \times k$ data matrix after mean subtraction (or standardization)

The eigenvector and eigenvalue associated with the covariance matrix are given by: $C_v = \lambda v$

where:

λ is the eigenvalue

v is the corresponding eigenvector

The principal component selection of the top k eigenvectors corresponding to the largest eigenvalues to form the principal components is given by: $PCA = xV_k$

where: V_k is the matrix for the top k eigenvectors

The computational complexity of PCA is influenced by the number of features D and the number of data points k is determined by: $O(D_k^3)$.

In this report, the Principal Component Analysis was achieved using '*PCA*' module from the *'scikit-learn'* library for dimensionality reduction. This step helps in simplifying the model and reducing computational load. Figure 9.0 illustrates feature extraction and dimensionality reduction through PCA.

Features/ Rows	F_1	F_2	F_3	....	F_n
R_1					
R_2					
R_3					
:	…	…	…	…	…
R_n					

Features/ Rows	P_1	P_2	P_3	....	P_{10}
R_1					
R_2					
R_3					
:	…	…	…	…	…
R_n					

Figure 9.0. Process of PCA dimensionality reduction

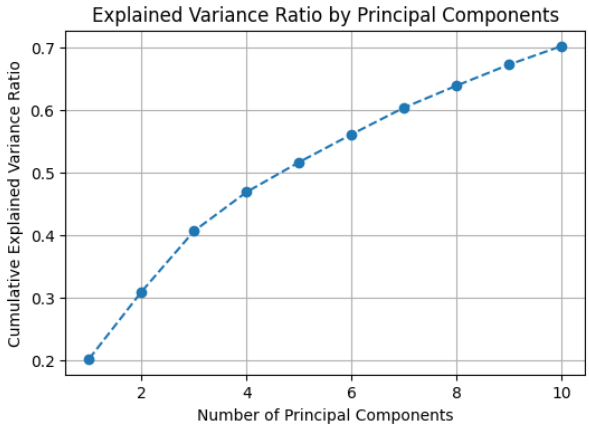


Figure 10.0. PCA variance ratio of CICIDS2017 dataset

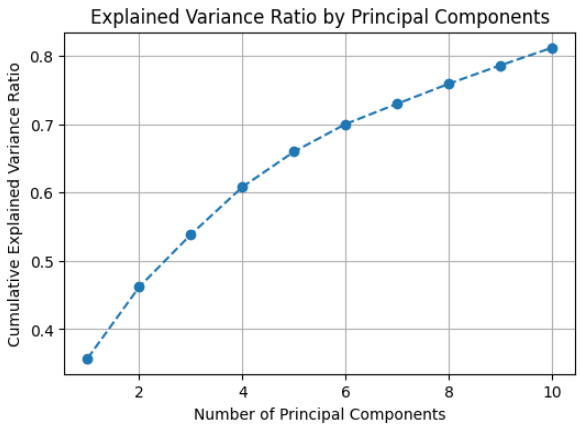


Figure 11.0. PCA variance ratio of UNSW-NB15 dataset

3.6.9. Splitting The Data

For this report, we use the standard 70 – 30 splits. This commonly used practice of data partition method allocates 70% of the data for training the model and the remaining 30% reserved for testing its performance.

This method ensures that the model has adequate data to learn from and a separate, unseen dataset to evaluate its effectiveness, thus providing a realistic measure of how the model is likely to perform in real-world scenarios. To ensure data consistency, we employed stratified sampling to maintain the original dataset's class distribution in both training and testing sets. In cybersecurity, this technique prevents bias towards the majority class and ensures accurate detection and classification of benign and malicious activities.

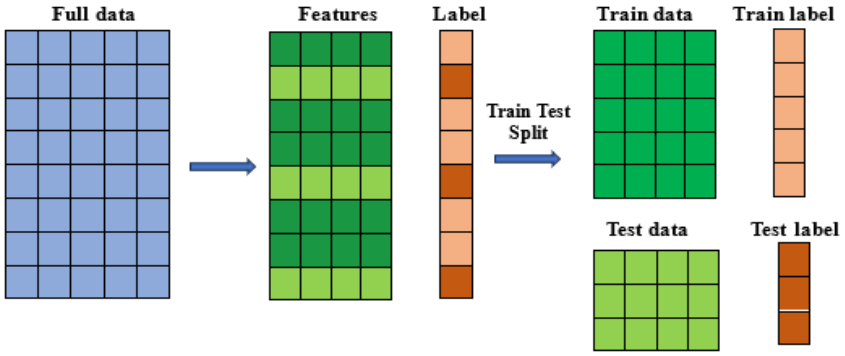


Figure 12.0. The process of data splitting.

3.7.Machine Learning Models

This section outlines the various machine-learning models for classifying the UNSW-NB15 dataset. These models include Support Vector Machine (SVM), Random Forest (RF), Extreme Gradient Boosting, and Decision Tree. Each model is evaluated using various metrics to assess their effectiveness in distinguishing between benign and attack traffic.

3.7.1. Support Vector Machine (SVM)

This study employs Support Vector Machines (SVMs) for their effectiveness in identifying complex cyber threat patterns, focusing on their application to enhance detection accuracy in cybersecurity. We employed an SVM with a radial basis function (RBF) kernel, effective in handling high-dimensional and non-linear data.

The SVM was trained using a grid search to optimize hyperparameters such as the regularization parameter C and the kernel coefficient γ. This process minimized the hinge loss function, improving the model's accuracy and performance.

As shown in Figure 13.0, SVMs discover the optimal hyperplane that maximizes the margin between classes, improving predictive accuracy and minimizing the risk of overfitting. Moreover, SVMs are easy to implement, require a small training dataset, and are suitable for extensive dataset analysis. This is crucial in threat detection, where identifying new, previously unseen threats is essential.

By focusing on support vectors, SVMs are less likely to overfit the training data, especially in high-dimensional spaces, making them reliable for threat detection.

The mathematical model of SVMs involves solving an optimization problem to find the optimal hyperplane for class separation, expressed as:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

Subject to: $y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$

Where $\boldsymbol{w}$ is the weight vector, $\boldsymbol{b}$ is the bias, $\boldsymbol{\xi_i}$ is slack variables, $\boldsymbol{C}$ is a regularization parameter, $\boldsymbol{x_i}$ are the feature vectors, and $\boldsymbol{y_i}$ are the labels.

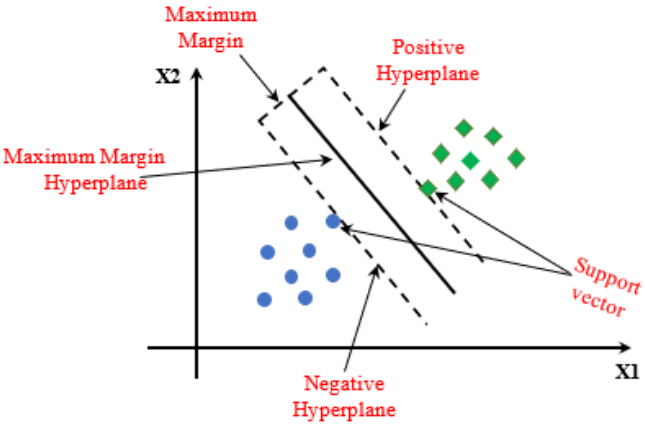


Figure 13.0. Support Vector Machine (SVM) Classifier Visualization.

3.7.2. Random Forest (RF)

The CICIDS2017 and UNSW-NB15 datasets are high-dimensional, characterized by class imbalance and complexity, which present significant challenges. These factors often lead to issues like overfitting and increased computational demands when using certain algorithms. In our study, we employ Random Forest (RF) because it is particularly well-suited for handling such complex data efficiently, offering both robustness and interpretability.

Moreover, RF's scalability allows it to be parallelized across multiple processors or machines, making it an ideal choice for large datasets like CICIDS2017 and UNSW-NB15. Additionally, RF provides valuable insights into feature importance, helping researchers identify the most critical features for detecting intrusions. The model's performance can be further optimized by tuning hyperparameters such as the number of trees, maximum depth, and minimum samples per split, using techniques like grid search or cross-validation. These combined qualities make Random Forest an excellent choice for developing effective Intrusion Detection Systems (IDS) with these challenging datasets.

3.7.3. Extreme Gradient Boosting (XGBoost)

Xgboost model builds a sequence derived from the function gradients to improve model performance. This objective function combines a loss function L to measure how well the model predicts the target and a regularization term Ω to penalize the complexity of the model (regularization).

$$Obj(t) = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{t=1}^{T} \Omega(f_t)$$

Where:

$\sum_{i=1}^{n} L(y_i, \hat{y}_i)$ is the loss function of the actual value y_i and predictive value $\hat{y}_i$.

$\sum_{t=1}^{T} \Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{t=1}^{T} w_j^{\,2}$ is the **regularization term**, which penalizes the complexity of the model to prevent overfitting.

In this research, we implement XGB for its ability to handle imbalanced datasets. Network intrusion datasets often exhibit a significant imbalance between benign and malicious activities. XGB's built-in mechanisms, such as scale_pos_weight, effectively manage this imbalance, making it adept at detecting rare threats. Additionally, XGB's high scalability allows it to process large datasets efficiently in real-world cybersecurity scenarios. The high predictive performance of XGB makes it outperform other models in accuracy, precision, and recall, classification metrics in threat detection. Furthermore, XGB includes regularization penalties to avoid overfitting, ensuring the model can generalize adequately.

3.7.4. Decision Tree (DT)

In this study, the Decision Tree (DT) model was trained on the UNSW-NB15 and CICIDS2017 training datasets and evaluated on their respective test datasets. We chose the DT model due to its ability to handle large datasets, such as our datasets, effectively and efficiently. DT can process extensive high-dimensional data with moderate computation complexity, making them well-suited for real-time intrusion detection where speed is crucial. Additionally, DTs are adept at modelling non-linear relationships between features, as cyberattacks involve complex, non-linear interactions among network parameters. To optimize performance, the hyperparameters were tuned using the grid search techniques. The core mathematical idea in decision trees involves selecting the optimal feature to split on, often measured by Gini impurity or information gain.

Gini impurity measures the probability of a randomly chosen element being incorrectly classified if it was randomly labelled according to the distribution of labels in the subset. It is defined as:

$$Gini(D) = 1 - \sum_{i=1}^{C} p_i^{\,2}$$

where: D is the dataset at the node, C is the number of classes and p is the proportion of instances in class i.

Information gain measures the reduction in entropy after splitting the dataset a particular feature. Entropy, a measure of the uncertainty in a dataset is defined as:

$$Entropy(D) = \sum_{i=1}^{C} -p_i \log_2(p_i)$$

where p_i is the probability of selecting an element of class i

Information gain is calculated as:

$$Information\ Gain(D, A) = Entropy(D) - \sum_{v \in Values(A)} \frac{|D_v|}{|D|} Entropy(D_v)$$

where:

A is the feature to split on.

$Values(A)$ are the possible values of feature A.

Dv is the subset of D for which feature A has value v.

3.8. Model Training and Evaluation

After preprocessing the data to remove noise and inconsistencies, the dataset was split into 70% for training and 30% for testing. The training set allowed the models to learn the patterns, relationships, and structures in the data.

Each machine learning model—SVM, RF, XGB, and DT— was trained on the training dataset to ensure they learned the underlying patterns and the correlations in the data. The testing set was used to evaluate the performance of the models.

This ensures unbiased assessment and exclusively reflects the models' ability to generalize to unseen data. Separating the training and testing phases is critical for validating the models' effectiveness and their potential real-world applicability.

3.8.1. Hyperparameter Tuning

Optimizing model parameters enhances the performance of ML models. This process involves selecting the best set of hyperparameters that maximize the model's performance on the validation set. Techniques like Grid Search and Random Search, often combined with cross-validation, are used for this purpose.

In Table 4.0, we explore these techniques and how they apply to RF, SVM, XGB, and DT.

Models	Hyperparameters
SVM	c = 1.0, kernel = 'linear'
RF	max_features = 'auto'max_depth = 10, min_samples _split = 5, n_estimators = 100, random state = 42
XGB	n_estimators = 100, learning_rate = 0.1, max_depth = 5, gamma = 0.3, sub_sample = 0.5
DT	Criterion = 'gini', splitter = 'best', max_depth = 10, min_samples_split = 5, max_features = 'auto'

Table 6.0. Hyperparameters of the various ML models

3.9.Experimental Setup

After describing the UNSW-NB15 and CICIDS2017 benchmark datasets, the experiments were conducted on a high-performance HP 2X-large virtual computer with the following specifications:

- Storage: SanDisk DA4064 with a total space of 58.4 GB and a JMicron Tech SCSI Disk Device with total spaces of 263.5 GB and 212.9 GB.
- Memory: 32 GB RAM to facilitate efficient data manipulation and complex model training.
- Processor: Intel(R) Celeron(R) N4120 CPU @ 1.10GHz with 4 cores, enabling rapid computational operations and in-depth analysis.

