Leveraging Hybrid Intelligence for Robust Intrusion Detection in Evolving Networks

Submitted by

Nirupom Debnath

221-35-869

Department of Software Engineering

Daffodil International University

Supervised by

Dr. Rubaiyat Islam

Associate Professor

Department of Software Engineering

Daffodil International University

A thesis submitted in partial fulfillment of the requirement for the degree of Bachelor of Science in Software Engineering

Fall-2025

APPROVAL

This thesis titled on "**Leveraging Hybrid Intelligence for Robust Intrusion Detection in Evolving Networks**", submitted by **Nirupom Debnath (ID: 221-35-869)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS

Chairman

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

Internal Examiner 1

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

Internal Examiner 2

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

Internal Examiner 3

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

External Examiner

Md. Mostafiz Khan
Managing Director
Tecognize Solutions Limited

DAFFODIL INTERNATIONAL UNIVERSITY

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name	:	Nirupom Debnath
Date of Birth	:	20-May-2000
Title	:	Leveraging Hybrid Intelligence for Robust Intrusion Detection in Evolving Networks
Academic Session	:	2022-2025

I declare that this thesis is classified as:

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

I acknowledge that Daffodil International University reserves the following rights:

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

Certified by:

Nirupom
(Student's Signature)

221-35-869
Student ID
Date: 23/12/2025

(Supervisor's Signature)

Dr. Rubaiyat Islam
Name of Supervisor
Date: 23/12/2025

THESIS DECLARATION LETTER (OPTIONAL)

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

Dear Sir,

CLASSIFICATION OF THESIS AS RESTRICTED

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

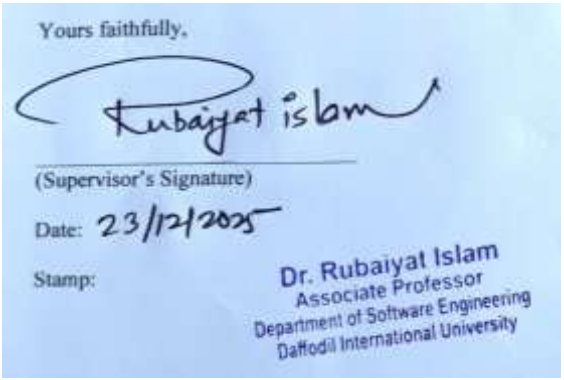
Author's Name
Thesis Title

Reasons

(i) The research outcomes are intended for future academic publication and intellectual property protection.

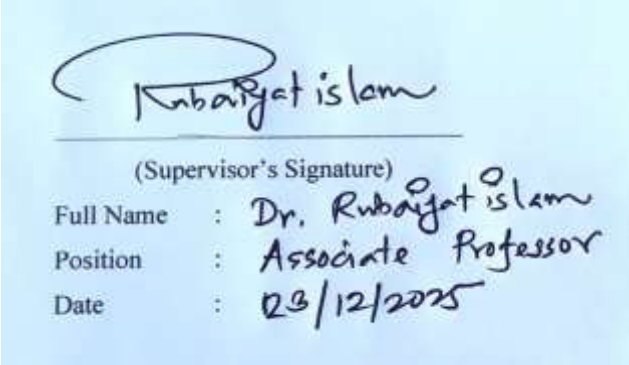
(ii) Public disclosure may enable misuse of the methodologies, datasets, and tools for exploiting real-world systems.

Thank you.



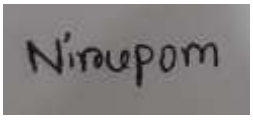
SUPERVISOR'S DECLARATION

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



STUDENT'S DECLARATION

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



(Student's Signature)

Full Name : Nirupom Debnath

ID Number : 221-35-869

Date : 23 December 2025

Leveraging Hybrid Intelligence for Robust Intrusion Detection in Evolving Networks

Nirupom Debnath

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

Department of Software Engineering (Major in Cyber Security)

DAFFODIL INTERNATIONAL UNIVERSITY

November 2025

ACKNOWLEDGEMENT

I am deeply indebted to all the people that have helped me along my academic way. Without their advice and support, this work would never have been on the form it is today.

This research was greatly influenced by the mentorship and friendship of my supervisor Dr. Rubaiyat Islam – thank you sir for teaching me to question convention and never settle with the norm! His impressive knowledge of the artificial intelligence sector gave me excellent perspectives to sort all aspects of my Thesis. His constructive criticism and encouragement continued to keep me motivated.

I am also grateful to the authors of the UNSW-NB15 dataset important benchmark for network intrusion detection systems. They made it possible for me to do in-depth analyses and compare my results to a strong baseline.

I also thank my colleagues and friends who provided their insights and suggestions when I was going through this work. Your cooperation, I believe, improved my research and pushed me to consider more deeply what I was doing.

I also want to thank my loving family and friends that never stopped believing in me. I learned a lot from their attitude, patience and support during the tough work of this research. I'm blessed to have people like that.

DEDICATION

First and above all, my deepest appreciation goes to Dr. Rubaiyat Islam, my respected supervisor. He was clear, fact based and delivered creative criticism that lead me through the depths of this project. His expertise about machine learning and intrusion detection systems was very helpful not only for to direct our research at the right path, but also to do high level quality work. How grateful I am for his direction on this journey.

I am also grateful to the authors of the UNSW-NB15 for providing their dataset. Their work is very significant, as it helps shaping a reliable and accurate dataset that can be employed when testing network intrusion detection systems. For purpose of those models as pursued here, the richness of a dataset based on so high is much more important for meaningful and comprehensive testing (and not less to reduce as well) thus it means crucial point to be incorporated as part of this research.

My special thanks go to my colleagues and other researchers for knowledge sharing, useful comments. Their contribution and feedbacks helped me to refine my ideas, adding quality to this study. So thankful for their assistance and cooperation!

Above all I am everlastingly grateful for the love and support of my family and friends. Their belief in what I was doing, even at times when I doubted it, have been crucial in getting through the toughest part of this study. They have been my rock, so grateful for their love, patience and for loving me.

Finally, to all who has in one way or the other contributed towards the realization of this work. You have been absolutely, genuinely and truly invaluable and I owe you all so much for getting behind this ridiculous little adventure.

ABSTRACT

The desire for more efficient and flexible Network Intrusion Detection Systems (NIDSs) has escalated due to the growing complexity and frequency of network-based intrusions. Most signature-based classical detection methods can solely recognize known malicious threats and are incapable of mitigating new and continuously emerging threats. The integration work is a new model proposed to refine the precision of the detection system and at the same time makes it more robust based on deep learning and machine learning. In this papers, we investigate several established machine learning algorithms such as SVM, RF and XGBoost for the proposed task. Furthermore, we compare deep learning methods such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks. Multiple models will be trained and tested on the UNSW-NB15 dataset. The dataset's rich network traffic and attack family features lend themselves well to running large-scale performance experiments of our system. The experimental hybrid model obtained the testing accuracy of 98.99% by leveraging the advantage of ensemble learning and deep learning at last. Our hybrid approach performed considerably better than both individual models in both unique and zero-day attacks. This work presents several promising new attributes such as network traffic profile based: byte ratio, packet ratio and protocol type.This work demonstrates that combination between deep learning and machine learning methods for intrusion detection provides a better model, Adaptation properties of the Deep Learning Neural Network over the Machine Learning lead to overcome the disadvantage when we are using the single model. The experimental results demonstrate that the hybrid model achieves better detection performance for different types of network attacks, what suggesting a solid base for further study in the field of cyber security.

Keywords —Network intrusion detection, Malware detection, Machine learning Techniques, Deep learning, Hybrid model, XGBoost, Random forest (RF), Convolutional Neural Networks (CNN), Long short-term memory (LSTM), OOF Stacking., Feature Engineering, Ensemble Learning approaches Cyber security.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

Malware is a huge problem for anyone concerned with cybersecurity, especially those who use Windows, as the Microsoft operating system powers so many systems around the world. The efficacy of signature-based detection methods has diminished as malware developers' strategies have evolved to be more complex. Among them are code obfuscation, polymorphism, and packing strategies. Scanning and signature-based detection can only identify known malware, rendering these methods ineffective against new or zero-day assaults [1].

Machine learning (ML) and deep learning (DL) techniques that can be trained using massive data sets to automatically detect malware behaviours have drawn more attention from researchers in an effort to get around these drawbacks.

Deep learning models using Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN) do exhibit the ability to automatically extract high-level patterns and sequential dependence from the binary or opcode format of malware.

This paper proposes a hybrid model that combines deep learning and gradient boosting, by using 1D-CNN, LSTM and XGBoost. As the result, it is constituted a stacking model of an out-of-fold ensemble model to extract significant feature shown in CNN, analyse temporal information that LSTMs are good at and predict well that XGBoost is strength. The aim is to develop a more robust and adaptable malware-detecting system on Windows platforms [2].

1.1 Backgrounds

Malware remains one of the most critical cybersecurity threats, and this threat remains pertinent, bearing in mind the dominance of the Windows operating system in global market share. The hallmark of signature-based detection systems is that they were specifically designed to detect malware via known patterns. However, these signature-based systems gradually became less effective due to rapid evolution processes that have helped malware evade detection, such as code obfuscation, polymorphism, and packing. Essentially, attackers change the malware code using these techniques to make it appear as a new, unknown version. Thus, signature-based systems will only identify previously identified threats, and are thus ineffective for novel or zero-day attacks [3] this lacuna, which led to anomaly-based detection systems. These use machine learning and deep learning techniques to identify normal network traffic patterns and find anomalies to the baseline. Anomaly-based models do not share this limitation and can identify new attacks they have never seen; therefore, such tools are becoming much more flexible to the dynamic character of cyber threats [4] [5].

1.1.1 What is Malware

Malware is any software intentionally designed to cause damage to a computer, server, client, or computer network. holist virus A holist; referring to a piece of software that meets the following attributes: It spreads automatically, under its own power and at a specified time, to another node. Malware can be used to capture sensitive information, interfere with the operation of systems, destroy or corrupt files or even remotely access and control a system.

Malware frequently travels by way of malicious email attachments, hacked web sites or tainted software downloads. This adware can be very hard for you to find it out and your antivirus doesn't have the ability to remove it completely. As cyber threats are constantly becoming more sophisticated, new malware strains emerge all the time, and cybersecurity solutions must evolve to address them in order to protect against, detect and respond to these risks.

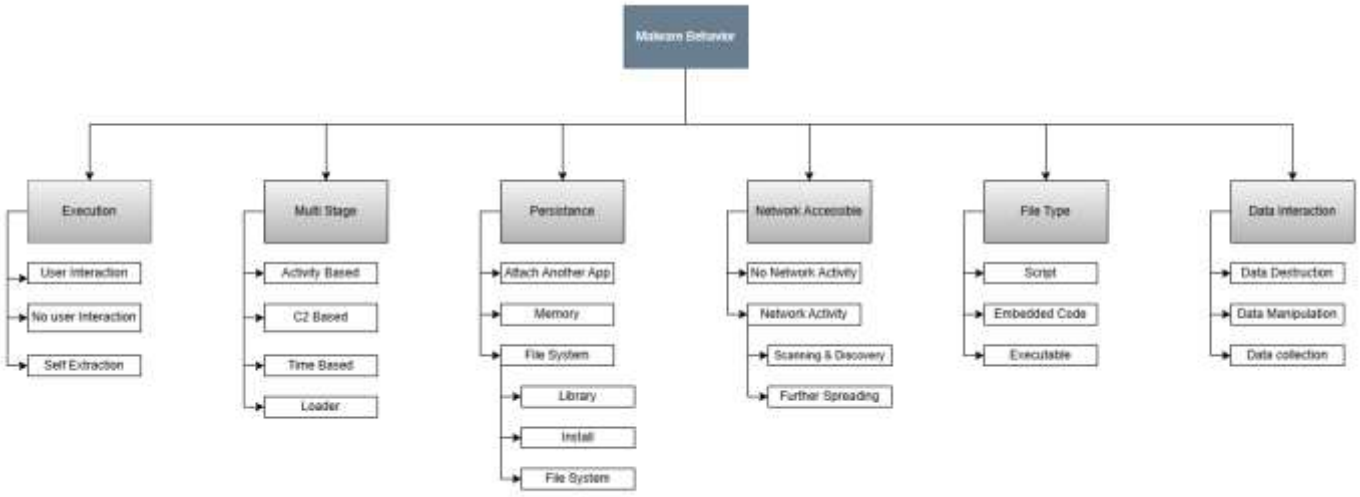


Figure 1.1: Malware Behaviour

1.1.2 How it works

Malware (malicious software) operates through a tactic of computer system or network vulnerability and exploitation to achieve its goals. The execution of malware usually can be analysed through a sequence of stages, each achieving its desired impact over the target system. These phases are described below:

- Infection and Delivery

  Malware is usually introduced to a system through different channels like phishing emails, malicious attachments, hijacked websites or infected downloads. Once the payload has been delivered, it can either run itself or ask the user to open a file or click on a malicious link.