The experiments were conducted using Jupyter Notebook, accessed through Anaconda Navigator, which provides an interactive and user-friendly environment for coding, data analysis, and visualization. To evaluate the performance of the models, we utilized pandas for data manipulation, Seaborn and Matplotlib for data visualization, and Scikit-learn for machine learning algorithms, leveraging the Python programming language.

3.10. Evaluation Metrics

To assess the performance of our optimized model, we used a range of standard evaluation metrics for classification models. These metrics included accuracy, precision, F1-score, recall, confusion matrix, and ROC-AUC curve. These metrics comprehensively assess the models' classification and prediction capabilities, covering overall correctness, precision in positive predictions, balance between precision and recall, ability to detect positive instances, detailed class-wise performance, and discrimination threshold effectiveness.

3.10.1. Accuracy

Accuracy is a fundamental metric for assessing a machine learning model's effectiveness, representing the proportion of correctly classified instances out of the total evaluated instances. It reflects how often the model's predictions align with actual outcomes. To assess the overall correctness of the model's predictions.

Mathematically, accuracy is expressed as:

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

where: TP: True Positives, TN: True Negatives,

FP: False Positives, FN: False Negatives

A high accuracy rate means the model's predictions align well with actual outcomes, suggesting effective threat detection capabilities.

3.10.2. Precision

Precision is an evaluation metric for a machine learning model that expresses the ratio of the true positive value predicted to the total number of positive values predicted by the model. It evaluates the accuracy of positive predictions, minimizing false alarms.

It is calculated using the formula:

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

High precision indicates that the model accurately identifies threats without raising too many false alarms.

3.10.3. Recall

Recall, or sensitivity, measures the proportion of actual positive cases (threats) correctly identified by the model. It measures the model's ability to correctly identify actual threats.

The formula for the recall is:

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

High recall indicates that the model effectively captures most threats, minimizing the chances of undetected security breaches.

3.10.4. F1-Score

The F1 score calculates the harmonic mean of the precision and recall scores for classification problems. It balances precision and recall, ensuring a comprehensive evaluation.

The F1-score ranges from 0 to 1, with 1 being the best possible score, indicating perfect precision and recall. The F1 score is expressed as:

$$F1-score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}}$$

A high F1-score means the model performs well in identifying threats while maintaining a low false alarm rate.

3.10.5. Confusion Matrix

This is a tabular visualization or representation that summarizes the performance of a classification model. It contains four combinations of predicted and actual values: True Positive (*TP*), True Negative (*TN*), False Positive (*FP*), and False Negative (*FN*). It

	Actual Positive	*Actual Negative*
Predicted Positive	*TP*	*FP*
Predicted Negative	*FN*	*TN*

This matrix provides a detail breakdown of the model's performance in each class.

3.10.6. ROC-AUC Curve

The ROC curve is a two-dimensional plot used to evaluate the performance of binary classifiers, plotting the true positive rate (sensitivity) against the false positive rate (1 - specificity). It visualizes the trade-off between sensitivity and the false positive rate across various threshold settings.

$$AUC = \int_0^1 ROC(t)dt$$

The Area Under the ROC Curve (AUC) provides an aggregate performance measure across all classification thresholds. A high AUC value of 1 suggests that the model excels at distinguishing between class labels, while an AUC value of 0 indicates poor predictive performance, implying results like random guessing. This curve indicates the effectiveness of the classification model.

3.11. Ethical issues

Several ethical considerations were identified and addressed in this research on leveraging machine learning for advanced threat detection and response in cybersecurity. The benchmark datasets used, UNSW-NB15 and CICIDS2017, contain network traffic data that could potentially lead to privacy breaches if mishandled. To mitigate this risk, the datasets were anonymized to remove any personally identifiable information before analysis.

Additionally, bias in machine learning is a concern, especially with imbalanced datasets. To address this, we employed confusion matrix analysis and class-wise performance metrics to identify and correct disparities in the model's performance across different classes. We also applied techniques like SMOTE (Synthetic Minority Over-sampling Technique) to handle class imbalance and ensure fairness without introducing additional bias.

CHAPTER FOUR
RESEARCH ANALYSIS AND RESULT

4. Introduction

Having gone through extensive data preprocessing, including noise removal, normalization, and feature engineering, we trained the models on the datasets to evaluate the efficacy in enhancing cybersecurity threat detection and response. Specifically, we compared the performance of four machine learning models—Support Vector Machine (SVM), Decision Trees (DT), Random Forests (RF), and Extreme Gradient Boosting (XGB)—within the context of Intrusion Detection Systems (IDS). Our analysis utilized two benchmark datasets: UNSW-NB15 and CICIDS2017. The evaluation focused on key performance metrics such as Recall, Accuracy, Precision, and F1-Score. Additionally, we assessed the models' performance using the confusion matrix and ROC-AUC curve.

4.1.Analysis Based on UNSW-NB15 Dataset

This analysis illustrates the performance of the models on the UNSW-NB15 dataset. The figures and tables provide a comprehensive view of our models' performance and highlight the impact of feature selection and preprocessing on IDS using the UNSW-NB15 dataset.

Table 7.0. offers insights into the performance metrics of the evaluated ML models. RF and XGB demonstrate superior performance in accuracy, each achieving 93.00%, indicating their high reliability in correctly classifying instances. SVM follows with an accuracy of 90.00%, while DT shows the lowest accuracy at 89.00%, though it still maintains a respectable performance.
SVM and DT exhibit high precision, each at 99%, indicating their effectiveness in correctly identifying positive cases with minimal errors. XGB also excels in precision with an impressive 94%, highlighting its ability to identify positive cases correctly. With 91% precision, RF shows a strong ability to make accurate positive predictions, though it is slightly more prone to false positives, unlike DT and XGB.

XGB leads in recall with 95%, showcasing its ability to identify nearly all positive instances. Meanwhile, RF and SVM follow with 90% and 72%, respectively, demonstrating their effectiveness in capturing positive cases. DT, with a recall of 70%, performs reasonably well but may miss some positive instances. Furthermore, XGB also achieves the highest F1-score at 94%, reflecting its excellent balance between precision and recall. RF follows with an F1-score of 90%, indicating strong overall performance. SVM and DT, with F1-scores of 83% and 82%, respectively, show balanced performance but fall short compared to RF and XGB.

In Figure 15.0, the features 'sttl', 'smean', and 'sloss' significantly impact the models' performance, with sttl alone contributing over 70%. This highlights the critical role of these features, though other features such as 'ct_state_ttl', 'dloss', etc. also make meaningful contributions to the overall model effectiveness.

Analyzing the confusion matrices in Figure 16.0 reveals the following performance metrics for the models: The TP rates are 94.59% for RF, 99.75% for SVM, 93.72% for DT, and 94.56%

for XGB. SVM leads with the highest TP rate, making it exceptionally strong in recognizing true positives, followed closely by RF and XGB, with DT slightly behind.

The TN rates are 89.91% for RF, 71.73% for SVM, 88.52% for DT, and 89.64% for XGB. RF and XGB have the highest TN rates, demonstrating their robustness in detecting non-benign traffic. SVM, despite its high TP rate, has a significantly lower TN rate, indicating it struggles more with false positives. Additionally, the FPR are 10.09% for RF, 28.27% for SVM, 10.36% for DT, and 3.76% for XGB. SVM has the highest FP rate, meaning it incorrectly labels more benign traffic as positive.

In contrast, RF and XGB have the lowest FP rates, making them more reliable in reducing false positives. Furthermore, the FN rates are 5.41% for RF, 0.25% for SVM, 6.28% for DT, and 5.44% for XGB. With its lowest FN rate, SVM rarely misses positive cases, aligning with its high TPR. Conversely, DT has the highest FN rate, indicating a higher tendency to miss positive cases, followed closely by XGB and RF.

Figure 17.0 displays ROC curves illustrating the performance of machine learning models on the UNSW-NB15 dataset. RF and XGB lead with AUC values of 0.92, closely followed by DT at 0.91. SVM trails with 0.86, still indicating strong performance. These high AUC scores across all models (0.86-0.92) demonstrate their robust predictive capabilities on this dataset, with tree-based and ensemble methods showing effectiveness. The results highlight the models' ability to differentiate between classes in the UNSW-NB15 dataset, providing valuable insights for cybersecurity classification tasks.

ML	*Accuracy*	*Precision*	*Recall*	*F1-Score*
SVM	90.00%	99.00%	72.00%	83.00%
RF	93.00%	91.00%	90.00%	90.00%
DT	89.00%	99.00%	70.00%	82.00%
XGB	93.00%	94.00%	95.00%	94.00%

Table 7.0. Performance analysis of ML models on UNSW-NB15 dataset

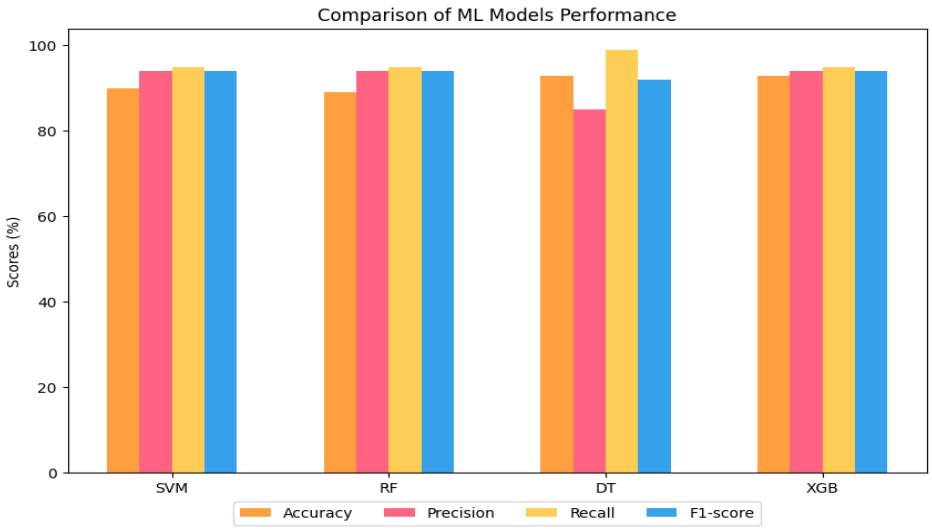


Figure 14.0. Bar chat of model performance on UNSW-NB15 dataset

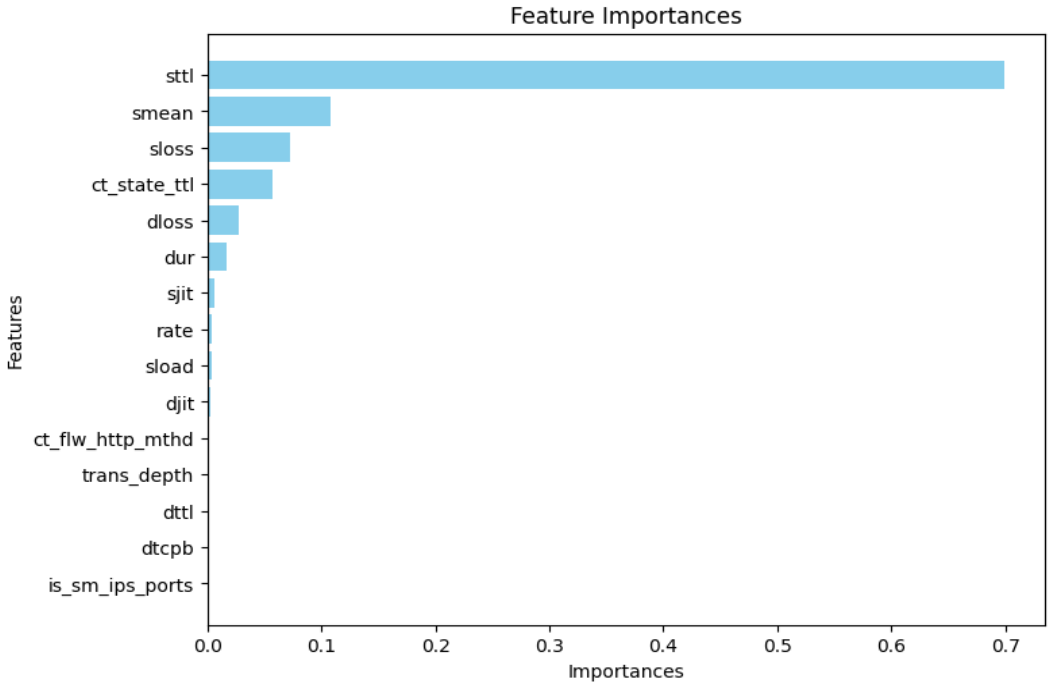


Figure 15.0. Feature importance of UNSW-NB15 dataset.

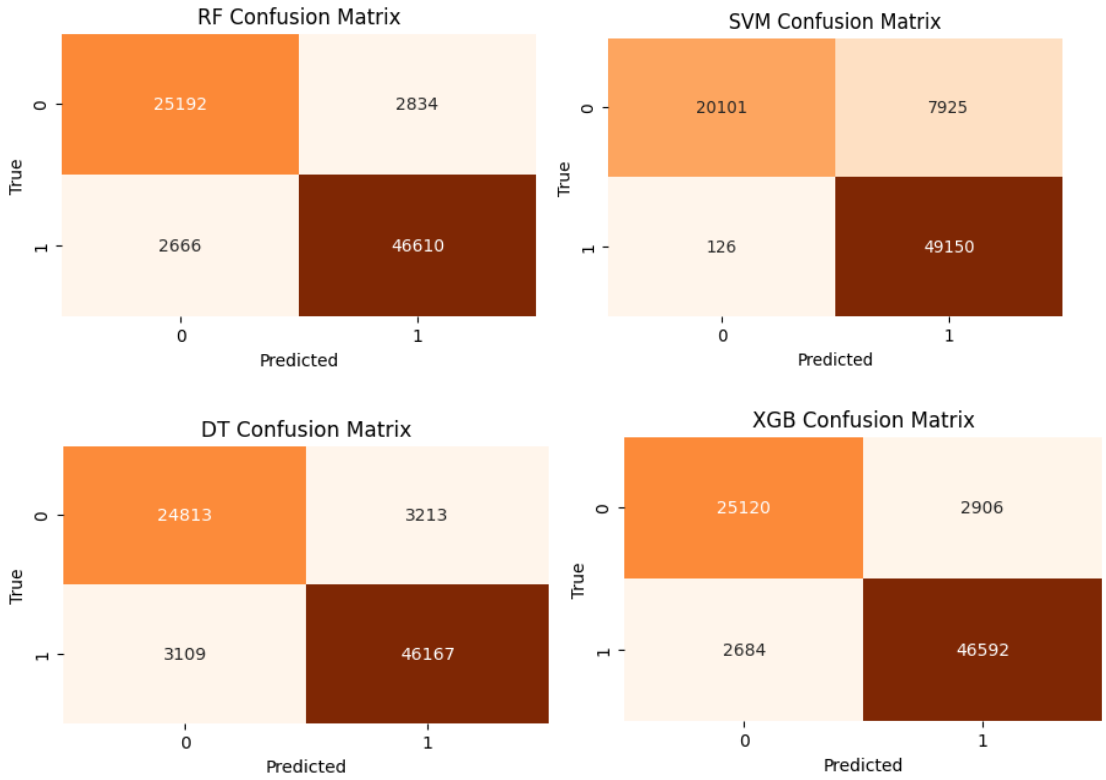


Figure 16.0. Confusion matrix of each ML model for UNSW-NB15 dataset

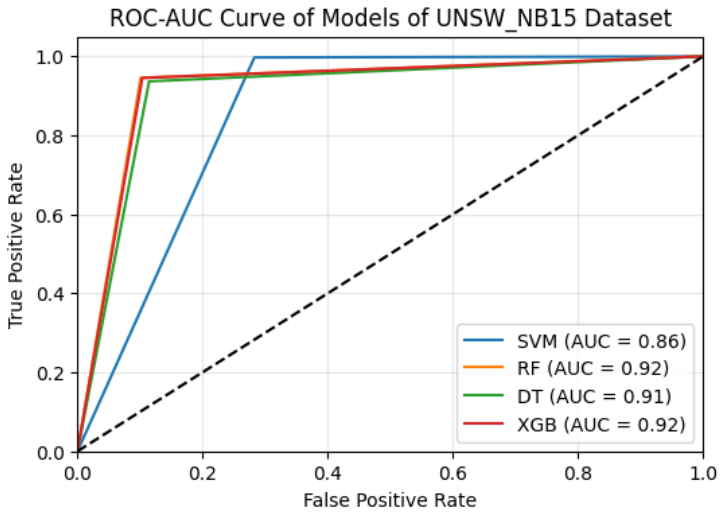


Figure 17.0. ROC-AUC curve for UNSW-NB15 dataset.

4.2.Analysis Based on CICIDS2017 Dataset

In this section, we present the results and analysis of various ML models applied to threat detection in cybersecurity using the CICIDS2017 dataset. The evaluated models include SVM, RF, DT, and XGB. We use multiple metrics, including accuracy, precision, recall, and F1-score, to evaluate the performance of these models.

Table 8.0. illustrates the performance analysis of each model. These figures provide a comprehensive view of our models' performance and the impact of feature selection and preprocessing on IDS using the CICIDS2017 dataset.

ML	*Accuracy*	*Precision*	*Recall*	*F1-Score*
SVM	90.48%	95.86%	94.05%	94.95%
RF	98.44%	99.22%	98.91%	99.07%
DT	98.38%	98.99%	99.08%	99.03%
XGB	98.52%	99.92%	98.97%	99.13%

Table 8.0. Performance analysis of ML models on CICIDS2017 dataset.

Performance analysis of ML models on UNSW-NB15 dataset				
ML	*Accuracy*	*Precision*	*Recall*	*F1-Score*
SVM	90.00%	99.00%	72.00%	83.00%
RF	93.00%	91.00%	90.00%	90.00%
DT	89.00%	99.00%	70.00%	82.00%
XGB	93.00%	94.00%	95.00%	94.00%
Performance analysis of ML models on CICIDS2017 dataset				
ML	*Accuracy*	*Precision*	*Recall*	*F1-Score*
SVM	90.48%	95.86%	94.05%	94.95%
RF	98.44%	99.22%	98.91%	99.07%
DT	98.38%	98.99%	99.08%	99.03%
XGB	98.52%	99.92%	98.97%	99.13%

Table 9.0. Performance analysis of ML models on UNSW-NB15 and CICIDS2017 datasets.

RF and XGB stand out with the highest accuracy rates of 98.44% and 98.52%, respectively. These figures demonstrate their high reliability and minimal error rates. However, DT shows strong performance with a 98.38% accuracy, closely lagging RF and XGB. Although SVM has the lowest accuracy of 90.48%, it still performs respectably, indicating its competency, but may not be as dependable as RF, DT, and XGB in certain scenarios.

XGB excels in precision with an impressive 99.92%, highlighting its ability to identify positive cases correctly with minimal errors. Furthermore, RF and DT exhibit high precision at 99.22% and 98.99%, respectively, indicating their effectiveness in making accurate positive predictions. However, SVM, with a precision of 95.86%, shows a strong ability to make accurate positive predictions, though it is slightly more prone to false positives than other models.

Moreover, DT leads in recall with 99.08%, indicating its superiority in identifying almost all positive instances. RF and XGB follow closely with recall rates of 98.91% and 98.97%,

respectively, demonstrating their strong performance in capturing positive cases. SVM, with a recall of 94.05%, indicates that it may miss some positive instances but still performs reasonably well in identifying positive cases.

Additionally, XGB achieves the highest F1-score at 99.13%, reflecting its exceptional balance between precision and recall. RF and DT also score high, with F1-scores of 99.07% and 99.03%, respectively, showcasing their well-rounded performance. The SVM, with an F1-score of 94.95%, while lower than the others, still demonstrates a balanced performance but highlights areas where it falls short compared to RF, DT, and XGB.

The confusion matrix presented in Figure 21.0 provides a comprehensive overview of the performance of RF, DT, XGB, and SVM models in classifying benign traffic. While all models demonstrated comparable accuracy rates, the SVM model identified 57,174 benign instances, achieving a TP rate of 94.06%, and demonstrated strong performance in classifying positive cases. However, it had a lower TNR of 83.08%, indicating difficulties in identifying non-benign traffic. In contrast, the RF, DT, and XGB models—DT with 50,228 instances, RF with 50,196 instances, and XGB with 50,163 instances—each achieved a TPR between 98.77% and 98.90%, highlighting their effectiveness in classifying benign traffic. These models also demonstrated higher TNR values, ranging from 95.77% to 97.10%, showcasing their ability to identify non-benign traffic compared to the SVM.

In the FN rate category, SVM has a rate of 5.94%, RF at 1.16%, DT at 1.10%, and XGB at 1.23%. Despite its highest TP rate, SVM's high FN rate means it misses more positive cases. Conversely, DT has the lowest FN rate, with RF and XGB close behind, demonstrating their robustness in minimizing missed positive cases.

Figure. 20.0. is the classification and misclassification rates analysis of the models of 251,277 instances. RF is the most effective, with a 98.38% correct classification rate (247,067 instances) and a 1.62% misclassification rate (4,210 instances). DT follows closely, with a 98.31% correct classification rate (247,006 instances) and a 1.69% misclassification rate (4,271 instances). XGB achieves a 97.45% correct classification rate (244,822 instances) and a 2.55% misclassification rate (6,455 instances). In contrast, SVM has a significantly lower correct classification rate of 88.48% (222,283 instances) and the highest misclassification rate at 11.52% (28,994 instances).

Figure 19.0. highlights the relative importance of features in classification. PSH Flag Count and Bwd Packets are the most influential, significantly impacting model performance. In contrast, features like FIN Flag Count and Bwd Avg Packets/Bulk have less impact.

In Figure 22.0, the ROC curve illustrates the performance of ML models on the CICIDS2017 dataset. The AUC values for DT, RF, and XGB approach the desirable threshold of 1, indicating an effective model for distinguishing between different classes. Notably, RF and XGB achieve the highest AUC scores, slightly above DT, while SVM has the lowest AUC value. Specifically, the AUC scores are 97% for DT, 98% for RF and XGB, and 89% for SVM.

These high AUC scores signify the strong predictive performance of the models on the CICIDS2017 dataset, underlining their robustness in class differentiation.

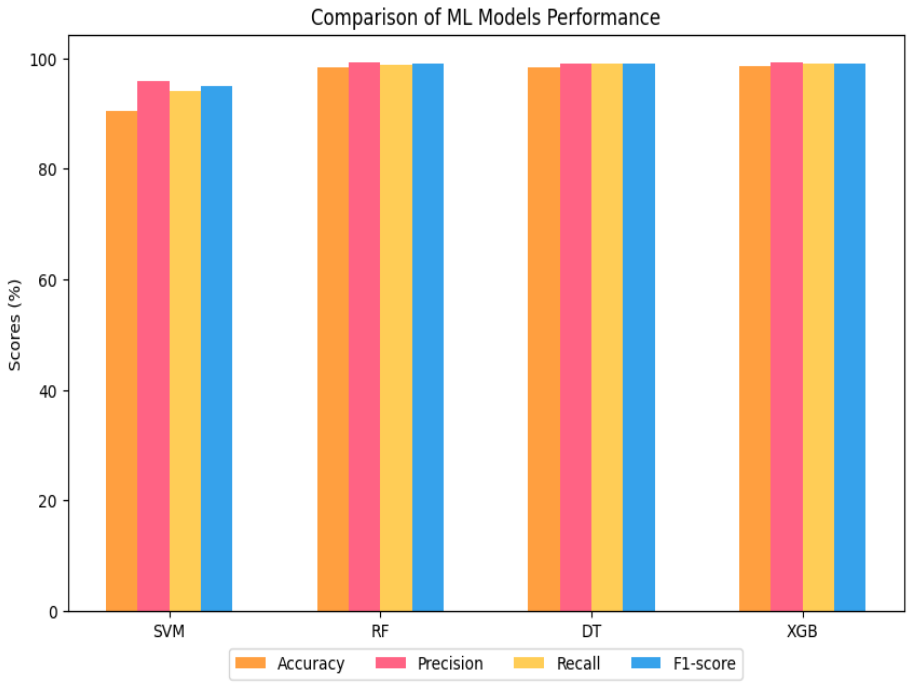


Figure 18.0. Bar chat of model performance on CICIDS2017 dataset.