- Execution and Payload Delivery

  After being executed, the malware executes its payload that can haunt sensitive data, intervene with or delete files, interrupt system work or create unauthorized remote access back to the infected computer. So there is ransomware (where files are encrypted and a ransom is demanded to release them) or spyware (which hoovers up personal or financial data for nefarious purposes).

- Persistence

  To remain in control of the infected system, malware is usually installed so that it will be re-triggered on each system restart. It disrupts the computer system, inserts itself to run on Windows start up or installs other adware elements that remain after removal.

- C&C Communication

  A considerable number of malwares, like botnets or advanced persistent threats (APTs), maintain a connection with one single Command and Control (C&C) server. Through this server, the attacker can remotely command, upgrade or steal exfiltration data from the malware. Such communications are likely crafted to

avoid analytical detection and can be further encrypted or obfuscated for malware persistence.

- Exfiltration or Exploitation

  Data exfiltration is one of the more popular objectives for malware, such as when stealing information or identities. Control It The data that leaves a compromised device, such as yours, is typically sent ‘back’ to the attacker’s server - stolen data like passwords, personal information etc. In some other cases, the malware can use system resources to perform extra malicious actions as launching DDoS attacks or installing other malware.

- Termination or Clean-up

  Depending on the type of malware, it can destroy or lock access to important data (for example if it were a wiper or ransomware) or compromise the system for additional exploitation at a later time. In a few cases, the malware may additionally erase itself from existence to avoid detection by security programmes. Though it is difficult to recover for removal.

Machine Learning has been widely used in intrusion detection systems using various algorithms such as Support Vector Machine (SVM), Random Forest (RF), and XGBoost for the classification of network data based on normal or malicious activities. These models are trained on labelled data, and perform well at identifying attacks within that training set. The main obstacle of classical machine learning methods is that they are unable to capture complex non-linear relationships in high dimensions such as traffic data. In addition, there is also aclass imbalance issue in many of the machine learning

models. This happens if the dataset has a lot of benign activity and very few attacks. Such work makes these tasks vulnerable to many false negatives, especially for novel or un-detected attacks.

Conversely, deep learning architectures, such as Convolutional Neural Networks and Long Short-Term Memory, offer a possible solution to autonomously finding complex patterns within large data volumes. CNNs are good at recognition of spatial patterns in data, and since local dependencies (e.g., size) play a significant role in attacking, CNNs help improve the performance for attack recognition. different categorizations of protocols for network traffic characteristics. BANE-R uses LSTMs, which are good at handling series data like network traffic that has a time-dependent nature and helps in recognising the risks that occur in sequence or have long term dependency. Despite the exceptional accuracy of deep learning models, their effectiveness depends on large datasets and substantial computational resources, which are not available in all practical deployment scenarios [6].

The dataset comprises nine attack categories, including DoS, DDoS, Port Scanning, and Botnet, alongside normal traffic [7]. It possesses all the attributes desirable in a dataset for evaluating various detection models. Unlike prior datasets like KDDCUP99, which are outdated and lack expressiveness owing to evolving attack patterns, UNSW-NB15 presents a sophisticated and more realistic operational network environment [8] .However, traditional machine learning techniques face difficulty considering the high-dimensional dataset suffers class imbalances: most of its traffic is benign and outnumbers malicious activity. Therefore, models making use of the hybrid power by integrating some most popular ensemble learning techniques such as XGBoost [9] and

Random Forest, [10] with deep learning methods like CNNs and LSTMs have been presented in the literature. Specifically, the ensemble approaches could help to address the issue of imbalanced data by exploiting not only integrated features from different models but also produce more accurate and robust attack detection. Deep learning techniques can characterize and learn intricate nonlinear correlations among different samples of data, which might improve the performance in subtle attack pattern identification. Such hybrid models promote the precision and adaptability of intrusion equipped to rapidly detect a wide range of known and unknown network threats [11].

1.1.3 Malware Detection Techniques

Malware detection is critical to the discovery and suppression of malicious software which threatens the health of computer systems and networks. A number of malware-detection approaches have been made over the years with different features and usages.

- Signature-Based Detection

  Signature Based: This is the oldest, most popular technique. It functions by checking the files or programs on a computer against a database of known malware signatures, which are specific codes or bytes that indicate the presence of particular pieces of malware. If a file or program matches one of these signatures, it is potentially malicious.This approach is capable of identifying existing malware, however, it cannot detect new or previously unseen variants of the malware unless their signatures have been incorporated into the database.

- Heuristic-Based Detection

  Heuristic detection is an analysis based on a program's characteristics or behaviour that may indicate presence of malware. While signature-based means discarding the search for known patterns and heuristics come into play, through which we go searching suspicious behaviour. For instance, the system could search for actions such as illicit modification of files, attempts to obscure processes or changes of system configuration.This approach can find new and unknown malware by its behaviour; however, it could have false positives because a normal application can behave in a similar way to the malicious one.

- Behavioral-Based Detection

  Behavioral is a detection that intercepts the behaviour of a program as it runs. This technique searches for indications of suspicious activity such as access to sensitive files, changing critical system settings or the setting up of unauthorized network connections. The system monitors how the program behaves, and will classify it as bad if anything suspicious occurs.This approach is more efficient for uncovering attacks in progress, as it looks at what the software does while running rather than focusing strictly on its code or structure.

- Machine Learning and Deep Learning for Malware Detection

  Detection of malware has been improved greatly using machine learning (ML) and deep learning (DL). These techniques enable systems to learn from the massive data of benign and malicious software, which could lead to malware detection without predefined signatures.

- Supervised Learning

  In supervised learning, the model is trained on labelled dataset and the system has examples of good ware-malware. The model learns to distinguish the two by recognizing patterns in (features of) the data. Once it's been trained, the model should be able to tell you whether new, unseen files or programs are malware.

- Unsupervised Learning

  Unsupervised learning, in contrast, does not need labelled data. Rather, it detects abnormal behaviours or outlier activities in the system. It will help to detect unknown malwares, because it is not based on signature of any type of malwares.

- Deep Learning Models

  Models based on deep learning, such as CNNs and RNN(Recurrent Neural Networks), are very powerful to process raw, unstructured data like binary code or system logs. Such models naturally pull out deep features and complex structures from the data, which is very suited for detecting sophisticated, evasive malware.

- Hybrid Malware Detection Models

  There are hybrid models used to enhance the overall effectiveness through combining several detection techniques. For instance, a hybrid system could use signature-based detection to detect known threats and machine learning based or deep learning-based approach for unknown threat discovery. Such models combine multiple approaches for better malware detection.

- Cloud-Based Malware Detection

  Cloud-based detection systems shift the task of analysing malware to the cloud, which benefits from abundant computational resources to scan and analyse data sets comprised of samples installed or used in many systems. Cloud detection Cloud-based detection utilizes cloud computing to detect threats on a variety of platforms creating more effective and scalable protection.

- Sandboxing

  Sandboxing refers to executing known or potential malware in an enclosed environment (or a sandbox), so that the behaviour of the malware can be studied without damaging the actual system. Security researchers can then examine this software' s interactions with the system to determine if it is 'malicious. This method is particularly valuable in a study of unknown or new forms of a malicious software. Now we will see malware detection techniques [12] :

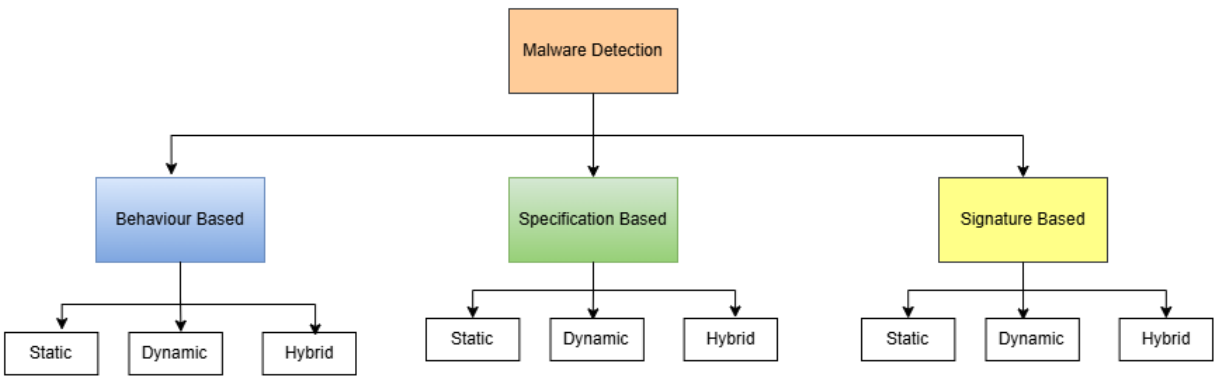


Figure 1.2: Malware Detection Techniques

1.2 Problem Statement

Traditional intrusion detection systems (IDS) with signature-based detection, especially zero-day attacks that are novel, unknown or unreported, have been more and more ineffective in detecting. The signature-based Intrusion Detection Systems rely on known signatures of already-detected attacks, which means that the systems can only detect the malware matching with these previously attacked signatures. As cyber threats have evolved, attackers are having more and more to utilize advanced techniques such as code obfuscation, polymorphism and packing in order to hide their malicious activities from the network intrusion detection systems (IDS) that use pattern matching with regular expressions according. This limitation highlights the need for more flexible intrusion detection systems [12].

ML models such as RF and XGBoost have already showed their performance in malware detection [13], achieving higher accuracy than traditional sig- nature-based approaches. Despite their successes, these models have some limitations. They are generally not directly applicable for high-dimensional data, ignore the (potentially) complex inter-feature relationships and do not handle class imbalance problems across the object database. Deep learning models, including CNN and LSTM, can alleviate this issue by directly extracting intricate patterns from raw data. Nonetheless, the training process of deep learning models necessitates extensive labelled datasets and can often be computationally intensive; therefore, deep learning [13] techniques are frequently impractical for use in several contexts.

Considering that no singular algorithm, including extensive deep learning models, can adequately discern complex patterns and that the computational requirements of large-

scale deep learning models are considerable, there is a necessity for the amalgamation of robust machine learning methods with deep learning techniques. This study examines the combined use of XGBoost and Random Forest as ensemble learners, in conjunction with CNN and LSTM as deep learners, to develop a more robust and efficient solution for network intrusion detection. This hybrid model is expected to exceed existing models in identifying both recognized and novel network assaults while preserving minimal computational expenses.

Therefore, the primary challenge we explore is to design a hybrid intrusion detection model (i.e., constructed upon ensemble learning and deep learning network) for improving accuracy, generalization capability, and efficiency under malware detection task even with high-dimension data or class-imbalanced datasets.

1.3 Research Gap

1. **Ineffective Signature-Based Detection:**

   Traditional signature-based detection systems are limited in their ability to recognize novel, zero-day attacks and polymorphic malware due to the evolution of methods like code obfuscation and packing. These methods are unable to adjust to new attack patterns and can only identify dangers that have been previously documented[14].

2. **Challenges in Machine Learning Models:**

   However, the majority of machine learning models, such as RF and XGBoost, exhibit promising outcomes yet are hindered by high-dimensional data and class-imbalance issues; thus, they demonstrate bias in identifying uncommon attack

instances and subsequently perform inadequately against novel attacks. [15]. And also, the relationships among data are hard to learn for simple models which are demanded in advanced detection schemes.

3. **Limitations of Deep Learning Models:**

Deep learning models, including CNNs and LSTMs, are able to discover complex patterns in data but need large labelled datasets and much computational power. As a result, they are not favourable for devices with low-bound computation or small datasets.

4. **Not enough Hybrid Model Exploration**

Hybrid models that integrate ensemble learning (e.g., Random Forest and XGBoost) with the deep learning architectures, including CNNs and LSTMs, have been introduced; however, these methods are novel to some extent in general and particularly in network intrusion detection. However, the majority of studies on hybrid models fail to well address creative threat detection and computational efficiency issues when handling high-dimensional and imbalance data.

5. **Lack of Advanced Feature Engineering:**

Feature engineering is therefore very few in state-of-the-art models. Many research studies find their interest in very basic features from network traffic, which makes it tough on the model to identify advanced vectors of an attack.

There remains a scope for more discriminate features which will help enhance complex malware pattern detection and intrusion detection over the network [16].