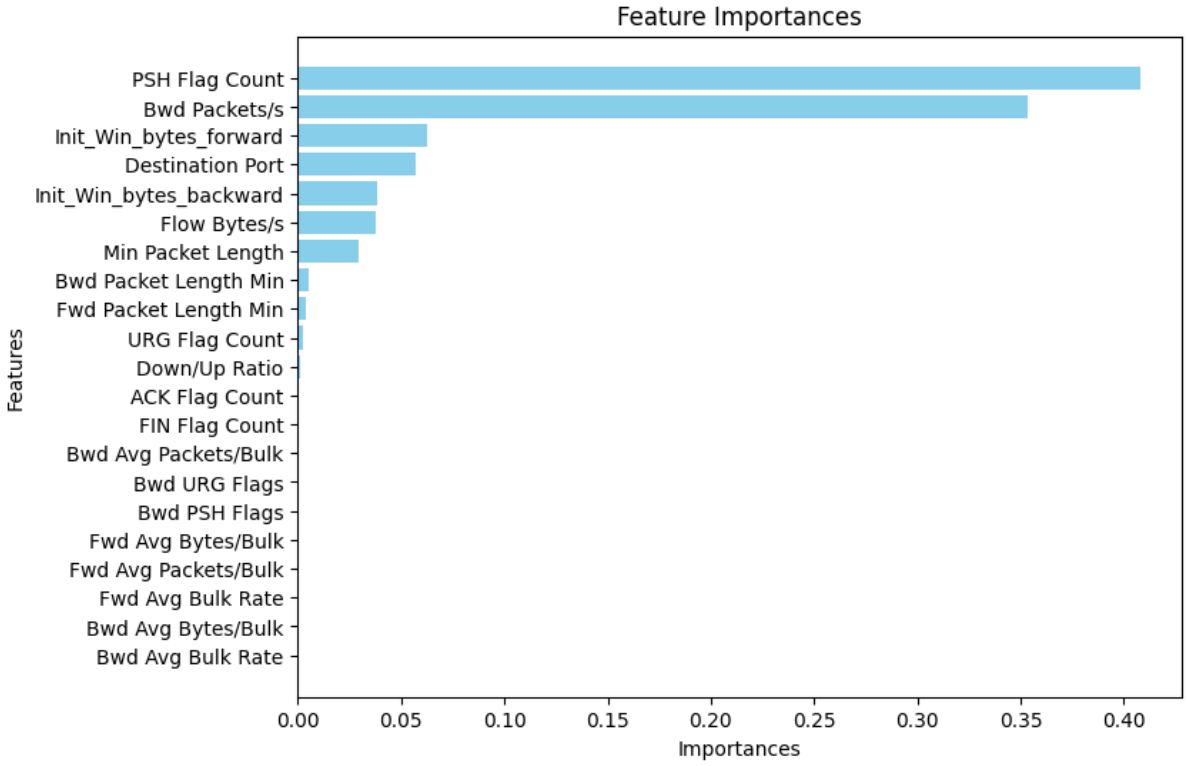


Figure 19.0. Feature importances of CICIDS2017 dataset.

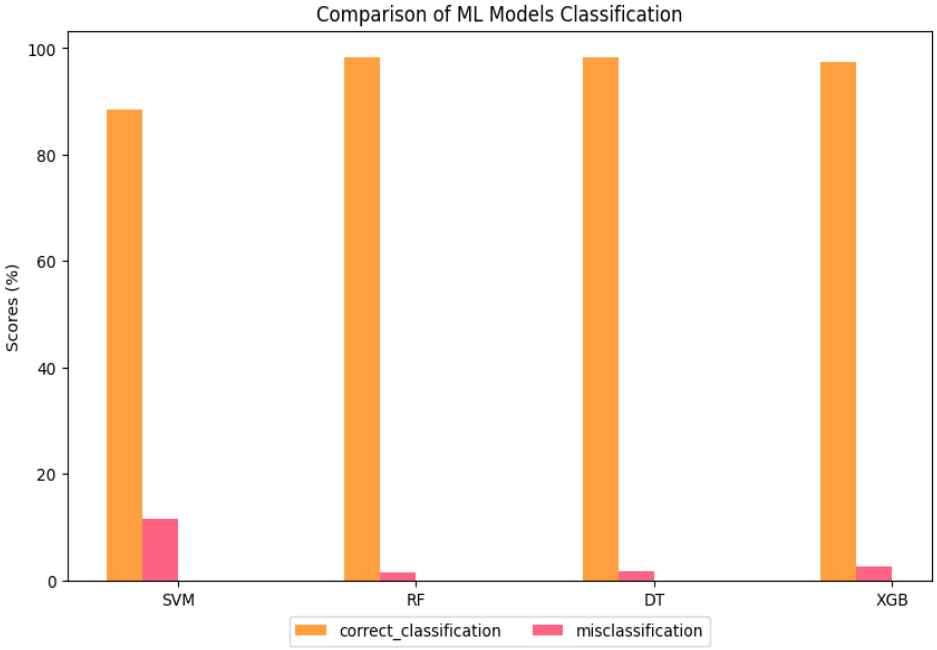


Figure 20.0. Classification and misclassification rate the models.

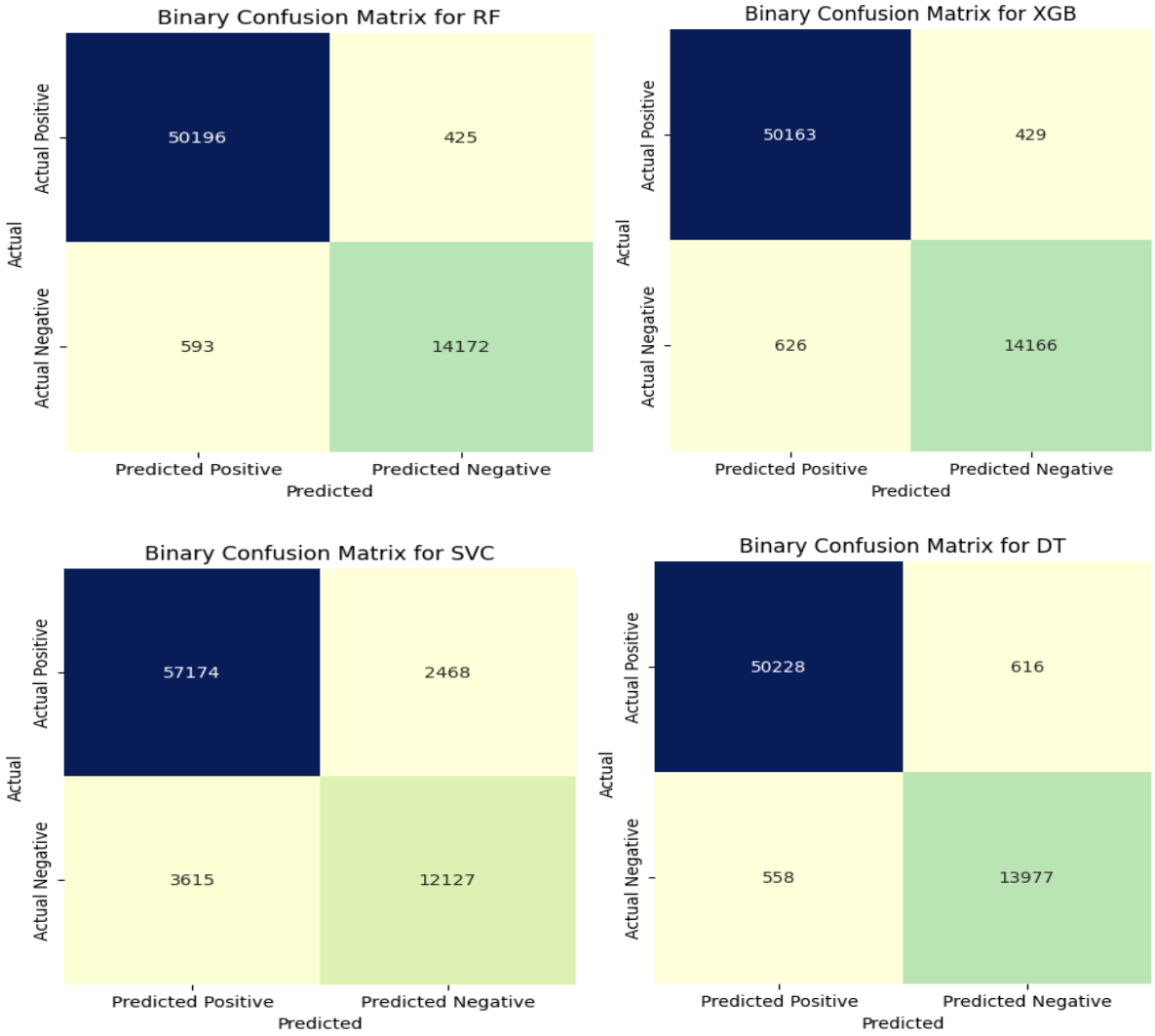


Figure 21.0. Confusion matrices of the models of CICIDS2017 dataset.

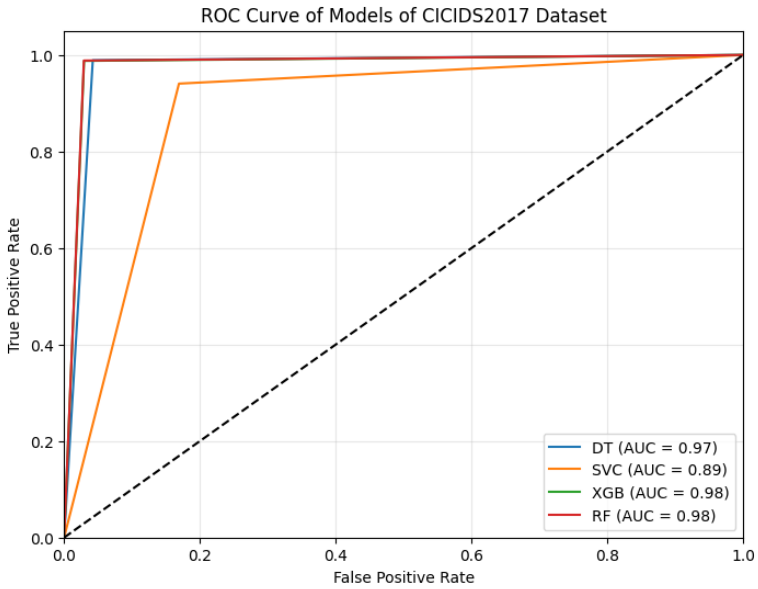


Figure 22.0. ROC-AUC curve of CICIDS2017 dataset.

Considering the performance analysis of the models on the two datasets, it can be concluded that the CICIDS2017 dataset yielded higher performance metrics for all models compared to the UNSW-NB15 dataset. This disparity may be due to differences in dataset characteristics, such as class imbalance and the nature of recorded attacks. Furthermore, models trained on the CICIDS2017 dataset demonstrated higher precision and recall, indicating better detection of normal and attack instances.

CHAPTER FIVE

DISCUSSION

5. Introduction

Our comprehensive analysis of machine learning models across two prominent datasets, UNSW-NB15 and CICIDS2017, reveals insightful patterns in their performance for network traffic classification. This study evaluated the performance of the machine learning models, RF, XGB, DT, and SVM, to improve threat detection and response in cybersecurity applications. By focusing on precision, recall, accuracy, and F1-score metrics, we provide a nuanced understanding of each model's capabilities across different scenarios.

The RF and XGB models demonstrated superior performance, achieving accuracy rates of 93% on the UNSW-NB15 dataset and over 98% on the CICIDS2017 dataset. These models excelled in threat detection, identifying anomalies and potential intrusions in network traffic.
Their high precision and recall rates show their effectiveness in accurately detecting threats while minimizing false positives and false negatives.

Furthermore, the high AUC values for RF and XGB (0.92 each for UNSW-NB15 and 0.98 each for CICIDS2017) underscore their strong ability to differentiate between benign and malicious network traffic, making them highly effective for real-world threat detection systems. In contrast, while Support Vector Machine (SVM) and Decision Tree (DT) models also performed well, their effectiveness was more dependent on the specific dataset, revealing limitations in their generalizability, especially with complex and imbalanced datasets.

Additionally, the confusion matrix analysis shows that while SVM effectively detects threats, it has a higher false positive rate. In contrast, RF and XGB offer a more balanced performance with lower false positive rates and strong true positive rates, making them better suited for practical cybersecurity applications where minimizing false alarms is crucial.
These results highlight the transformative potential of ensemble learning methods, including RF and XGB, in cybersecurity applications. Their ability to handle high-dimensional, complex data and their robustness in detecting various cyber threats underscore their suitability for real-time intrusion detection systems.

However, the performance variation of SVM and DT models across different datasets suggests that these models may require carefully selected features and tuning to achieve optimal results. This emphasizes the need for a tailored approach in machine learning for cybersecurity, with model and feature choices aligned with the network traffic characteristics and threat types.

5.1.Comparison with Previous Studies

Comparing the results of this study with existing literature, it is evident that the findings align with, and in some cases extend, existing research. Studies such as those by Aldweesh et al. (2020), Zhao et al. (2017), Talukder et al. (2024), and Ferrag et al. (2020) have highlighted the effectiveness of ensemble methods like RF and boosting techniques like XGB in handling large-scale and imbalanced cybersecurity datasets. This research corroborates these findings

and further demonstrates that these models not only perform well in controlled experimental settings but also show potential for practical deployment in real-world cybersecurity systems.

Moreover, the precision-recall trade-offs observed in SVM and DT models are consistent with findings in other studies, which have noted similar limitations when these models are applied to complex, high-dimensional cybersecurity data. This study adds to the body of knowledge by providing a more nuanced understanding of these trade-offs, particularly in real-time threat detection.

5.2.Implications for Theory and Practice

Our findings have significant implications for developing and implementing intrusion detection systems. The consistently high performance of RF and XGB confirms their effectiveness in cybersecurity, particularly in their ability to prioritize and utilize the most relevant features for accurate threat detection. This insight is crucial for the design and implementation of intrusion detection systems (IDS), as selecting the right features and ensuring accurate model interpretation can significantly enhance the system's effectiveness.

Moreover, the strong performance of ensemble methods in distinguishing between normal and malicious traffic—evidenced by high AUC scores—highlights their potential to reduce false positives, a common challenge in intrusion detection systems. This improvement could lead to more efficient and reliable security operations, reducing alert fatigue among cybersecurity professionals.

Additionally, features such as 'sttl' (source time-to-live), 'smean' (source mean packet size), and 'sloss' (source packet loss) in the UNSW-NB15 dataset, along with 'PSH Flag Count,' 'Bwd Packets/s,' and 'Init_Win_bytes_forward' in the CICIDS2017 dataset, were identified as highly influential. These features significantly enhance the models' ability to detect cyber threats. This insight is crucial for cybersecurity practitioners, as it identifies key network traffic elements indicative of malicious activity. Focusing on these critical features can make threat detection systems more targeted and efficient. By focusing on key features identified through models like RF and XGB, organizations can enhance the efficiency and accuracy of their threat detection systems, thereby improving overall cybersecurity posture.

5.3. Limitations

Despite the promising results, several limitations persist in the study. The reliance on benchmark datasets, while valuable, involves synthetic data that may not fully capture the complexity and variability of real-world network traffic. Additionally, the study focused exclusively on supervised learning models, which require labelled data for training. In practice, obtaining such data can be challenging, especially for new or emerging threats. Moreover, the study did not explore unsupervised or reinforcement learning techniques, which could offer additional benefits in detecting previously unknown threats.

5.4. Ethical Considerations and Challenges

The study also highlighted several ethical considerations, particularly concerning data privacy and model bias. The use of real network traffic data, even if anonymized, raises concerns about

potential privacy breaches. Ensuring that datasets are thoroughly anonymized, and that no personally identifiable information (PII) is exposed is critical to maintaining ethical standards.

Model bias is another significant challenge, especially given the imbalanced nature of datasets like CICIDS2017. Models trained on imbalanced data may be more likely to miss minority class threats or generate false positives, leading to a skewed understanding of the threat landscape. Techniques such as Synthetic Minority Over-sampling Technique (SMOTE) and careful evaluation of model performance across different classes are essential to mitigating these risks.

5.5.Future Research Directions

Based on the findings and the challenges identified, several recommendations for future research can be made.

Future research should explore the application of unsupervised and reinforcement learning techniques in cybersecurity, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), particularly in scenarios where labelled data is scarce or unavailable.

Additionally, investigating hybrid models that combine the strengths of different machine learning approaches could result in more robust and flexible threat detection systems. For instance, integrating the high precision of SVM with the robustness of RF could result in a model that balances accuracy, interpretability, and performance.

Finally, expanding the research scope to include large and continuously expanding real-world datasets will be crucial for understanding the practical application and scalability of these models. Diverse datasets will also be essential for validating the generalizability of these findings.

5.6.Contribution to Cybersecurity Field

This study significantly contributes to the cybersecurity field by providing a detailed comparative analysis of popular machine-learning models for network traffic classification. It highlights the consistent performance of RF and XGB across different datasets and provides insights into the dataset-specific strengths of DT and SVM.

By evaluating and comparing these models, the research provides valuable guidance for selecting the most effective machine-learning approaches for various cybersecurity applications, thereby enhancing the accuracy and reliability of threat detection systems.

Additionally, the feature importance analysis identifies key indicators for detecting network traffic anomalies, aiding in proactive threat mitigation and understanding attack strategies. The ability to identify and adapt to different types of discriminative features across datasets further underscores the complexity of network traffic classification and the necessity for adaptable models in cybersecurity.

CHAPTER SIX

6. Conclusion

This research project explored the application of machine learning techniques to enhance cybersecurity, focusing on network traffic classification for threat detection using two benchmark datasets: UNSW-NB15 and CICIDS2017. Through a comprehensive analysis of several ML models — RF, XGB, DT, and SVM, the study aimed to evaluate and compare their effectiveness in identifying cyber threats and optimizing ML-based threat detection systems.

The findings highlight the superior performance of ensemble methods, particularly RF and XGB, which consistently achieved high accuracy, precision, recall, and F1-scores across both datasets. Their strong discriminative power, evidenced by high AUC values, underscores their effectiveness in accurately identifying cyber threats while minimizing false positives and negatives, making them highly suitable for real-world threat detection systems. Attributed to this success is their ability to reduce bias and variance, effectively capturing the complex, non-linear relationships in network traffic data.

However, the study also revealed that model performance is significantly influenced by dataset characteristics. The improved results on the CICIDS2017 dataset suggest that clearer patterns or more reliable data can enhance model accuracy. This highlights the importance of using diverse and representative datasets in cybersecurity research to ensure the generalizability and robustness of the findings. The performance variance of the DT model across different datasets further illustrates the sensitivity of some models to dataset features, which can influence their applicability depending on the specific context.

While SVM and DT models showed consistent performance, their lower accuracy compared to RF and XGB indicates potential limitations in adapting to diverse traffic patterns. However, their high precision on the UNSW-NB15 dataset highlights their potential in scenarios where minimizing false positives is crucial.

In conclusion, this research confirms the potential of ensemble learning methods in enhancing cybersecurity threat detection. Their high accuracy, precision, and recall rates make them reliable choices for real-world applications. The insights gained from this study provide a strong foundation for the further development and optimization of machine learning-based security systems. Future research should integrate these models with more diverse and dynamic datasets and explore unsupervised and hybrid learning techniques to enhance the robustness and adaptability of cybersecurity solutions. This approach will advance the security and resilience of digital infrastructures in an increasingly complex and interconnected world.

REFERENCES

1. Abadi, M., Chu, A., Goodfellow, I., McMahan, H., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 308-318). https://doi.org/10.1145/2976749.2978318

2. Adebowale, M. A., Lwin, K. T., & Hossain, M. A. (2019). Deep learning with convolutional neural network and long short-term memory for phishing detection. pp. 1-8. https://doi.org/10.1109/SKIMA.2019.8917494

3. Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications, 60*, 19-31. https://doi.org/10.1016/j.jnca.2015.11.016

4. Ahsan, M., et al. (2022). Cybersecurity threats and their mitigation approaches using machine learning—A review. *Journal of Cybersecurity and Privacy, 2*(3), 527-555. https://doi.org/10.3390/jcp2030027

5. Aleesa, A. M., Younis, M., Mohammed, A. A., & Sahar, N. M. (2021). Deep-intrusion detection system with enhanced UNSW-NB15 dataset based on deep learning techniques. *Journal of Engineering Science and Technology*, *16*(1), 711-727.

6. Alissa, K. A., Alyas, T., Zafar, K., Abbas, Q., Tabassum, N., & Sakib, S. (2022). Botnet attack detection in IoT using machine learning. *Computational Intelligence and Neuroscience*, 2022. https://doi.org/10.1155/2022/9962874

7. Almseidin, M., Alzubi, M., Kovacs, S., & Alkasassbeh, M. (2017). Evaluation of machine learning algorithms for intrusion detection system. *2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)*, 277-282. https://doi.org/10.1109/SISY.2017.8080566

8. Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., ... & Van Dhun, A. (2019). A state-of-the-art survey on deep learning theory and architectures. *Electronics, 8*(3), 292. https://doi.org/10.3390/electronics8030292

9. Altwaijry, N., Al-Turaiki, I., Alotaibi, R., & Alakeel, F. (2024). Advancing phishing email detection: A comparative study of deep learning models. *Sensors, 24*(7), 2077. https://doi.org/10.3390/s24072077

10. Anderson, H. S., Kharkar, A., Filar, B., & Roth, P. (2017). Evading machine learning malware detection. *Black Hat*, 2017, 1-6.

11. Antonio, C., Dias, T., & Benthem, T. V. (2022). Illegal: The SolarWinds hack under international law. *European Journal of International Law, 33*(4), 1275-1286. https://doi.org/10.1093/ejil/chac063

12. Atawneh, S., & Aljehani, H. (2023). Phishing email detection model using deep learning. *Electronics, 12*(20), 4261. https://doi.org/10.3390/electronics12204261

13. Baker, K. (2023). Malware analysis. *CrowdStrike*. Retrieved from https://www.crowdstrike.com/cybersecurity-101/malware/malware-analysis/

14. Balzarotti, D., Cova, M., Karlberger, C., Kirda, E., Kruegel, C., & Vigna, G. (2010). Efficient detection of split personalities in malware. In *Proceedings of the 17th Annual Symposium on Network and Distributed System Security (NDSS 2010)*.

15. Barnard, P., Marchetti, N., & DaSilva, L. A. (2022). Robust network intrusion detection through explainable artificial intelligence (XAI). *IEEE Networking Letters*, *4*(3), 167-171.

16. Barolli, L., Xhafa, F., & Spaho, E. (2021). *Advances in intelligent networking and collaborative systems: The 2021 International Symposium on Intelligent Networking and Collaborative Systems (INCoS-2021)*. Springer.

17. Beh, Y. Z., & Lim, K. Y. (2024). Enhancing phishing website detection: A comparative analysis. In *2024 3rd International Conference on Digital Transformation and Applications (ICDXA)* (pp. 137-141).

18. Bernieri, G., Conti, M., & Turrin, F. (2019). Evaluation of machine learning algorithms for anomaly detection in industrial networks. In *2019 IEEE International Symposium on Measurements & Networking (M&N)* (pp. 1-6).

19. Bold, R.D., Al-Khateeb, H.M., & Ersotelos, N. (2022). Reducing False Negatives in Ransomware Detection: A Critical Evaluation of Machine Learning Algorithms. *Applied Sciences*. DOI: 10.3390/app122412941

20. Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2014). Network anomaly detection: methods, systems and tools. *IEEE Communications Surveys & Tutorials, 16*(1), 303-336. https://doi.org/10.1109/SURV.2013.052213.00046

21. Caleb, S., & Thangaraj, S. J. J. (2023). Anomaly detection in self-organizing mobile networks motivated by quality of experience. In *Fifth International Conference on Electrical, Computer and Communication Technologies* (pp. 1-6).

22. Cakir, B., & Dogdu, E. (2018). Malware classification using deep learning methods. In *Proceedings of the ACMSE 2018 Conference*.

23. Casas, P., Fiadino, P., & D'Alconzo, A. (2016). Machine-Learning Based Approaches for Anomaly Detection and Classification In Cellular Networks. *Traffic Monitoring and Analysis*.

24. Chesti, I. A., Humayun, M., Sama, N. U., & Jhanjhi, N. Z. (2020). Evolution, mitigation, and prevention of ransomware. In *2020 2nd International Conference on Computer and Information Sciences (ICCIS)* (pp. 1-6). https://doi.org/10.1109/ICCIS49240.2020.9242314

25. Codecademy. The Evolution of Cybersecurity. https://www.codecademy.com/article/evolution-of-cybersecurity.

26. Cybher. (2021). The Evolution of Cybersecurity: Where Did This All Begin? https://www.cybher.org/2021/06/14/the-evolution-of-cybersecurity-where-did-this-all-begin/

27. Dai, G., et al. (2015). SVM-based malware detection for Android applications. In Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks.