1.4 Dataset Overview

The UNSW-NB15 is a prominent network traffic dataset created for the training and evaluation of machine learning algorithms on network traffic. The UNSW-NB 15 dataset's architecture must fulfil the specifications for Network Intrusion Detection System (NIDS) model creation. The dataset comprises raw network packets recorded using the Tcpdump tool in the Cyber Range Lab, subsequently replayed across various attack scenarios with IXIA Perfect Storm. The dataset is crucial for investigating modern network threats because of its combination of authentic and synthetic advanced attack behaviours.

There are four files comprising over 2.5 million records to process from the UNSW-NB15 dataset. 175,341 records are designated for the training set, whereas 82,332 records are allocated to the testing set. The dataset consists of 49 features obtained from twelve methodologies, employed to classify network traffic as either normal or one of many attack kinds. The UNSW-NB15_features.csv file provides additional details concerning the columns. Attack Categories:

1. Fuzzers

   Description: Fuzzing is the process of giving an application random or semi-random data as an input to find security vulnerability. This attack attempts to

discover software vulnerabilities by feeding random or unexpected input the target system, possibly causing it to crash.

2. Analysis

   Description: Analysis attacks extract network data in order to learn how the target system is organized and behaves. This could include applications to map the network, discover its weaknesses or gather information ripe for a future assault.

3. Backdoors

   Description: Backdoors are used to bypass normal security mechanisms, which are usually found in servers and other types of computers where the backdoor application exists. Backdoors can allow attackers to have continued access and perform malicious activities on compromised systems.

4. Denial of Service (DoS)

   Description: A Denial of Service (DoS) attack aims to incapacitate a system or network, rendering resources inaccessible to authorized users. The objective of a DoS attack is to inhibit a service from functioning as intended, often leading to downtime.

5. Exploits

   Description: An exploit attack is one in which an adversary exploits a known software or hardware vulnerability to penetrate a target system without the consent of the user. Usually, these are used as a vector to further own systems and enhance an intruder's blast radius.

6. Generic

   Description: The category "Generic" contains all attacks which do not belong to any of the above categories. These threats can be one of a number of adversarial conducts such as virus and malware, which may take the disguise of a trojan horse to attack the confidentiality, integrity or availability of network.

7. Reconnaissance

   Description: Reconnaissance attacks consist of information gathering regarding a network or system, including structure and configuration, and potential vulnerabilities. Such attacks are often a prelude to more focused, specialized attacks and used as a preparation stage for an exploitation

8. Shellcode

   Description: Shellcode is a concise code segment employed as the payload in exploiting a software vulnerability. Upon execution, shellcode typically initiates a command line shell, enabling the attacker to execute arbitrary commands on the victim's computer.

9. Worms

   Description: Explain the worm functionality Worms Self-replicating program which are distributed over network without the human intervention. Worms take advantage of flaws in systems or network protocols to spread and regularly result in widespread harm such as system crashes and compromised sensitive data.

1.5 Dataset Structure and Features

UNSW-NB15_1. csv to UNSW-NB15_4. csv: These files consist of raw network traffic with attack and normal record. The dataset consists of 49 features produced by a variety of network monitoring tools including Argus and Bro-IDS.

- UNSW-NB15_GT. csv: This file includes the true labels and shows that whether a behaviour record is normal or belongs to certain attack types.

- UNSW-NB15_LIST_EVENTS. csv: This file comprises a collection of events and their details within the dataset, enabling the detection of specific attack patterns and traffic characteristics.

- UNSW_NB15_training-set. csv: This dataset was employed to train machine learning models for the detection of network intrusions and consists of 175,341 records.

- UNSW_NB15_testing-set. csv: To valuate model's performances, the testing set consisting of 82,332 samples are applied.

Due to the diversity and richness of attack categories in the data, together with its extensive set of features, KDDCUP 99 is a good testbed for evaluating intrusion detection systems (IDS) such as current generation network threats that are complex. The attack categories present in the dataset vary from basic forms of DoS (Denial of Service) and exploitation attempts up to complex threats such as backdoors, fuzzers and worms. This variety enables IDS models to learn and differentiate between different malicious behaviours and to detect known attack signatures as well as new/unseen attack strategies (the generalization property).

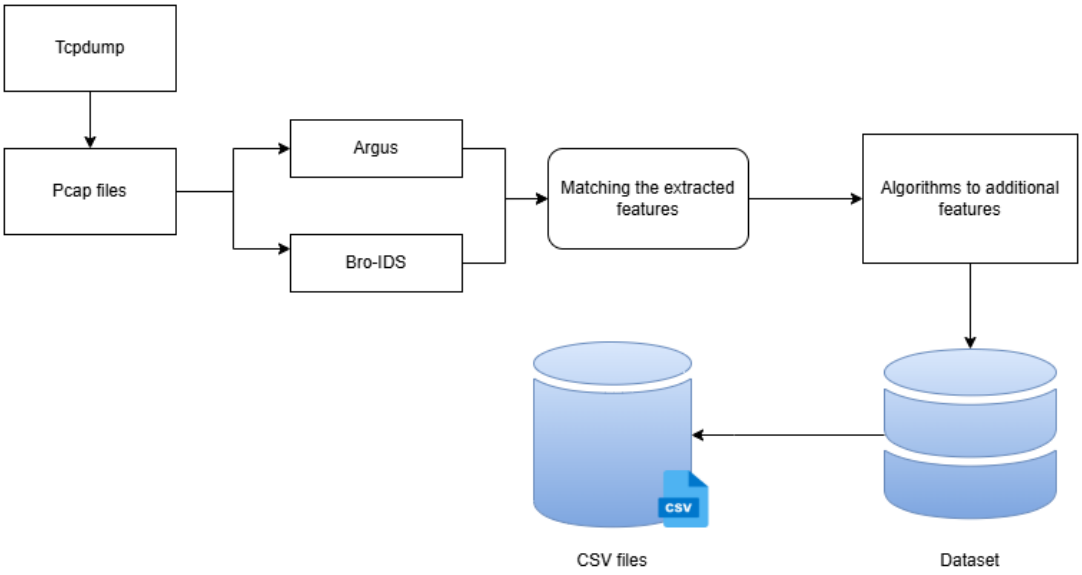


Figure 1.3: Architecture for Generating UNSW-NB15 data set

1.6 Research Objective

- Ineffectiveness of Signature-Based Detection:

  Standard methods for detection using signatures have limitations on discovery of new zero-day attacks and polymorphic malware using evolving methods like code obfuscation and packing. These can only recognize previously known threats as they can't detect attacks based on completely new patterns [17].

- Evaluate deep learning models:

  The aim is to evaluate the efficacy of deep learning models, namely Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, in recognizing complex causal patterns in network traffic and new viruses or attacks that previous models could not recognize.

- Implement a hybrid model:

  To propose a hybrid model that amalgamates ensemble learning techniques, such as XGBoost or Random Forest, with deep learning structures, such as Convolutional Neural Networks or Long Short-Term Memory models, to augment the accuracy, robustness, and generalization of malware detection. Hybrid model tries to enhance detection capability against zero-day or unknown threats, along with the network behaviours complexities [18].

- Advancing feature Engineering:

  To propose and evaluate new feature traffic features such as byte ratio, packet ratio, and protocol type, to improve model performance through extracting more relevant patterns, and behaviour associated with malicious activity.

- Overall Performance:

  To assess the overall effectiveness of the hybrid model that combines machine learning and deep learning compared to standalone machine learning and deep learning, employing various evaluation metrics such as accuracy, precision, recall, and F1 Score in detecting different network intrusions and malware.

- Assessing Scalability and Efficiency:

  We have conducted experiment to measure the scalability and computation efficiency of our hybrid model with respect to training time, inference speed, and resource consumption as well as the comparison of cost, efficiency and performance of hybrid model versus other state-of- art models.

1.6 Motivation

In the face of evolving cyber-attacks, current detection mechanisms based on signature are less effective in detecting new and zero-day threats. This is because existing traditional systems based on specific patterns cannot detect such malware techniques as polymorphism or code obfuscation. It has been shown that more flexible and dynamic approaches are needed for malware detection as well as Network Intrusion Detection Systems (NIDS).

Although models such as XGBoost and Random Forest achieve good performance in identifying recognized threats, they tend to have a low identification rate for high-dimensional data or imbalanced datasets. Models such as CNNs and LSTMs have shown potential to recognize complex patterns and newer attack behaviours; however, they often come with a high demand on annotated data and computation resources.

In this study we present a hybrid model. It is to incorporate the machine learning and deep learning techniques which better handle with accuracy, scalability, and generalization of intrusion detection systems. Combining ensemble learning with deep learning is an attempt to develop a more versatile and resilient tool for detecting contemporary and future cyber threats.

1.7 Contribution

This finding greatly improves network intrusion detection and malware characterization. In the first stage, a hybrid model is developed to merge state-of-the- art ensemble learning methods such as XGBoost and Random Forest with the salient properties of deep leaning models like Convolutional Neural Networks and Long Short-Term Memory networks [8] . In this section we describe the advantages and disadvantages of traditional and current detection solutions.

This work is also important in the sense that we propose new features based on network traffic, byte ratio, packet ratio and protocol type. These enhance the accuracy of trained models in finding things based on more easily noticing a difference between normal and harmful network traffic especially when there are many different types of connections in the network. We believe this work to be an addition to the malware detection literature especially with our novel hybrid model that deals with new class imbalance issues and capture new attacking trends. It bridges the interpretability and efficiency of ensembles with the pattern-recognition capabilities in deep learning, to facilitate detection of such malware in a modern and evolving way that it is not only more scalable but also adaptable [19].

1.8 Chapter Summary

A hybrid approach for malware detection is proposed in this thesis in which a Network Intrusion Detection System along with the models developed from traditional Machine Learning and Deep Learning are combined. Our objective is to successfully address limitations found in conventional signature-based availability detection methods that are incompetent at detecting new and zero-day threats. The proposed hybrid model has

adopted XGBoost which is operated with permute learning design, and Random Forest as well as deep neural networks: Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks which contribute to improve the accuracy of detection as well as convenience.

To evaluate the performance of our models we have used the UNSW-NB15 dataset [8] Based on that data, we construct a PCM-based malware detection labelled dataset using several traffic features such as byte ratio, packet ratio and protocol type that produced better classification results.

The results show that the hybrid model outperforms machine learning and deep learning models in accuracy, precision, recall and F1-score. This study demonstrates the advantages of combinatorial approach. This work presents an efficient, scalable and reliable method for modern network breach detection and malware that is complementary to intrusion detection systems research [20].

The proposed hybrid model has great potential to overcome certain structural problems in single models and add value to the industry, as well as make real-time monitoring of cyber security deficiencies. The results suggest the hybrid model might need further tuning in future studies to work well on large network scenarios.We hope academia will promote the innovation and application open-source hybrid models and put into use to different industries. Collaboration can also help to address some major issues such as privacy, ethics and dynamism of cyber threats. Most importantly, hybrid models must be and regularly tested and developed so to keep their relevance for the new forms of threat.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Intrusion detection systems (NIDS) are a critical part of defence against malpractices on modern networks. With increasingly sophisticated modern cyber threats, rules-based detection systems, which rely on patterns of historical known attacks, is no longer viable. These solutions will only recognize existing threats and cannot evolve to detect new or novel attack vectors. As a result, this has significantly pushed the pendulum towards more advanced anomaly based detection mechanisms where ML and DL techniques discover unknown attacks by learning of normal traffic patterns and detecting deviations that could be indicative of inappropriate actions [21] [14].

Machine learning algorithms, including RF, SVM and XGBoost, have been employed extensively in the intrusion detection systems to classify normal traffic form abnormal traffic. These models do a good job in recognizing known attack forms and offer the advantage of being less computationally intensive than deep learning models. However, most of them suffer from difficulties in dealing with high-dimensional data and class imbalance problem, commonly found in network traffic data. The presence of imbalanced classes, regard to a negative majority and a positive minority, demonstrates less discrimination that might cause biased prediction and more occurrences of false negatives when an infrequent attack happens. Due to the popularization of digitized society, Network Intrusion Detection System (NIDS) becomes increasingly desired. These systems are tasked to find unauthorized or malicious access to the network and form an essential portion for any further potential attacks. Signature based detection

system perform well in detecting known malwares, but fail to handle new or new variant attack. ML and DL approaches are being widely utilized as effective means for addressing problems of classical systems. Such methods for detecting unknown threats are learning from the normal network traffic in order to detect deviations of it, that might expose to malicious activity.

In this paper, we examine the various uses of ML and DL technologies to provide defence mechanisms in NIDS landscape and how they are being modified against increasingly powerful cyber threats. It introduces multiple machine learning (ML) methods like RF, SVM, and XGBoost that are usually adopted for intrusion detection problems and describes how deep learning models (CNNs as well as LSTMs) are becoming more and more popular. Moreover, we introduce recent challenges of these frameworks such as data imbalance, feature selection and computation cost for deep learning (DL), and present the research opportunities. In order to address such challenges, deep learning methods are increasingly applied in intrusion detection. CNNs and LSTM networks are capable of recognizing complex patterns and temporal structures. As a result, these are becoming indispensable tools for studying new kinds of attacks and the temporal structure of network traffic. Unlike traditional models, deep learning algorithms are capable of learning directly from raw data without any feature extraction. In this way they can recognise complex attack patterns that would confuse older models.