28. Dakota Murphey. (2019). A History of Information Security. IFSECGlobal, https://www.ifsecglobal.com/cyber-security/a-history-of-information-security/

29. Darko Galinec, Darko Možnik & Boris Guberina (2017). Cybersecurity and cyber defence: national level strategic approach, Automatika, 58:3, 273-286, DOI: 10.1080/00051144.2017.1407022

30. Dua, S., & Du, X. (2016). Data mining and machine learning in cybersecurity. CRC Press. https://doi.org/10.1201/9781315371597

31. Dwork, C., & Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science, 9*(3–4), 211-407.

32. Egele, M., Scholte, T., Kirda, E., & Kruegel, C. (2012). A survey on automated dynamic malware-analysis techniques and tools. ACM Computing Surveys, 44(2), Article 6, 42 pages. https://doi.org/10.1145/2089125.2089126

33. Elie, B. (2017). Inside the infamous Mirai IoT botnet: A retrospective analysis. *Cloudflare Blog*. Retrieved from https://blog.cloudflare.com/inside-mirai-the-infamous-iot-botnet-a-retrospective-analysis

34. Ford, V., & Siraj, A. (2014). Applications of machine learning in cyber security. Retrieved fromhttps://vford.me/papers/Ford%20Siraj%20Machine%20Learning%20in%20Cyber%20Security%20final%20manuscript.pdf

35. Gaspar, D., Silva, P., & Silva, C. (2024). Explainable AI for intrusion detection systems: LIME and SHAP applicability on multi-layer perceptron. *IEEE Access*. https://doi.org/10.1109/ACCESS.2024.3368377

36. Gonaygunta, H. (2023). MACHINE LEARNING ALGORITHMS FOR DETECTION OF CYBER THREATS USING LOGISTIC REGRESSION. *International Journal of Smart Sensor and Adhoc Network*. https://doi.org/10.47893/ijssan.2023.1229

37. Gorment, N., Selamat, A., Lim, K. C., & Krejcar, O. (2023). Machine Learning Algorithm For Malware Detection: Taxonomy, Current Challenges And Future Directions. IEEE Access. https://doi.org/10.1109/ACCESS.2023.3256979

38. Gulyás, O., & Kiss, G. (2023). Impact of cyber-attacks on financial institutions. *Procedia Computer Science, 219*, 84-90. https://doi.org/10.1016/j.procs.2023.01.267

39. Helmi, R., Elghanuni, R., & Abdullah, M. I. (2021). Effect the graph metric to detect anomalies and non-anomalies on Facebook using machine learning models. In *2021 International Conference on Signal Processing and Communication Systems (ICSPCS)* (pp. 7-12) https://doi.org/10.1109/ICSGRC53186.2021.9515227

40. Horchulhack, P., Viegas, E. K., & Santin, A. O. (2022). Toward feasible machine learning model updates in network-based intrusion detection. *Computer Networks, 202*, 108618. https://doi.org/10.1016/j.comnet.2021.108618

41. Hu, W., & Tan, Y. (2017). Generating adversarial malware examples for black-box attacks based on GAN. *arXiv preprint arXiv:1702.05983*.

42. Huang, K., Liu, C., Siegel, M., & Hao, S. (2022). Reinforcement learning for cyber deception. In *Proceedings of the 2022 ACM Asia Conference on Computer and Communications Security* (pp. 479-492). https://doi.org/10.1145/3488932.3517403

43. IBM, (2024). Types of Cyberthreats. https://www.ibm.com/think/topics/cyberthreats-types

44. Jha, S., Prashar, D., Long, H. V., & Taniar, D. (2020). Recurrent neural network for detecting malware. *Computers & Security, 99*, 102037. https://doi.org/10.1016/j.cose.2020.102037

45. Jiang, Y., Zhou, C., Guan, Y., & Gu, Y. (2022). A survey on deep reinforcement learning in cybersecurity. *Frontiers in Computer Science, 4*, 888164. https://doi.org/10.3389/fcomp.2022.888164

46. KAO, D.-Y., HSIAO, S.-C., & TSO, R. (2019). Analyzing WannaCry ransomware considering the weapons and exploits. In *2019 International Conference on Information and Communication Technology Convergence* (pp. 1098-1107). https://doi.org/10.23919/ICACT.2019.8702049

47. Kamiran, F., & Calders, T. G. K. (2012). Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems, 33*(1), 1-33. https://doi.org/10.1007/s10115-011-0463-8

48. Kang, B., Tan, V. Y., & Hosseini, H. (2019). Generating adversarial examples for malware detection using generative adversarial networks. In *Proceedings of the 2019 IEEE International Conference on Cybersecurity and Resilience (CyRes)* (pp. 1-4). https://doi.org/10.1109/CyRes.2019.8761961

49. Ketcham, M., et al. (2023). Spam text detection using machine learning model. In *2023 International Technical Conference on Circuits/Systems, Computers, and Communications (ITC-CSCC)* (pp. 1-6).

50. Kaushik, P., & Rathore, S. P. (2023). Deep learning multi-agent model for phishing cyber-attack detection. *International Journal on Recent and Innovation Trends in Computing and Communication*.

51. Khammas, B. M., Monemi, A., Bassi, J. S., Ismail, I. B., Nor, S. M., & Marsono, M. N. (2015). Feature selection and machine learning classification for malware detection.

52. Kouliaridis, V., & Kambourakis, G. (2021). A comprehensive survey on machine learning techniques for Android malware detection. *Information, 12*(5), 185. https://doi.org/10.3390/info12050185

53. Kumar, N., Sen, A., Hordiichuk, V., Jaramillo, M., Molodetskyi, B., & Kasture, A. (2023). AI in cybersecurity: Threat detection and response with machine learning. *Tuijin Jishu/Journal of Propulsion Technology, 44*(3), 38-46.

54. Kumar, A. (2014). Zero Day Exploit. Available at SSRN: https://ssrn.com/abstract=2378317 or http://dx.doi.org/10.2139/ssrn.2378317

55. Kumari, N. S., & Vurukonda, N. (2024). Support vector machine with grid search cross-validation for network intrusion detection in cloud. *International Journal of Intelligent Systems and Applications in Engineering, 12*(16s), 106-113.

56. Llauradó, D. G., Mateu, C., Planes, J., & Vicens, R. (2018). Using convolutional neural networks for classification of malware represented as images. *Journal of Computer Virology and Hacking Techniques, 15*, 15-28.

57. Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. *Applied Sciences, 9*(20), 4396.

58. Loftis, J. R., Mehrabi, N., Reiter, M., Lewis, J. M., & Galstyan, A. (2021). Fairness in model reliance: Identifying unwanted strategy incentives. *Proceedings of the AAAI Conference on Artificial Intelligence, 35*(7), 6252-6260.

59. Lok, L., Hameed, V., & Rana, M. E. (2022). Hybrid machine learning approach for anomaly detection. *Indonesian Journal of Electrical Engineering and Computer Science, 27*, 1016-1024. https://doi.org/10.11591/ijeecs.v27.i2.pp1016-1024

60. Losing, V., Hammer, B., & Wersing, H. (2018). Incremental online learning: A review and comparison of state-of-the-art algorithms. *Neurocomputing, 275*, 1261-1274.

61. M. A. Salitin and A. H. Zolait, "The role of user entity behavior analytics to detect network attacks in real time," In *2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, Sakhier, Bahrain, 2018, pp. 1-5. https://doi.org/10.1109/3ICT.2018.8855782

62. M. M. Silveira, et al., "Data protection based on searchable encryption and anonymization techniques," In *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, Miami, FL, USA, 2023, pp. 1-5. https://doi.org/10.1109/NOMS56928.2023.10154280

63. M. Sudhakar and K. P. Kaliyamurthie, "Machine learning algorithms and approaches used in cybersecurity," In *2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT)*, Bangalore, India, 2022, pp. 1-5. https://doi.org/10.1109/GCAT55367.2022.9971847

64. Makkar, A., Garg, S., Kumar, N., Hossain, M. S., Ghoneim, A., & Alrashoud, M. (2021). An efficient spam detection technique for IoT devices using machine learning. *IEEE Transactions on Industrial Informatics, 17*(2), 903-912. https://doi.org/10.1109/TII.2020.2968927

65. Malhotra, P., Vig, L., Shroff, G. M., & Agarwal, P. (2015). Long short-term memory networks for anomaly detection in time series. In *The European Symposium on Artificial Neural Networks*.

66. Maram, S. V. (2021). SMS spam and ham detection using Naïve Bayes algorithm. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3566078

67. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR), 54*(6), 1-35. https://doi.org/10.1145/3457607

68. Michael Houghton (2023). https://www.aztechit.co.uk/blog/cyber-security-trends

69. Minuchehr, A. (2023). Neural networks and security. LinkedIn. Retrieved from https://www.linkedin.com/pulse/neural-networks-security-ali-minoo/

70. Munir, M., Siddiqui, S. A., Dengel, A. R., & Ahmed, S. (2019). DeepAnT: A deep learning approach for unsupervised anomaly detection in time series. *IEEE Access, 7*, 1991-2005. https://doi.org/10.1109/ACCESS.2018.2886457

71. Murdoch, W. J., Liu, P. J., & Yu, B. (2018). Beyond word importance: Contextual decomposition to extract interactions from LSTMs. *arXiv preprint arXiv:1801.05453*. https://doi.org/10.48550/arXiv.1801.05453

72. Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., & Yu, B. (2019). Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences, 116*(44), 22071-22080.

73. Nguyen, P.-C., et al. (2021). An ensemble feature selection algorithm for machine learning based intrusion detection system. In *2021 8th NAFOSTED Conference on Information and Computer Science (NICS)* (pp. 50-54).

74. Niu, W., Zhang, X., Yang, G., Ma, Z., & Zhuo, Z. (2017). Phishing emails detection using CS-SVM. In *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)* (pp. 1054-1059).

75. Okoli, U. I., Obi, O. C., Adewusi, A. O., & Abrahams, T. O. (2024). Machine learning in cybersecurity: A review of threat detection and defense mechanisms. https://doi.org/10.30574/wjarr.2024.21.1.0315

76. Pajouh, H. H., Dehghantanha, A., Khayami, R., & Choo, K. (2018). A deep recurrent neural network-based approach for Internet of Things malware threat hunting. *Future Generation Computer Systems, 85*, 88-96. https://doi.org/10.1016/j.future.2018.03.007

77. Paper, R. P., Goyal, M., & Sharma, A. (2015). An efficient malicious email detection using multi naive Bayes classifier.

78. Pham, N.T., Foo, E., Suriadi, S., Jeffrey, H., & Lahza, H.F. (2018). Improving performance of intrusion detection system using ensemble methods and feature selection. *Proceedings of the Australasian Computer Science Week Multiconference*. https://doi.org/10.1145/3167918.3167951

79. Raheja, S., & Kasturia, S. (2022). Analysis of machine learning techniques for spam detection. In *Applications of machine learning in big-data analytics and cloud computing* (pp. 43-62). River Publishers.

80. Ren, J., Chaabouni, N., Zhou, W., Zhang, Z., Bhuiyan, M. Z. A., & He, X. (2019). Concept drift detection for network intrusion detection through feature selection. In *2019 IEEE International Conference on Communications (ICC)* (pp. 1-6).

81. Rieck, K., Holz, T., Willems, C., Düssel, P., & Laskov, P. (2008). Learning and classification of malware behaviour. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment* (pp. 108-125). Springer, Berlin, Heidelberg.

82. Rajora, K., & Abdulhussein, N.S. (2023). Reviews research on applying machine learning techniques to reduce false positives for network intrusion detection systems. *Babylonian Journal of Machine Learning*. https://doi.org/10.58496/bjml/2023/005

83. Sabir, B., et al. (2021). Machine learning for detecting data exfiltration: A review. arXiv. https://doi.org/10.48550/arXiv.2012.09344

84. Sahane, M. S. (2024). Spam detection in social networks using machine learning. *International Journal of Advanced Research in Science, Communication and Technology*.

85. Salahdine, F., El Mrabet, Z., & Kaabouch, N. (2021). Phishing attacks detection: A machine learning-based approach. In *2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)* (pp. 0250-0255). https://doi.org/10.1109/UEMCON53757.2021.9666627

86. Saxe, J., & Sanders, H. (2018). *Malware data science: Attack detection and attribution*. No Starch Press.

87. Sharma, A., Kumar, S., & Aslam, M. (2014). A comparative study between Naive Bayes and neural network (MLP) classifier for spam email detection.

88. Shen, Y., Zheng, K., Wu, C., Zhang, M., Niu, X., & Yang, Y. (2018). An ensemble method based on selection using bat algorithm for intrusion detection. *The Computer Journal, 61*(4), 526–538. https://doi.org/10.1093/comjnl/bxx101

89. Shilpashree, S. (2019). Decision tree: A machine learning for intrusion detection. *International Journal of Innovative Technology and Exploring Engineering, 8*(5). https://doi.org/10.35940/ijitee.F1234.0486S419
90. Shiravi, A., Shiravi, H., Tavallaee, M., & Ghorbani, A. A. (2012). Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Computers & Security, 31*(3), 357-374. https://doi.org/10.1016/j.cose.2011.12.012
91. Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence, 2*(1), 41-50. https://doi.org/10.1109/TETCI.2017.2772792
92. Shriram, S., & Sivasankar, E. (2019). Anomaly detection on shuttle data using unsupervised learning techniques. In *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)* (pp. 221-225). https://doi.org/10.1109/ICCIKE47802.2019.9004325
93. Sikorski, M., & Honig, A. (2012). *Practical malware analysis: The hands-on guide to dissecting malicious software*. No Starch Press.
94. Singh, P., Borgohain, S. K., & Kumar, J. A. (2022). Performance enhancement of SVM-based ML malware detection model using data preprocessing. In *2022 2nd International Conference on Emerging Frontiers in Electrical and Electronic Technologies (ICEFEET)* (pp. 1-4). https://doi.org/10.1109/ICEFEET.2022.9901618
95. Smith, M., & Mulrain, G. (n.d.). Equi-Failure: The national security implications of the Equifax hack and a critical proposal for reform.
96. Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In *2010 IEEE Symposium on Security and Privacy* (pp. 305-316). https://doi.org/10.1109/SP.2010.25
97. Sonare, B., Dharmale, G. J., Renapure, A., Khandelwal, H., & Narharshettiwar, S. (2023). E-mail spam detection using machine learning. In *2023 4th International Conference for Emerging Technology (INCET)* (pp. 1-5).
98. Stefano, L. (2023). Network anomaly detection using machine learning. Retrieved from https://thesis.unipd.it/retrieve/8470dabc-f701-4661-9b6e-d345f453f730/Leggio_Stefano.pdf
99. Subashini, K., & Narmatha, V. (2023). Develop a hybrid classification using an ensemble model for phishing website detection. *International Journal on Recent and Innovation Trends in Computing and Communication*.
100. Sudhakar, M., & Kaliyamurthie, K. P. (2022). Machine learning algorithms and approaches used in cybersecurity. In *2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT)* (pp. 1-5). https://doi.org/10.1109/GCAT55367.2022.9971847
101. Sumathi, M., & Raja, S. P. (2023). Machine learning algorithms-based spam detection in social networks. https://doi.org/10.21203/rs.3.rs-3069722/v1

102. Tang, T. A., Mhamdi, L., McLernon, D., Zaidi, S. A. R., & Ghogho, M. (2018). Deep recurrent neural network for intrusion detection in SDN-based networks. In *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)* (pp. 202-206). https://doi.org/10.1109/NETSOFT.2018.8460090

103. Tan, Z., Jamdagni, A., He, X., Nanda, P., & Liu, R. P. (2020). TRANSCEND: A transfer learning approach for deep-model updating in ensemble-based detection of cyber-anomalies. *IEEE Transactions on Information Forensics and Security, 16*, 1-1.

104. Talukder, S., & Talukder, Z. (2020). A survey on malware detection and analysis tools. *International Journal of Network Security & Its Applications, 12*(2), 37-57. https://doi.org/10.5121/ijnsa.2020.12203

105. Talukder, M. M. H., Ali, A., Nandy, S., Taha, T. M., Mohammed, S., & Berrached, A. (2024). Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction. *Journal of Big Data, 11*(33). https://doi.org/10.1186/s40537-024-00886-8

106. Torabi, Z. S., Nadimi-Shahraki, M. H., & Nabiollahi, A. (2015). Efficient support vector machines for spam detection: A survey. *International Journal on Recent and Innovation Trends in Computing and Communication, 13*(1).

107. Ustebay, S., Turgut, Z., & Aydin, M. A. (2018). Intrusion detection system with recursive feature elimination using random forest and deep learning classifier. *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*, 71-76. https://doi.org/10.1109/IBIGDELFT.2018.8625318

108. Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. *IEEE Access, 7*, 41525-41550. https://doi.org/10.1109/ACCESS.2019.2895334

109. Westyarian, Rosmansyah, Y., & Dabarsyah, B. (2015). Malware detection on Android smartphones using API class and machine learning. In *2015 International Conference on Electrical Engineering and Informatics (ICEEI)* (pp. 294-297).

110. Woźniak, M., Siłka, J., Wieczorek, M., & Alrashoud, M. (2021). Recurrent neural network model for IoT and networking malware threat detection. *IEEE Transactions on Industrial Informatics, 17*, 5583-5594. https://doi.org/10.1109/TII.2020.3021689

111. Xu, J., Wang, T., Zhu, Y., & Huang, T. (2019). Differentially private deep learning for network traffic analysis. In *2019 IEEE Conference on Communications and Network Security (CNS)* (pp. 1-9). https://doi.org/10.1109/CNS.2019.8802770

112. Xu, J., et al. (2021). An ensemble learning method with feature fusion for industrial control system anomaly detection. In *2021 33rd Chinese Control and Decision Conference (CCDC)* (pp. 2563-2567).

113. Yuan, X., Li, C., & Li, X. (2017). DeepDefense: Identifying DDoS attack via deep learning. In *2017 IEEE International Conference on Smart Computing (SMARTCOMP)* (pp. 1-8). https://doi.org/10.1109/SMARTCOMP.2017.7946998

114. Zeaadally, S., Adi, E., Baig, Z., & Khan, I. A. (2020). Harnessing artificial intelligence capabilities to improve cybersecurity. *IEEE Access, 8*, 23817-23837. https://doi.org/10.1109/ACCESS.2020.2968045

115. Zhaolin, C. (2020). Deep learning for cybersecurity: A review. In *2020 International Conference on Computing and Data Science (CDS)* (pp. 7-18). https://doi.org/10.1109/CDS49703.2020.00009

116. Zhao, J., Shetty, S., & Pan, J. (2017). Feature-based transfer learning for network security. In *2017 IEEE Military Communications Conference (MILCOM)* (pp. 17-22). https://doi.org/10.1109/MILCOM.2017.8170749

117. Zouina, M., & Outtaj, B. (2017). A novel lightweight URL phishing detection system using SVM and similarity index. *Human-centric Computing and Information Sciences, 7*, 1-13.

Appendix

Year/Author	ML Models	Dataset	Classification Report	Limitations	Key Findings
Anish Halimaa A and Dr. K. Sundarakantham (2019)	SVM and Naïve Bayes	NSL-KDD	SVM (97.29% accuracy, 2.705% misclassification) Naive Bayes (67.26%, 32.74% misclassification)	Only 19,000 instances of the dataset were used. Other metrics such as precision, F1-score, etc. were not considered.	SVM performed better than Naïve Bayes in terms of accuracy and misclassification Feature reduction and normalization techniques were applied to improve performance.
Talukder, M., Islam, M.M., Uddin, M.A., Hasan, K.F., Sharmin, S., Alyami, S.A., & Moni, M.A. (2024)	Decision Tree, Random Forest, Extra Tree Classifier	CICIDS2017, CICIDS2018, UNSW-NB15	CICIDS2017 DT, RF, ET (99.99% accuracy) CICIDS2018 DT, RF (99.94% accuracy) UNSW-NB15 DT, RF (99.95% accuracy)	The study did not employ deep learning models along with optimization techniques, which could further improve the performance of the intrusion detection system.	Stacking Feature Embedded (SFE) method enhances detection accuracy by incorporating meta-features. The study presents a novel intrusion detection approach using efficient preprocessing, oversampling, stacking feature embedding, and dimensionality reduction.
Kamran Shaukat, Suhuai Luo, Shan Chen, Dongxi Liu (2020)	Deep Belief Network (DBN) Decision Tree (DT) Support Vector Machine (SVM)	Spambase, Twitter dataset, Enron, malware datasets	DBN: 98.39% precision, 97.50% accuracy, 98.02% recall (on Enron dataset) DT: 98.00% precision, 96.00% accuracy, 94.00% recall (on Enron dataset) SVM: 95.20% precision, 93.60% accuracy (on Twitter dataset)	Lack of up-to-date benchmark datasets, especially for intrusion detection Existing datasets lack diversity and examples of sophisticated attacks. Machine learning classifiers themselves can be vulnerable to adversarial attacks.	ML techniques still lagging evolving cybercrime techniques. DBN generally performed well across tasks, especially for spam detection. DT showed the best accuracy (99.96%) for intrusion detection on KDD dataset. Need for new benchmark datasets with more diversity and modern attack examples

Mostofa Ahsan, Kendall E. Nygard, Rahul Gomes, M. Chowdhury, Nafiz Rifat (2021)	Bidirectional Long Short-Term Memory (Bi-LSTM), Gated Recurrent Units, Random Forest and a proposed Convolutional Neural Network and Long Short-Term Memory	UNSW-NB15 NSL-KDD	NSL-KDD (accuracy 99.54% to 99.64%) UNSW-NB15 (accuracy 90.98% to 92.46%)	Analyzing network data and building data-driven machine learning models is crucial for designing intelligent security systems.	The dynamic nature of cyberattacks requires constant support from experts and institutions to provide the latest datasets for training machine learning models, Future research should also explore the threats posed by quantum computing and its impact on public key encryption
Khalid Alissa, Tahir Alyas, Kashif Zafar, Qaiser Abbas, Nadia Tabassum, 2022	XGBoost, Logistic Regression, Decision Tree	UNSW-NB15	DT: 94.00% accuracy, 94.00% precision, LR: 78.00% accuracy, 73.00% precision, XGBoost: 94.00% accuracy, 94.00% precision	Machine learning-based botnet detection models are limited to the specific dataset they are trained on.	SMOTE was highlighted as an effective method to balance the dataset. This technique generated synthetic samples to balance the class distribution, improving model performance.
BAHRI Mohamed Ala Eddine; JEMILI Farah; KORBAA Ouajdi, 2023	Random Forest, Decision Tree, Logistic Regression, K-Nearest Neighbour, AdaBoost	CICIDS2017	RF (99.86% accuracy, 99.86% precision) DT (99.83% accuracy, 99.83%), AB (70.39% accuracy, 51.62% precision), KNN (98.60% accuracy, 98.62 precision) GNB (4.43% accuracy, 58.61% precision), LR (43.28 accuracy, 61.66% precision)	No specific limitation mentioned	- The machine learning and deep learning algorithms tested achieved over 99% accuracy in malware detection using the CICIDS2017 dataset. GaussianNB (GNB) had an accuracy of 4.43% and a precision of 58.61 %, indicating a significant misclassification of data points.

Ackah Benjamin, 2024	SVM, RF, DT, XGB	CICIDS2017, UNSW-NB15	CICIDS2017 SVM (90.48% accuracy, 95.86% precision) RF (98.44% accuracy, 99.22% precision) DT (98.38% accuracy, 98.99% precision) XGB (98.52% accuracy, 99.92% precision) UNSW-NB15 SVM (90.00% accuracy, 99.00% precision) RF (93.00% accuracy, 91% precision) DT (89.00% accuracy, 99.00% precision) XGB (93.00% accuracy, 94.00% precision)	The dataset was synthetic and may not fully capture the complexity of real-world network traffic. The study focused labelled data for training, which is costive and scarce. The study did not explore unsupervised or reinforcement learning techniques.	Ensemble method RF and XGB outperformed other models across both datasets, in terms of accuracy

Comparison performance metrics of ML models of existence studies and my model.