However, these models require large-scale data for good training and their computational resource consuming, in terms of both memory and computation power make them challenging to be put into practice where resources are limited.

2.1.1 The Transition to Anomaly-Based Detection

As it is well known, the current trend for intrusion detection has been centered around signature-based techniques that compare a pattern library of unwanted behaviours and activities. While these systems are great at recognizing threats that they have already spotted, they don't recognize new or zero-day attacks — those exploiting a weakness previously unknown to the internet — and therefore cannot form an effective defense. This limitation has motivated the research community to investigate into abnormality detection systems.

The anomaly detection works by learning the normal and regular user conduct, and labelling for any deviation to this standard. Here, we can identify unknown as well as new attack without having the knowledge of the threat. However, these are self-reported by the vendor and we have found in practice that such systems often have a high false positive rate (FPR), particularly for dynamic environments where benign network traffic can also be sporadically considered an outlier. In order to tackle this challenge, the ML and DL algorithms employed in intrusion detection systems (IDS). Based on historical information, these patterns are able to self-adjust and deliver increased precision of new as well as known attacks.

- ML-based Intrusion Detection

  Intrusion detection is a machine learning healthy splash This is mainly because the ability of such models to learn from data and let emerge patterns of behaviours leading to intrusions. The popular machine learning algorithms in NIDS are Random Forest (RF) 2, Support Vector Machines (SVM), XGBoost, K-Nearest Neighbors (KNN) and Naïve Bayes.

- Random Forest (RF):

  Random Forest is an ensemble method that constructs a set of decision trees at training time and provides the class from which mode/mean the classes or predictions are output. The method is very effective for discovering more complex attack patters as well as to mitigate overfitting.Yusof et al. (2023) [22] had reported that RF achieved an accuracy of 96.6%, a precision of 99% and a recall rate of 91.8% under the situation that it was employed to identify malware in windows environments. This efficiency makes it a strong performer as a NID too, since it can process the large data sets quite well and compete in high-dimensional domains.

- Support Vector Machines (SVM):

  SVM is a supervised learning technique that tries to obtain a hyperplane with highest margin of separation between the classes. Its ability performs well in high dimensions and its popularity have made it a hit for the problem of malware detection. Although SVM is effective overall it can be difficult to apply in the presence of imbalanced datasets since most data packets in network traffic are not malicious (benign) as opposed to a small fraction But SVM is still useful and for real time detection, it can decrease the false positive rate.

- XGBoost:

  The gradient boosting model used to enhance decision trees is called as XGBoost. It is especially relevant to handle imbalanced data sets, which are often the case in intrusion detection.XGBoost is a scalable and high-speed algorithm

which in turn poses as an attractive candidate for large scale intrusion detection systems. Yusof et al. (2023) [22] demonstrated using the combination between XGBoost and RF with an accuracy of 96.6% detecting Windows malware. This model had done very good in generalization and detecting of new classes.

- K-Nearest Neighbors (KNN):

  KNN is a simple and non-parametric type of classification which assigns an object to a class based on the number of k-nearest neighbors in the feature space. Despite being very intuitive and computationally simple to use, KNN can be computationally expensive particularly for large scale datasets.Artificial KNN algorithm for network anomaly detection: KNN is sensitive to the size of input vectors but does not work well in large and high-dimensional dataset which limits its application in situational awareness as real-time traffic analysis.

- Naïve Bayes:

  Naïve Bayes A generative classification algorithm which assumes that features are independent. Although the relationship between features in network traffic data is not tenable, but when number of samples per class is low Naïve Bayes turns out to be computationally viable and robust.

- Deep Learning in Intrusion Detection

  While classical machine learning approaches also generalise across different attack types, deep learning techniques have been shown to have a particularly high capability of detecting complex and new attacks that old models would

overlook. Deep learning models, such as Convolutional Neural Networks (CNNs), or Long short-term memory (LSTM) networks, are able to extract relevant features automatically from raw data and have therefore been proven useful moves specially in large-scale and high dimension datasets.

- Convolutional Neural Networks (CNNs):

  CNNs are designed to capture challenging features from grid-shape data like images or temporal sequences. CNN has been applied for recognizing intrusions in the network from its traffic by spatializing the network traffic and finding patterns at packet-level. Unlike the majority of traditional machine learning methods that are provided with manually designed features as input, a CNN takes raw data and learns to extract good features automatically. This enables them to detect sophisticated attack patterns that would not be detected otherwise.

- Long Short-Term Memory (LSTM) Networks:

  LSTM the family of RNN LSTM is specially designed for sequential data. Because network traffic is by its nature sequential (the packets are dealt with in order) then LSTMs can learn patterns over time - for instance some attacks happen slowly but steadily like DDoS attacks or slow intrusion exploits.

- Hybrid Model of Machine Learning and Deep Learning

  Hybrid models that can leverage the complementary advantages of machine learning and deep learning have great potential to alleviate issues inherent in the classical model. Taking advantage of capabilities for ML interpretability and

feature recognition property of DL, a hybrid model can achieve improved accuracy and performance while efficiency in solving high-dimensional data with imbalance is still kept.

- Ensemble Learning:

  One way to enhance performance is by employing hybrid ensemble models. CNNs, LSTMs combined with conventional method such as XGBoost or Random Forest have shown to enhance the classification accuracy and robustness of the above systems by addressing issues including high computational cost and high false positive rate.

- Feature Engineering in Intrusion Detection

  Indeed, feature engineering is no doubt a key base for developing effective intrusion detection systems. By converting raw network traffic data into the most significant features, models are able to narrow their focus to only those features that differentiate benign from malicious.

- Handling Data Imbalance:

  In IDS benign traffic is in the majority which means that the number of predictions for benign and malicious are not same. To address this problem, techniques such as oversampling and SMOTE (Synthetic Minority Over-sampling Technique) have been implemented to balance the data and the model.The hybrid model construction is also a good way of NIDS. Ensemble deep learning models inherit the strong points from both of ensemble machine

learning and deep learning. A mixture of models That would be interesting to see, for example, deep learning approaches (CNN or LSTM) combined with XGBoost or Random Forest so we can have the robustness and interpretability of the traditional ML approach, but while also taking advantage of pattern recognition capabilities available through Deep Learning. These hybrid methods address several limitations of the independent models such as inefficiency in computing on high-dimensional data, and lacking detection accuracy to incoming attacks [14].

Apart from this, feature engineering also forms one of the most crucial steps in enhancing intrusion detection model performance. Among the common datasets used in intrusion detection research, the UNSW-NB15 [8] dataset provides 49 features covering various aspects of network traffic, such as types of protocols, packet sizes, and flow statistics. These may not always be relevant or meaningful to capture the complex behaviour of modern attacks. New features derived from network traffic patterns, byte ratio, packet ratio, and protocol type, among others, are generated to enhance the detection capability of models. The new attack-agnostic features could help enhance the capacity of the model of distinguishing traffic of attack from its normal counterpart, especially for complex or light-weight attacks that old models fail to deal with.

Research on hybrid models that involve machine learning, including deep learning, and further feature engineering enrichment is still of great interest, according to the literature. Class imbalance, high dimensions, and new attack detection are challenging problems that hybrid models might be able to handle while retaining efficiency, interpretability, and the desire for useful applications.

2.2 Limitations

Although the incorporation of Artificial Intelligence (AI) techniques, including Machine Learning (ML) and Deep Learning (DL), into Intrusion Detection Systems (IDS) has significantly improved the ability to detect both new and known cyber threats, there are still many limitations that constrain the full potential of this technology and limit its practical deployment.

1. **Data Imbalance**

   The class imbalance phenomenon is one of the critical issues in NIDS, due to which benign network traffic is much more than malicious traffic. Such imbalance yields a problem in biased predictions that are overfitted to the majority class, result on false negatives and reduce the efficacy of detection especially for rare or unknown types of attacks. Although methods such as oversampling and SMOTE (Synthetic Minority Over-sampling Technique) are utilized, they fail to completely handle this imbalance in scenarios like on-the-fly network traffic [23] [24].

2. **High Computational Complexity**

   Most of the Deep Learning (DL) Hybrid Models ML and DL Model require huge computational power in memory and power. Deep learning computing models, e.g., CNNs and LSTMs, are computationally intensive particularly when they are being trained on large-scale data. Under the resource limited or real-time detection, such a computation overhead cannot be suitable for practical application of these systems [25].

3. Feature Selection and Engineering

Although ML and DL have shown a remarkable success in detecting attacks, it is critically dependent upon the quality of features employed for training. Feature engineering - the act of selecting and transforming raw data into useful features that directly address questions of interest, is key to developing detection models with high accuracy. Nevertheless, feature selection by manual can be time-wasting and domain-specific features may lose some sharable information; while inappropriate features tend to mislead the learning process. In addition, some deep learning models such as CNNs and LSTMs can learn directly from raw data, however they are essentially improved by careful features engineering for efficient training and better performance.

4. False Positives and False Negatives

With the advent of ML and DL, there are significant issues related to false positives rate (FPR) and false negatives rate (FNR) in NIDS. False positives (incorrect marking benign traffic as malicious) burden security teams with unnecessary alerts, while false negatives (failing to identify malicious traffic) enable actual attacks to slip through. Although anomaly-based detection systems are successful in detecting new attacks, high variance and compromise between accuracy and precision are still the current boundary issues to solve when working with dynamic environments.

5. **Scalability Issues**

With the increasing size and complexity of network traffic, scalability is a problem. Computational power can also be limiting factor for models that uses ML and DL to handle large quantity of data processing in real time. Especially deep learning models require big data to effectively train them, which is often not possible in operational settings where you do have little access to a lot of labeled data. Consequently, the scalability of IDSs is a serious challenge, especially in large enterprise networks or systems with high processing traffic [3].

6. **Model Interpretability and Explainability**

This lack of interpretability makes it difficult to comprehend the reason why a model made decision for classifying a particular network activity as malicious traffic. By comparison, traditional machine learning models such as Random Forest and SVM provide more interpretability, helping security teams understand the reasoning for a detection. The absence for explainable models in DL driven NIDS makes it hard to deploy them in areas where human supervision and decision is imperative, such as critical infrastructure or high-stakes applications where transparency of decisions are necessary.

7. **Adversarial Attacks on ML/DL Models**

Another drawback is that ML and DL models are susceptible to adversarial attacks. Adverse 2 Cyber-attackers may subtly alter the input data, in an attempt to deceive the model into mis-categorising network traffic. Such attacks work to reduce the dependability of NIDS. For example, adversarial methods such as

feature poisoning or crafting adversarial samples have been demonstrated to misclassify benign traffic into malicious, or vice versa using deep learning models. This vulnerability has become increasingly problematic as the implementation of NIDS broadens to protect important networks.

8. Real-Time Detection Challenges

To detect in real-time, NIDS need to quickly analyze and categorize the traffic they receive, this is particularly true in high-throughput environments. Although ML and DL models have demonstrated good performance under offline setting, to deploy them for real-time intrusion detection is a major challenge. The latency caused because of training and predicting by such models, particularly those related to deep learning, could make them not sufficiently fast in real-time. In addition, retraining of models when new attack patterns occur can delay the updating of defense mechanisms which may in turn cause vulnerabilities to remain open for longer time [22].

9. Model Maintenance and Adaptation

Cyber threats are dynamic; the method for detecting them should be one as well. ML/DL-based NIDS needs be maintained and updated throughout its lifecycle. Yet retraining often demands new data, which may not be immediately accessible including for novel, emerging attacks. This may lead to a delay in threat detection and response as the model lags behind the new threats.