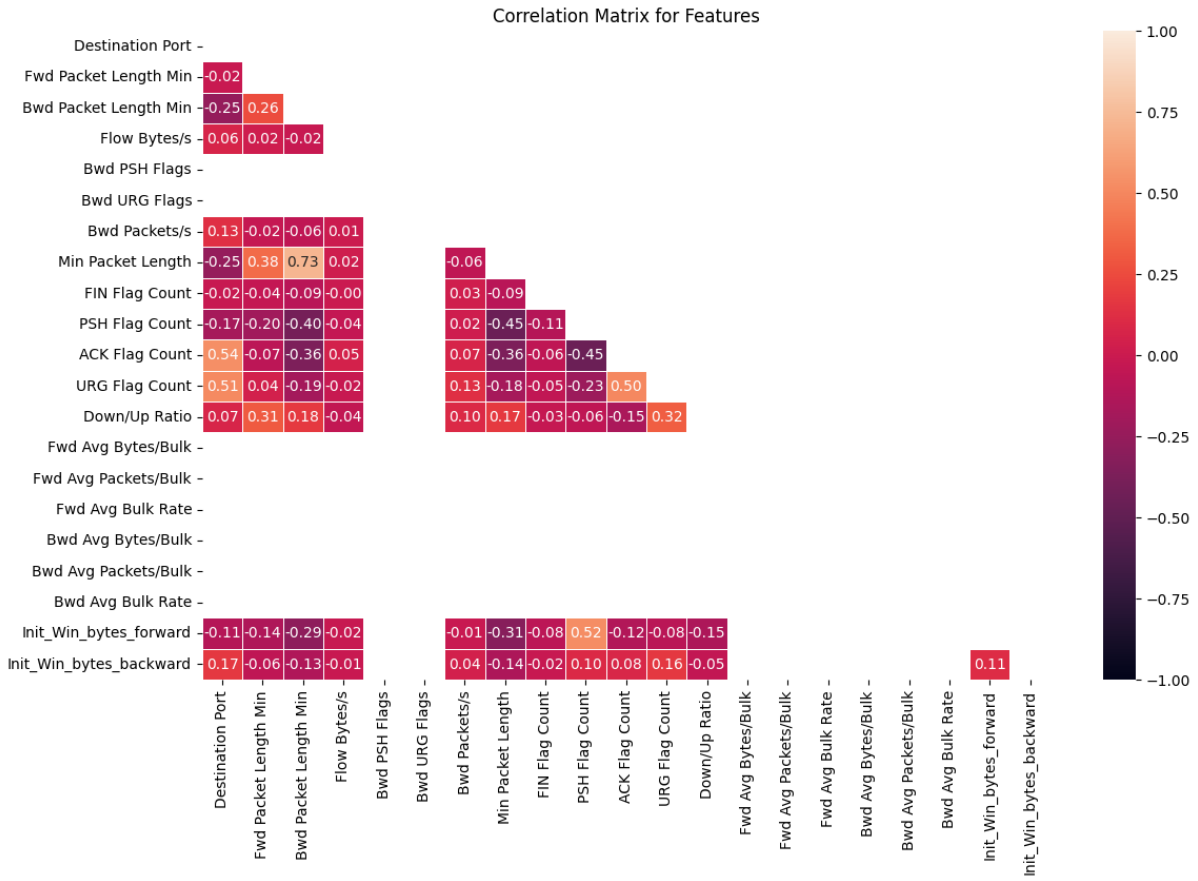
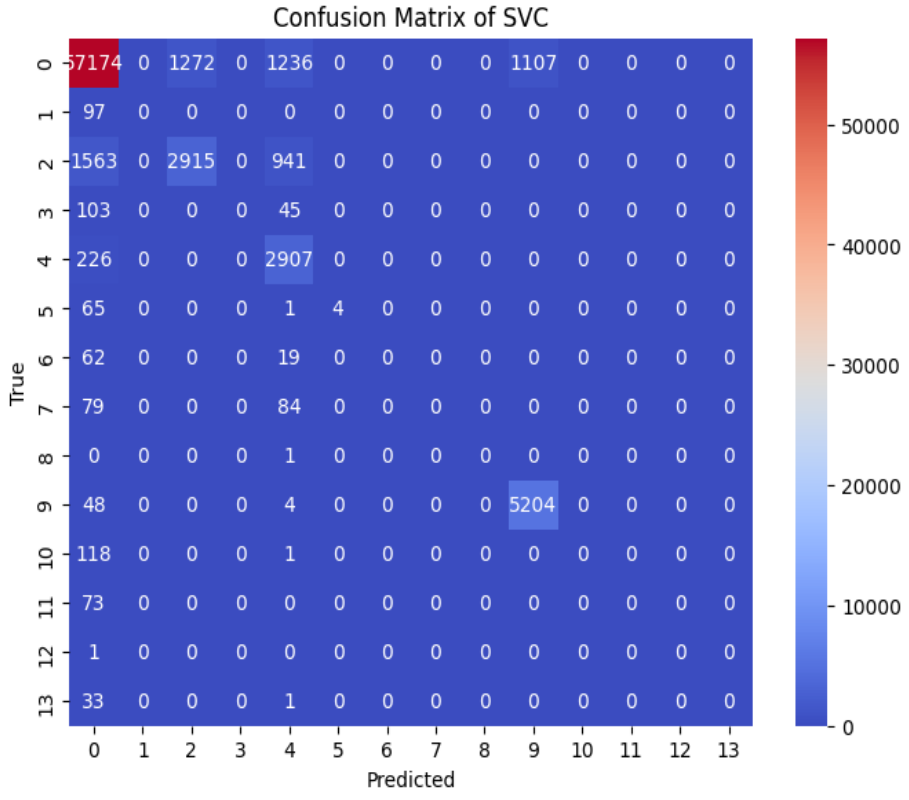
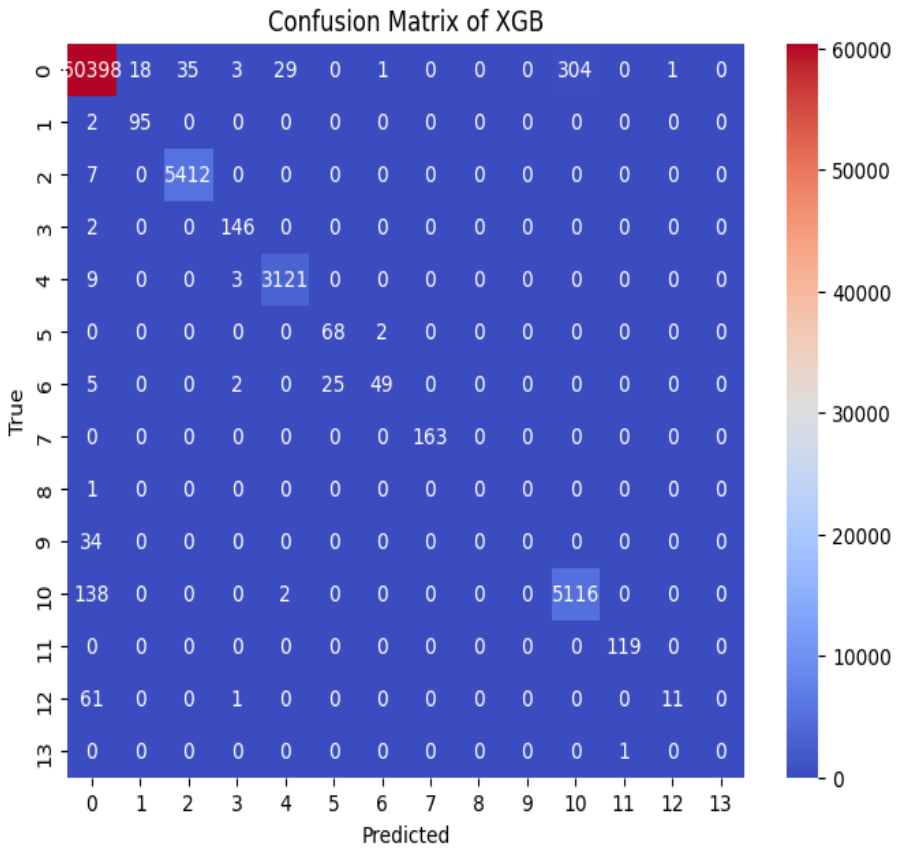
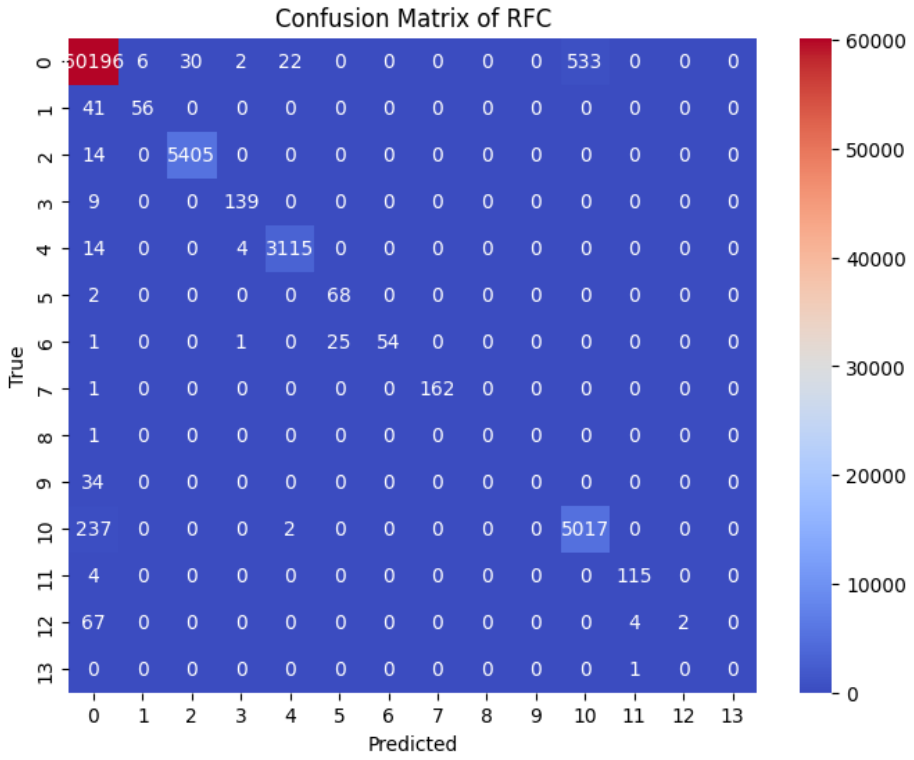
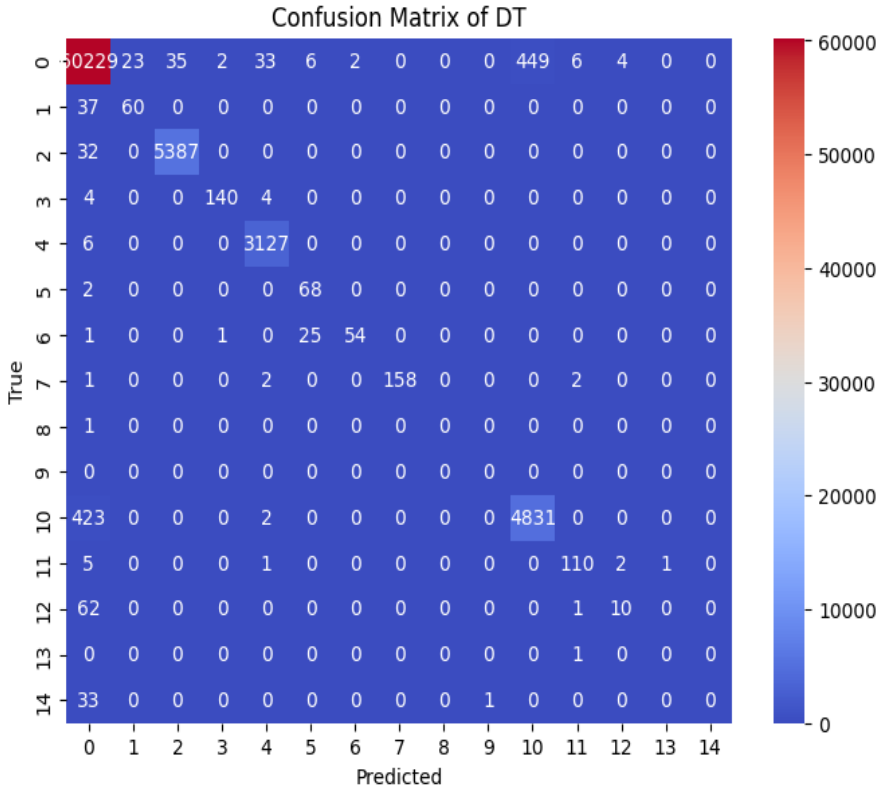


Figure 8.0. Correlation heatmap of the features of CICIDS2017 dataset.




Confusion Matrix of XGB
True
Predicted
0 1 2 3 4 5 6 7 8 9 10 11 12 13
60398 18 35 3 29 0 1 0 0 0 304 0 1 0
2 95 0 0 0 0 0 0 0 0 0 0 0 0
7 0 5412 0 0 0 0 0 0 0 0 0 0 0
2 0 0 146 0 0 0 0 0 0 0 0 0 0
9 0 0 3 3121 0 0 0 0 0 0 0 0 0
0 0 0 0 0 68 2 0 0 0 0 0 0 0
5 0 0 2 0 25 49 0 0 0 0 0 0 0
0 0 0 0 0 0 0 163 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0
34 0 0 0 0 0 0 0 0 0 0 0 0 0
138 0 0 0 2 0 0 0 0 0 5116 0 0 0
0 0 0 0 0 0 0 0 0 0 0 119 0 0
61 0 0 1 0 0 0 0 0 0 0 0 11 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0
60000
50000
40000
30000
20000
10000
0





Here is the full form of the acronyms used in the document:

- **ML**: Machine Learning