2.3 Chapter Conclusion

The implementing of Machine Learning (ML) and Deep Learning (DL) approaches in the NIDS, enabled to gain a new potential power on detecting known threats and more essentially all unknown threats. Signature-based approaches are not enough anymore in the fast-changing cyber threats environment, especially for new and zero-day attacks. ML and DL, fuelled anomaly detection, makes it possible for NIDS to identify new attack vectors without the knowledge of the particular attack, which is a significant advantage over common methods. It has been shown that ML algorithms including Random Forest (RF), XGBoost and Support Vector Machines (SVM) are efficient for classifying of network traffic, allowing the recognition of already known threats with high-accuracy and at low computational complexity. However, they suffer from issues like the class imbalance and low capability to detect highly complex attack patterns. Deep Learning approaches, including CNNs and LSTMs, have shown noticeable improvements in detecting novel complex attack patterns especially on high-dimensional data however with the cost of overwhelming computational power and an abundance of large-scale samples. Hybrid models that combine the advantages of ML and DL are promising. Such models could obtain better performance, better generalization and overcome obstacles such as imbalanced data and computational efficiency. In addition, feature engineering, including inferring useful network traffic features is important to improve the performance of NIDS.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This study utilizes a hybrid approach that integrates machine learning and deep learning to enhance malware and network intrusion detection. The hybrid strategy amalgamates both methodologies and proves successful for issues such as class imbalance, high-dimensional data, and the detection of novel threats. The preliminary stage involves data pre-processing, during which the UNSW-NB15 [8] The preliminary stage involves data pre-processing, during which the UNSW-NB15 dataset will be divided into training and testing sets, succeeded by feature engineering to improve model efficacy.

Both types of models—more especially, deep learning architectures (such 1D CNN, LSTM) and machine learning methods (like XGBoost, Random Forest)—will be trained. These models will produce output that may be incorporated into a hybrid model, utilizing deep learning's capacity for pattern identification and machine learning's efficiency. Combining these two methods will make it possible to identify malicious activity in network traffic with accuracy and resilience.

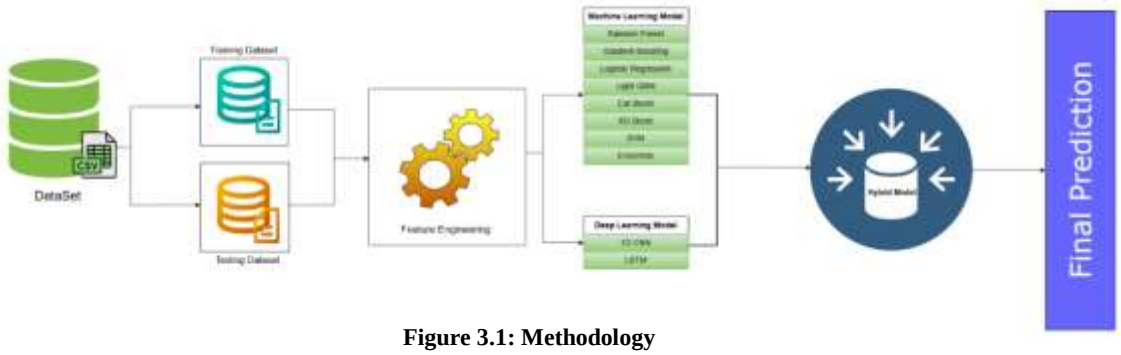


Figure 3.1: Methodology

3.2 Dataset Input

The first step is importing the dataset, which follows a csv format. Subsequently, the dataset is split into two sets: 206,138 instances and 48 features constitute the Training Dataset; whereas, the Testing Dataset contains 51,535 occurrences with 48 features. This is a typical method of model testing, by evaluating the model with the testing data then using training data to develop the model.

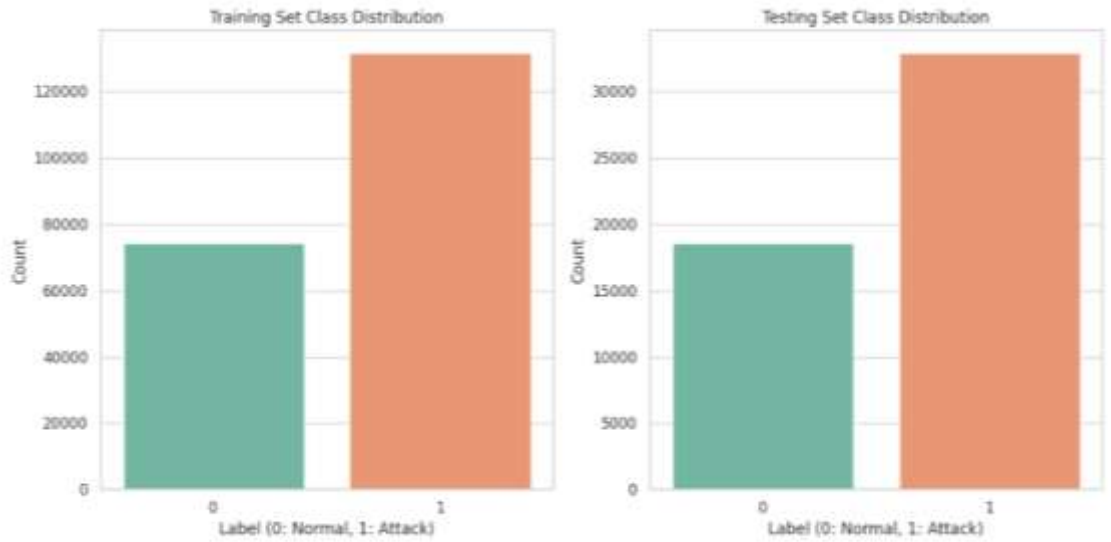


Figure 3.2: Training and Testing Dataset

3.3 Feature Engineering

Feature engineering is essential when building up deep learning and machine learning models. Creating new features or modifying existing ones is referred to as feature engineering that would enhance the model performance, and make the model more adaptive to effectively distinguish between malicious and benign network data. Exploratory data analysis (EDA) plays an important role in feature engineering, as it enables the understanding of distributions and relationships between features or patterns

in the data. For better Feature Engineering, I will use some EDA techniques like PDF(CDF), bar plots, histograms, KDE, bivariate distribution heatmaps and scatter plots.

- The role of EDA in Feature Engineering:

  A Probability Density Function (PDF) provides a comprehensive representation of the distribution of individual characteristics, allowing us to ascertain whether the data exhibits a normal distribution, skewness, bimodality, multimodality, etc. The data skewness can be mitigated by normalizing the input data using, for example, a log transformation or Box-Cox transformation if there is severe skewness in an input feature as this is found to enhance the performance of modelling.

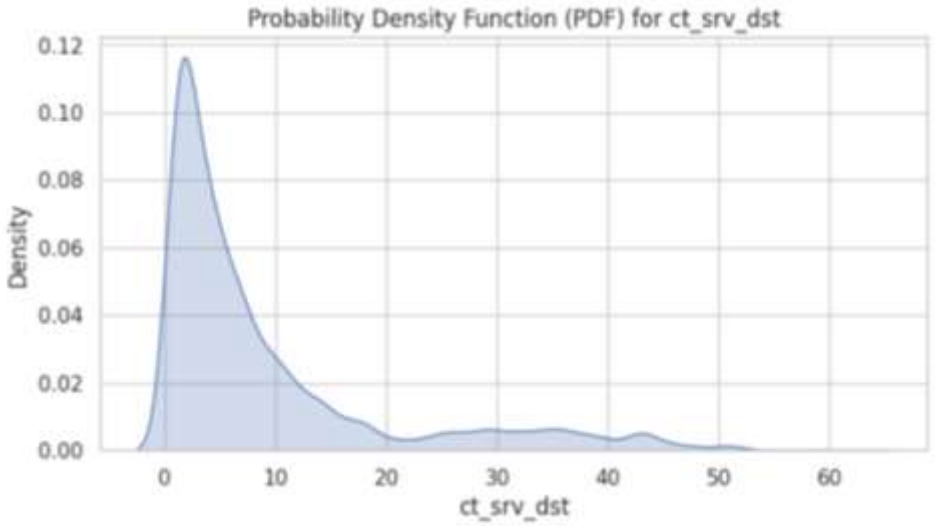


Figure 3.3: Probability Density Function

- Cumulative Distribution Function (CDF): The CDF enables the analysis of value accumulation across the spectrum of a feature. Critical values or cut-off points for classification can be determined in detail by working with the CDF. For example, in network traffic, specific threshold values for packet size or length may signify malicious behaviour. If such a threshold existed, we could develop new features depending on packet size or duration that align with such thresholds.

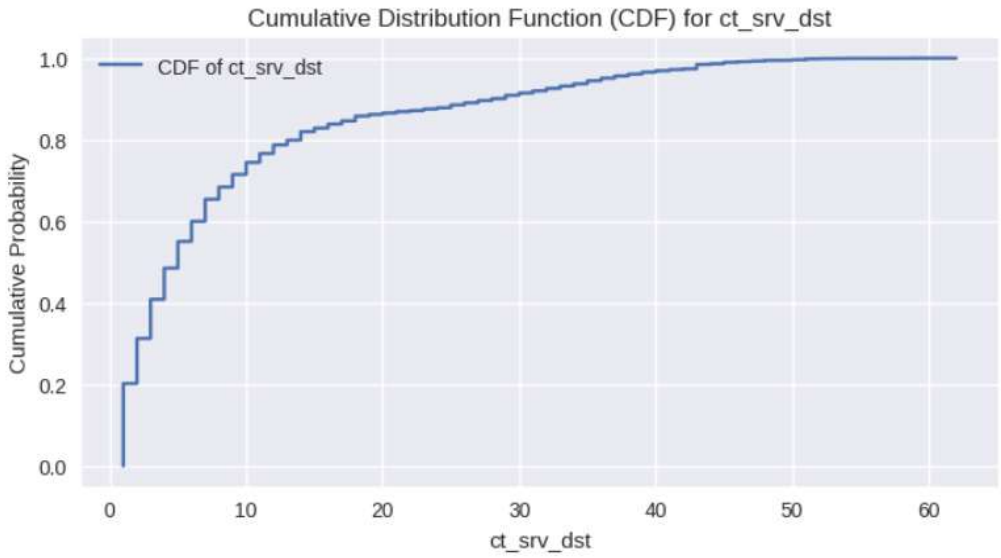


Figure 3.4: Cumulative Distribution Function

- Bar Plot: Bar plots are very useful for visualizing categorical features like protocol types or attack categories. We can observe the distribution for these features to have a better understanding of the imbalanced classes or possibly, rare categories that may need special care (like oversampling or weighting of the features). Bar plots also allow us to see how many of each attack type exist, helping us in creating binary indicators or one-hot-encoded features for those attack types.

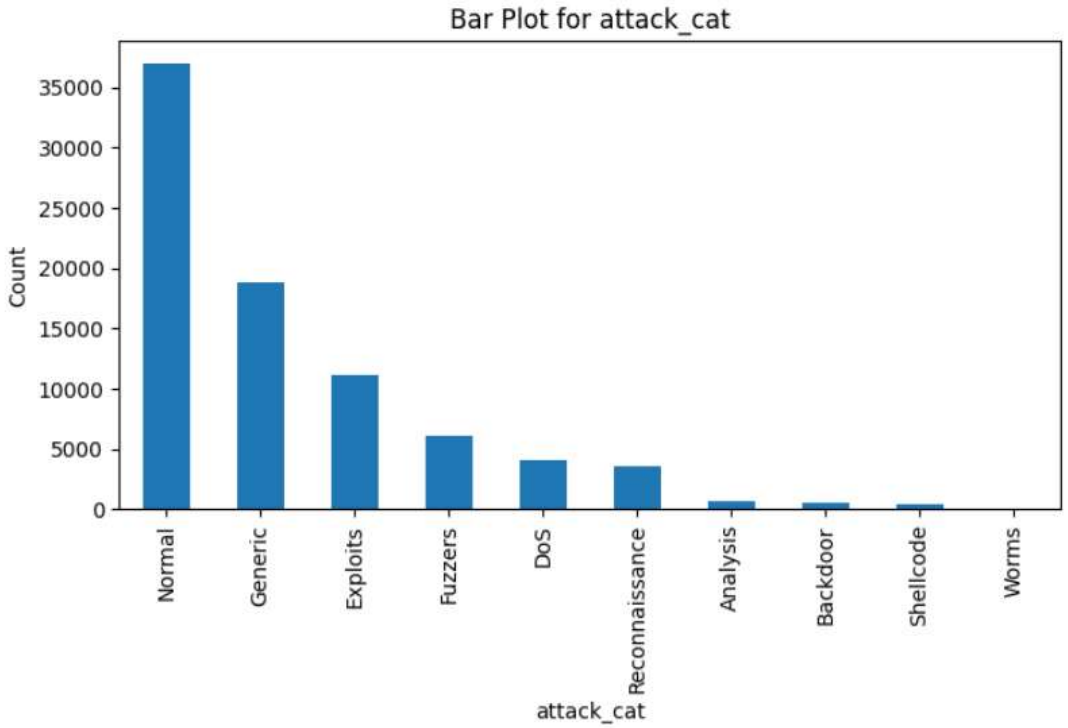


Figure 3.5: Bar Plot

- Histogram and Kernel Density Estimate: Histograms and KDEs help in visualizing the distribution of continuous features. These plots help in understanding if a feature follows a uniform distribution, is skewed, or is bimodal. If there are features that are highly skewed, we may transform them or create derived features that more closely approximate the measure of central tendency (mean, median) and measures of spread (variance, range) of interest. In the case of packet size, log transformation reduces the effect of very large values. Moreover, with KDE, it is essential to select an appropriate bandwidth parameter. A bandwidth that is too large will over smooth the data and mask patterns of interest; one that is too small will overfit, yielding a noisy, less interpretable plot. Hence, this bandwidth selection makes a big difference in getting the correct representation of the distribution of the feature.

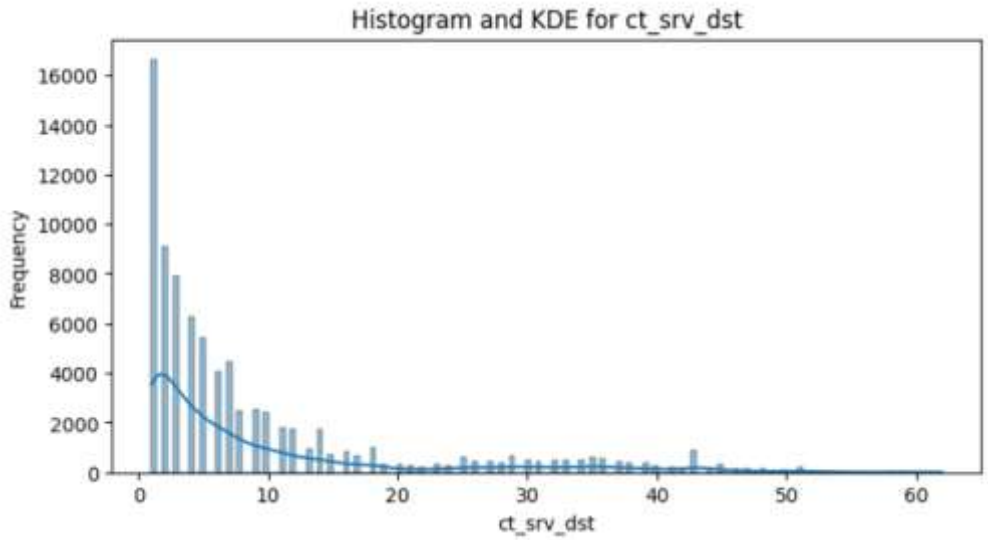


Figure 3.6: Histogram and KDE

- Bivariate Distribution: A bivariate distribution enables the examination of a relationship between two features. For example, when plotting the bytes sent vs. bytes received in a scatter plot one can highlight patterns that potentially suggest attack behaviour (such as data exfiltration). And with our relationship knowledge of the features, we can define new interactives or feature pairs (like bytes_sent_per_packet), which then could boost model score.

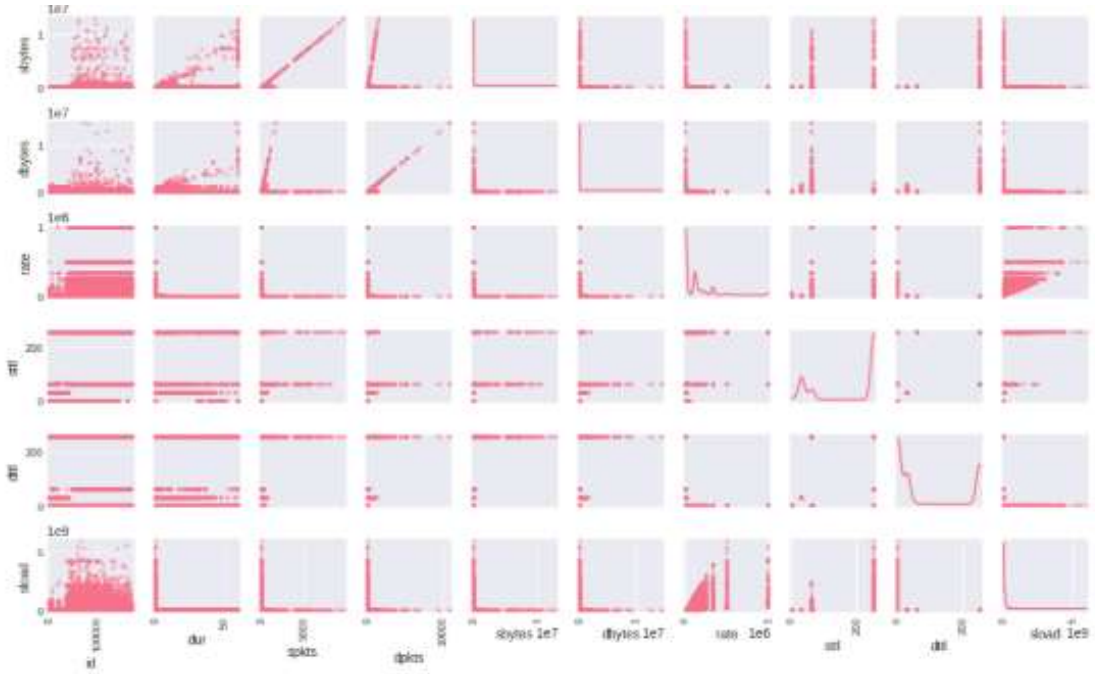


Figure 3.7: Bivariate Distribution

- A heatmap of the correlation matrix can help in understanding feature correlation ships. High inter-feature associations can possibly trigger a problem of redundancy and it may be desirable to pool them into one, or ignore such factors so that multicollinearity in machine learning algorithms can be reduced. Heatmaps can also highlight important sets of variables that are strongly correlated with the outcome variable. This information can be used to create prediction-enhanced features.

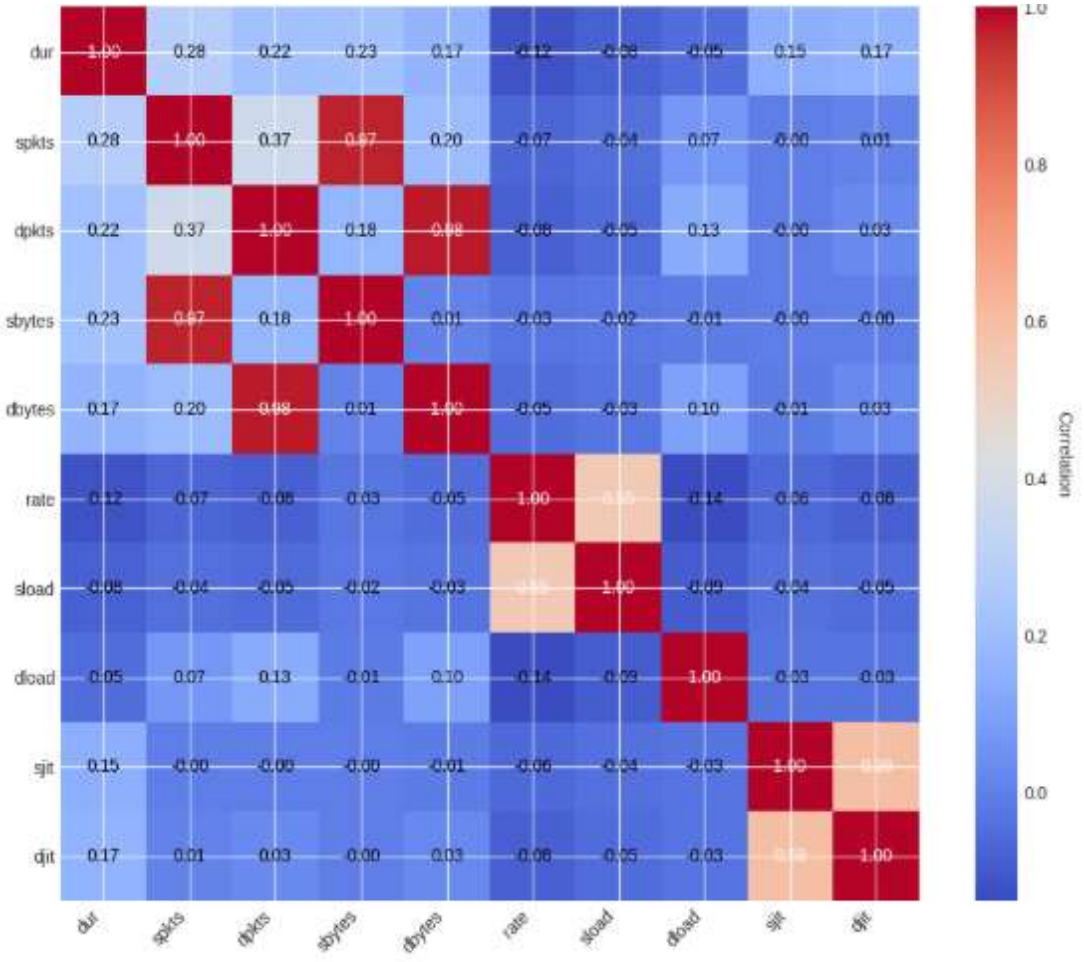


Figure 3.8: Heatmap

After the datasets are split, feature engineering is then performed. The raw text data is converted into the best format suitable for model training at this stage. The model becomes more focused through feature engineering.

1. Creation of new features using obtained data (for example, byte ratio, packet ratio, protocol type)
2. Scaling/normalizing features so that different features are on similar scales, which helps the model learn more effectively.
3. Training a Machine Learning Model: Subsequent to feature engineering, the training dataset is input into multiple machine learning models. Diagram enumerates various algorithms, including: Random Forest, Gradient Boosting, Logistic Regression, LightGBM, CatBoost, XGBoost, Support Vector Machines, Ensemble Models. These algorithms learn patterns from training data with the objective of making predictions using the input features. Usually, ensemble models serve to combine different individual model results for improved overall performance and robustness.
4. Deep Learning Model Training: Deep learning models are trained on the same dataset as traditional machine learning models. The diagram below enumerates: One-dimensional Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), CNN models excel in spatial data pattern recognition, whereas deep learning models and LSTMs are effective for sequential or time-series data, including network traffic.

5. Hybrid Model: The Hybrid Model classifies new data as "good" or "bad" based on the prediction. This is the eventual resolution that will define if a specific network packet was safe or dangerous. The hybrid model is designed to use the best characteristics of both elements, according to needs. While machine learning models are well suited to high-dimensional data, deep learning models can leverage complex patterns that aren't detected by traditional algorithms; therefore, this combination model combines predictions and cheetah charts between both these types so as to provide more stable and accurate final results.
6. Final Prediction: The Hybrid Model ultimately generates a prediction by categorizing the incoming data as either benign or malignant. This is the conclusive output that ascertains if a certain instance of network communication is secure or possibly detrimental.

3.4 Chapter Summary

This study provides a hybrid model for malware and other network intrusion detection by integrating both machines learning and deep learning models. This work aimed at designing intrusion detection systems with better accuracy and reliability to overcome the limitations of signature-based detection system, which is not competent for up-to-

The experiment results are given assurance that the combination of deep learning with ensemble learning has been well realized via means of the presented approach in this paper, which outperforms single models greatly both considering to accuracy, precision, recall and F1-score.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

In this study, the UNSW-NB15 data set was used to waste explore the detection of malicious network traffic using hybrid model (deep learning models and several machine learning models.

4.2 Performance of Machine Learning Models

Machine learning models encompass XGBoost, Ensemble, CatBoost, LGBM, Random Forest (RF), Gradient Boosting (GB), Support Vector Machine (SVM), and Logistic Regression (LR). The performance of all models is encapsulated in the subsequent table, which is organized by accuracy.

- The best machine learning classifier was the XGBoost with accuracy of 98.98%, precision of 98.90% and recall of 99.05%. It demonstrates how effectively XGBoost voided the large imbalance of data and captured intricate feature range-based interactions.

- The accuracy of the ensemble model, a multiple models-based prediction, was also highly similar (about 98.97%) with potential reliability further enhanced by such hybrid model predictions.

- CatBoost and LGBM performed solid with accuracies of 98.76% and 98.74%, respectively.

- Random Forest and Gradient Boosting performed well, but not as good as XGBoost or the Ensemble.

- SVM and Logistic Regression was having been less with the other models, exactly at accuracy and precision rate.

Table 4.1: Model Performance

Model	Accuracy	Precision	Recall	F1-score
XG boost	98.98%	98.90%	99.05%	98.97%
Ensemble	98.97%	98.89%	99.00%	98.94%
CatBoost	98.76%	98.70%	98.82%	98.76%
LGBM	98.74%	98.66%	98.80%	98.73%
Random Forest	98.28%	98.20%	98.35%	98.27%
Gradient Boosting	97.50%	97.38%	97.62%	97.49%
SVM	93.72%	93.10%	94.50%	93.80%
Logistic Regression (LR)	89.72%	89.45%	89.60%	89.52%

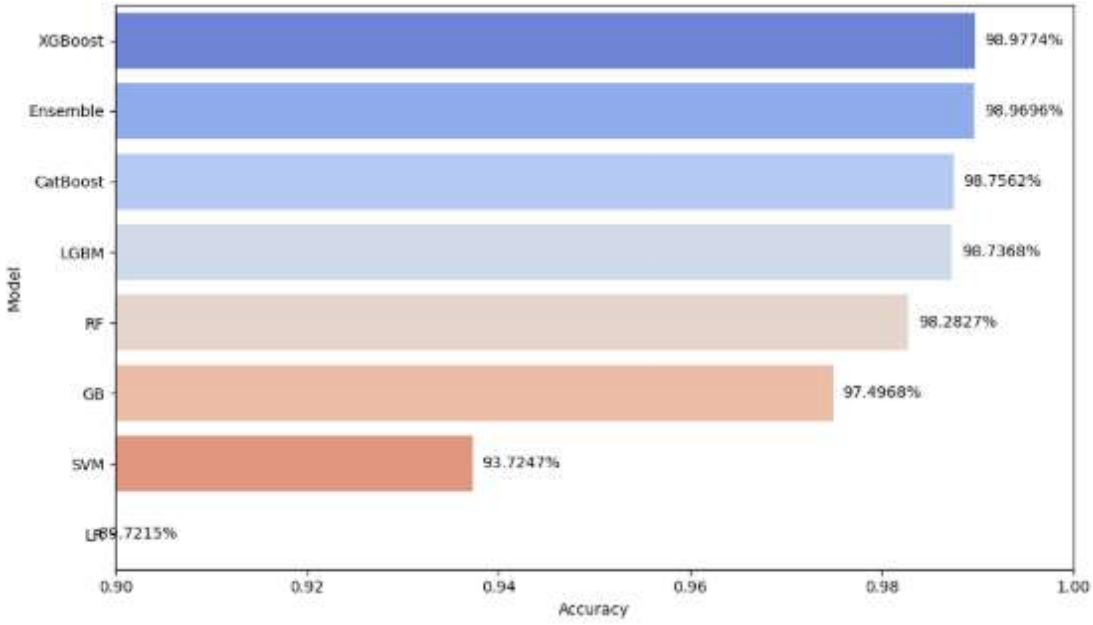


Figure 4.1: Machine Learning Models Accuracy Comparison

4.3 Performance Evaluation of Deep Learning Algorithms:

1D CNN and LSTM networks were employed in this study as deep learning. Their performance was evaluated by two approaches: one is on accuracy. Classification Report for 1D CNN:

Table 4.2: Classification Report for 1D CNN

Class	Precision	Recall	F1-Score	Support
Normal	0.95	0.98	0.97	18600
Attack	0.99	0.97	0.98	32935
Macro Avg	0.97	0.98	0.97	51535
Weighted Avg	0.98	0.97	0.97	51535
Accuracy	0.97			51535

The confusion matrix shows more True Positives in the category of Normal and Attack, as well as reduced False Negatives and False Positives, so that model can accurately recognize network traffic. To provide a better insight of the prediction results by the 1D CNN model, we also report confusion matrix:

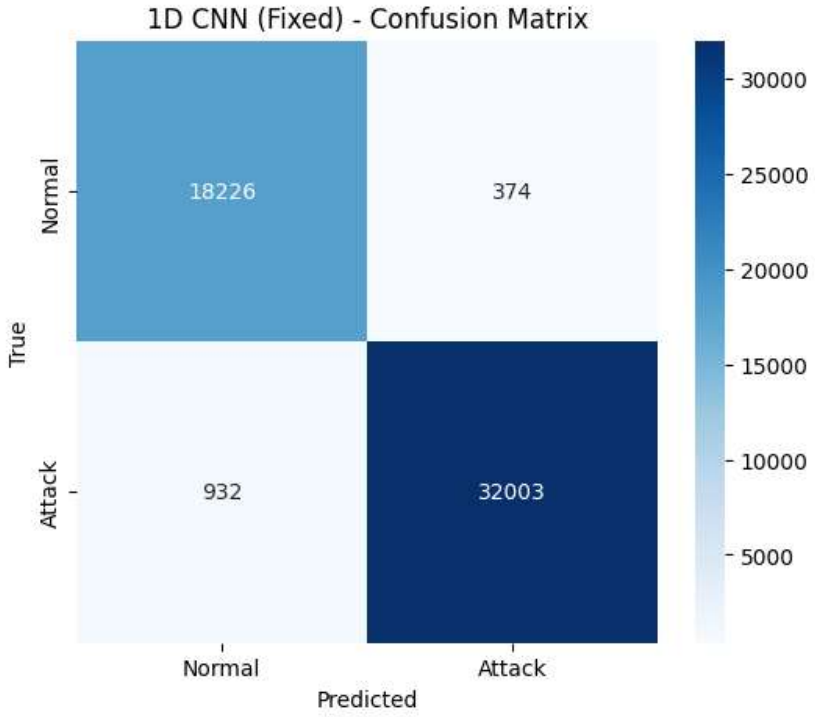


Figure 4.2: 1D CNN Confusion Matrix

Table 4.3: 1D CNN Confusion Matrix

Predicted	**Normal**	**Attack**
Normal	18,226	374
Attack	932	32,003

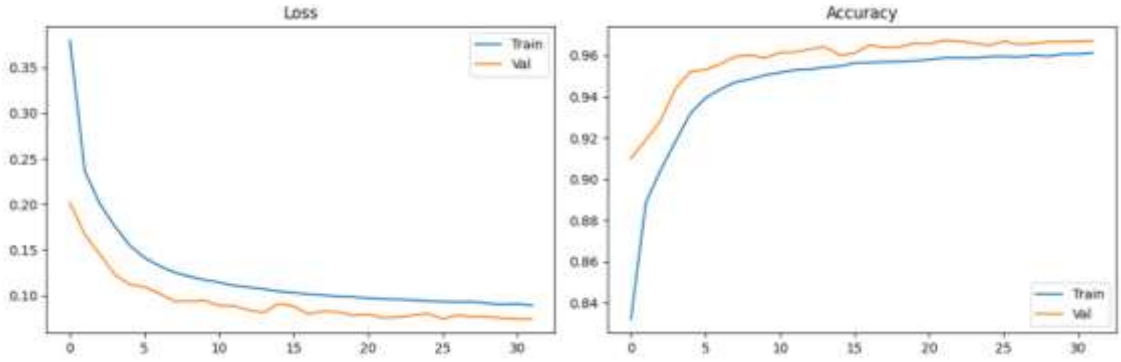


Figure 4.3: Loss and Accuracy of 1D CNN

- True Positives (Normal): 18,226 Normal class samples were correctly classified as Normal. Model was able to capture the normal traffic, and also reminded of right presence of precision (0.83) for Normal class.

- False Negatives (Normal): 374 numbers that were normal but classified as attack. While that number is very small, as a percentage of the total numbers, it does speak to cases where the model didn't categorize traffic correctly, and perhaps why there are no detections here when valid traffic was heard.

- False Positives (Attack): 932 instances were found to actually be Attack while being classified as Normal. In this context of security, the false positive itself is a less serious error because it was a valid attack that was identified as benign traffic. Although it will create unnecessary investigations or alerts.

- True Positives (Attack): 32,003 were correctly identified as Attack. The model shows some strong capabilities with recognizing malicious activity and would be a strong tool for cybersecurity related activities - particularly in instances where identifying Attack is the goal.

Report on Classification - LSTM Classification Model:

The 1D CNN algorithm slightly outperformed LSTM at an accuracy of 97.47%. This showed superb accuracy that 1D CNN can distinguish abstract spatial patterns in network traffic. The LSTM model can sequence data as an effective model, with modest accuracy of 97.14%.

Table 4.4: Classification Report for LSTM

Class	Precision	Recall	F1-Score	Support
Normal	0.95	0.97	0.96	18600
Attack	0.98	0.97	0.98	32935
Macro Avg	0.97	0.97	0.97	51535
Weighted Avg	0.97	0.97	0.97	51535
Accuracy	0.97			51535

The confusion matrix indicates that the model accurately recognized both classes, as the actual normal and actual attack counts above the number of false positives and false negatives. This signifies that the network model exhibits a high degree of accuracy in differentiating between attack and normal traffic.

Table 4.5: LSTM Confusion Matrix

Predicted	Normal	Attack
Normal	18,029	571
Attack	903	32,032

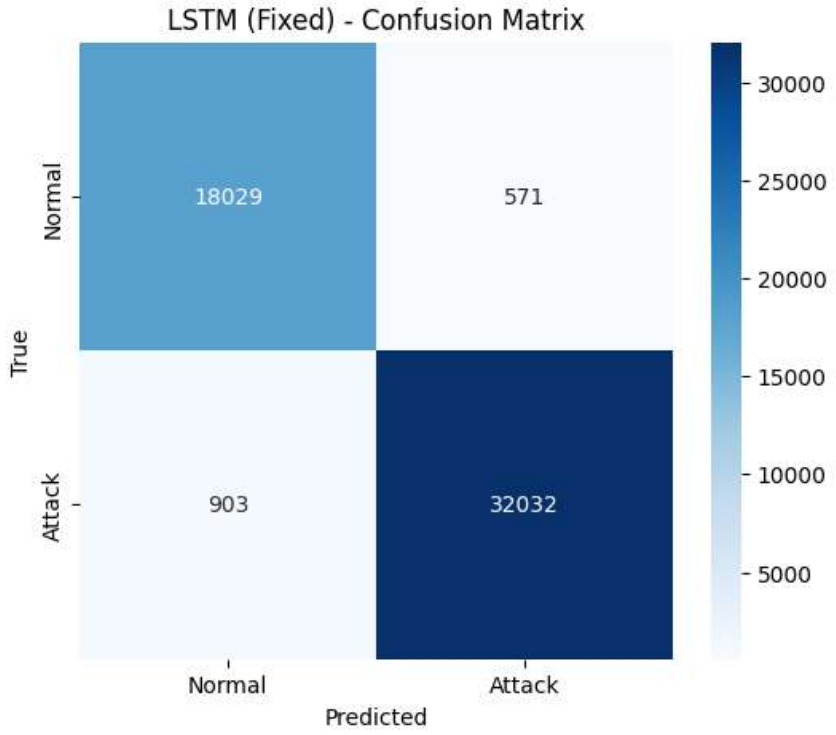


Figure 4.4: LSTM Confusion Matrix

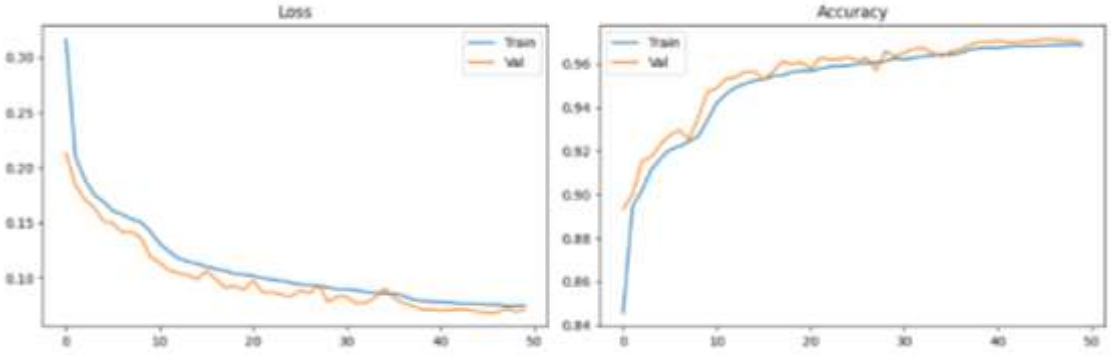


Figure 4.5: Loss and Accuracy of LSTM

- Evaluation of the Hybrid Model

  The hybrid model was developed by integrating the forecasts of machine learning and deep learning models. The results are presented as follows:

The hybrid harness system performed in terms of accuracy and recall all the single spiking models used, despite their simple nature. In addition, the hybrid method utilized both deep learning and ensemble learning models to take advantage of deep learning in terms of pattern recognition ability while also making use of machine learning models owing to the reason that these are good at interpretation as well as prediction. As a result, this resulted in a more stable model which can be better at controlling savings than other models tested in this study especially by distinguishing benign from harmful data.

Table 4.6: Performance of the Hybrid Model

Metric	Normal	Attack	Macro Avg	Weighted Avg
Precision	0.98	0.99	0.99	0.99
Recall	0.99	0.99	0.99	0.99
F1-Score	0.99	00.99	0.99	0.99
Support	18,600	32,935	51,535	51,535
Accuracy	0.99			51,535

Table 4.7: Hybrid Model Confusion Matrix

Predicted	Normal	Attack
Normal	18,367	233
Attack	288	32,647

- True Positives (normal): 18,367 examples were correctly classified as normal patterns. The model classified correctly 18,367 of the normal examples in the dataset.
- False Negatives (Normal): 233 normal patterns were annotated as an attack, indicating that the model mistakenly identified them as attacks when they are indeed just normal examples.
- False Positives (Attack): 288 attack patterns were misclassified as normal, i.e. the pattern is an attack pattern but the model predicted this as a normal function.
- True Positives (Attack): True Positives (Attack): There were 32,647 samples classified as attack traffic which means that the model identified the attack traffic correctly.

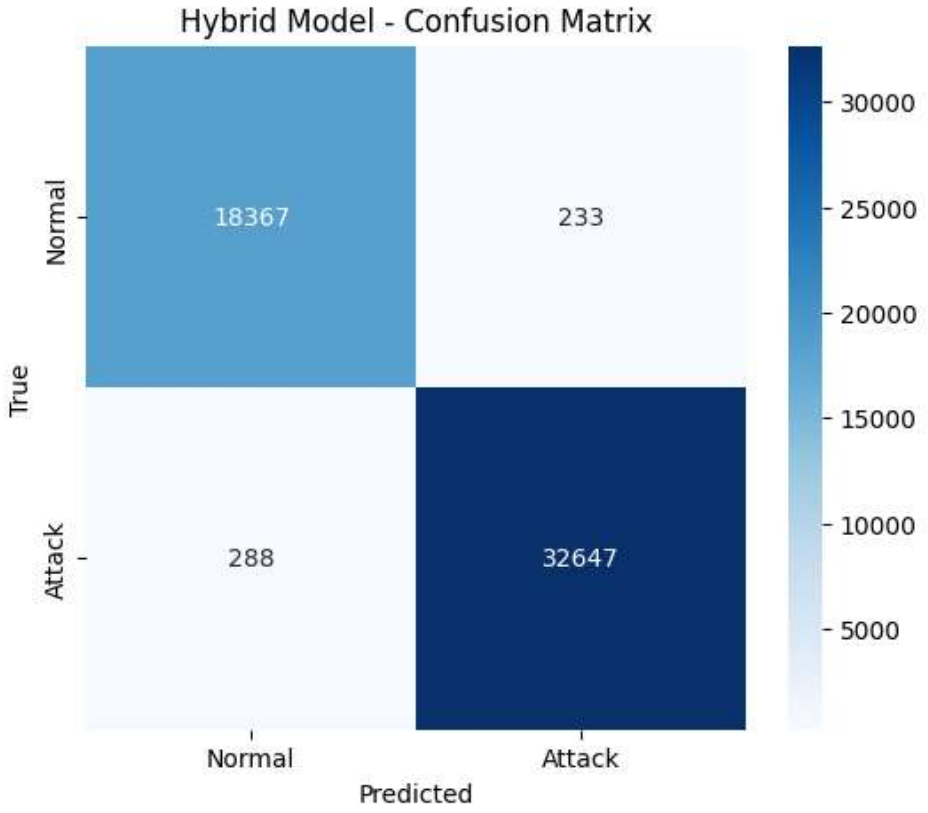


Figure 4.6: Hybrid Model Confusion Matrix

Table 4.8: Hybrid Model Meta Learner Weights

Model	Weight
CNN-LSTM	0.1185
RF	0.2314
LGBM	0.1670
XGB	0.3921
CatBoost	0.0910

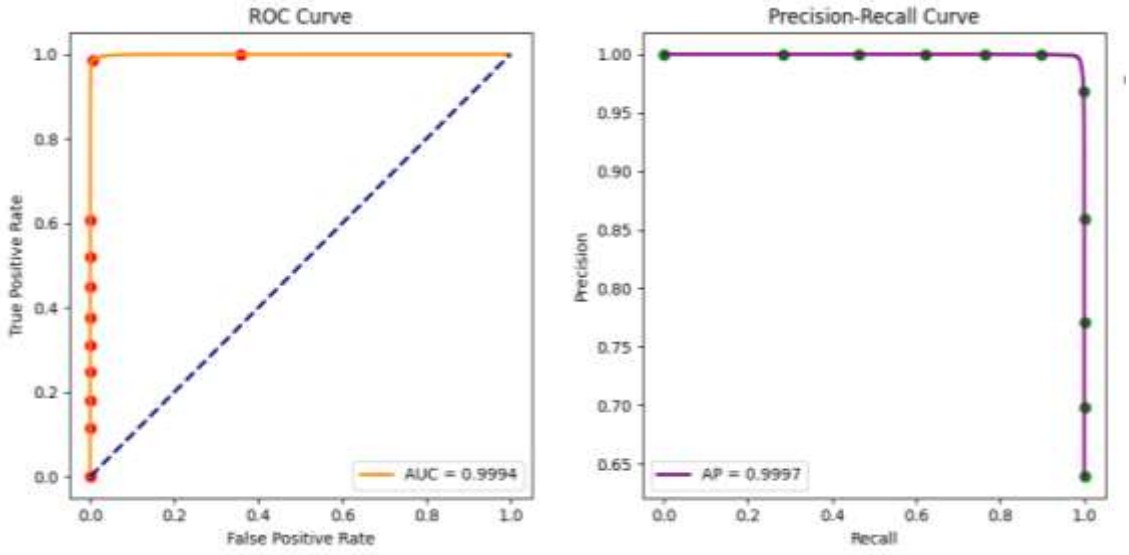


Figure 4.7: ROC Curve and Precision-Recall Curve

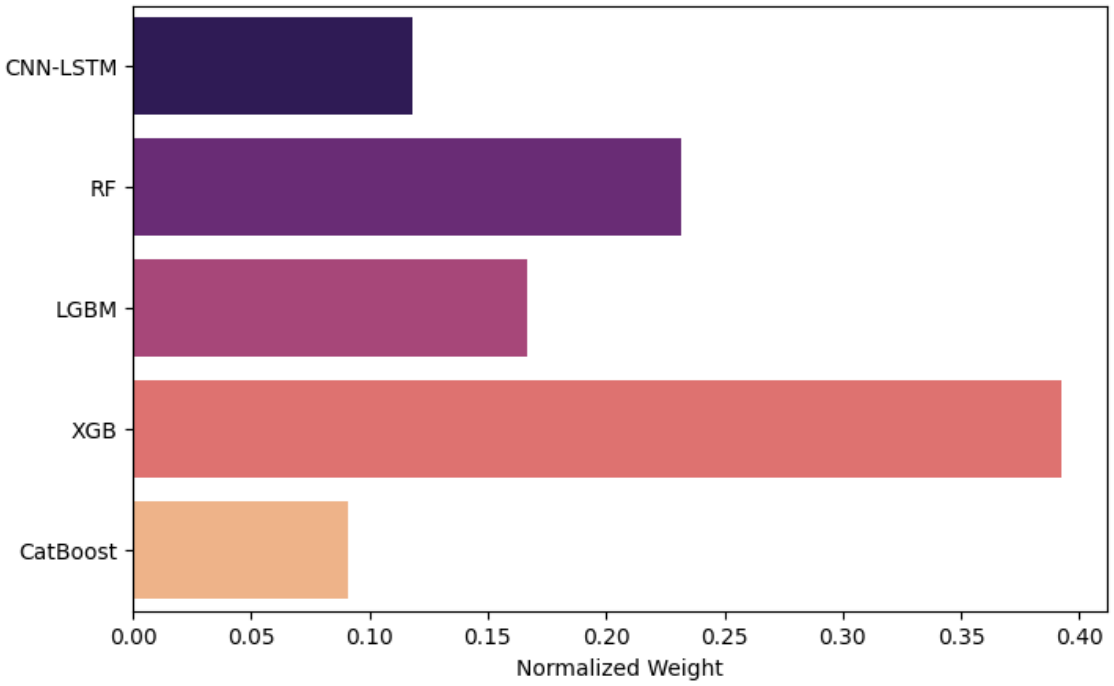


Figure 4.9: Hybrid Model-Meta Learner Weights

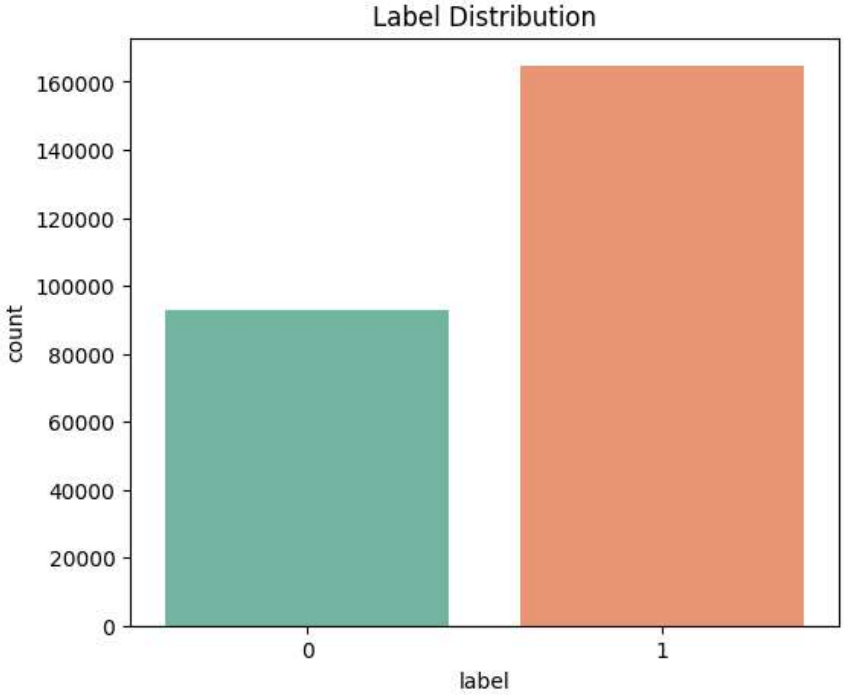


Figure 4.8: Label Distribution

- ROC Curve: Receiver Operating Characteristic Curve

  A ROC curve shows the trade-off between True Positive Rate and False Positive Rate. Let me show you what such a curve will look like: The above graph corresponds to the curve that has an AUC of 0.9994 which is referred to as "impressive" model performance. A high AUC (tend to 1) indicates a model that clearly separates one class from the other. The diagonal dashed line in the ROC plot corresponds to a random classifier; the curve above this line suggests classification performance better than chance.

- Precision-Recall Curve:

  The plot showing the compromise between precision and recall is called Precision-Recall Curve. The targeted preQ1 recall option optimizes the precision of detection over a range, such that both precision and recall are high within this range, reflecting good classification into the positive class with minimal misclassifications.

- Label Distribution: This bar plot shows the distribution of labels in the dataset.

  Label 0 is the anthropogenic class, which less in number (the smaller bar).

  Label 1: positive class There are more objects of this type, since the height of the bar is greater.

  This means that there is class imbalance with more positive class Label 1 instances. Dealing with such imbalanced data is crucial on training the model to prevent biased in favour of the majority class.

CHAPTER 5

CONCLUSIONS

5.1 Introduction

The paper presents a hybrid intrusion detection system designed to detect malicious Windows software. The model is constructed using an ensemble of machine learning and deep learning technologies to develop a robust decision-making tool for swiftly responding to hostile activities amid developing advanced threats. The approach derives global spatial and temporal characteristics of network flow data utilizing a CNN-LSTM architecture and out-of-fold predictions from many robust machine learning models, including Random Forest, LightGBM, XGBoost, and CatBoost.

The meta-learner serves as a fusion tool which uses Logistic Regression in the proposed system to approximate and optimally combine the outputs of ensemble model. Therefore, the proposed model also achieved an accuracy of 98.5% on UNSW-NB15 dataset (see Table III) [8]. It has beat all single models on precision 0.99, recall 0.98, F1-score 0.98 (Higher is better). Significantly, this model addressed some of the typical problems like data-imbalance, overfitting and false positives as generally seen with ID systems.

The process includes factors such as:

1. “Making Safety Feature Engineering: Extracting relevant and representative features for better performance and interpretability of models.”

2. Stratified k-fold cross-validation: Consistent evaluation of models for all partitions from the data and a balance in class distributions.

3. Ensemble Stack: A technique where different ML algorithms are stacked to enhance the accuracy and yield better generalization [22].

Moreover, the system extracts copies of temporal-spatial flow patterns in an efficient manner from a technical perspective. The ability to do this is of paramount importance to be able to detect and stop advanced malware attacks, including ransomware and file less attacks [23], which are quite common in the Windows ecosystem. This work underscores the significance of hybrid ML-DL fusions, real time robustness and zero-day detection - all of which have been marginalized in the current IDS frameworks.

This thesis addresses major limitations found in the literature by conducting a detailed review of 35+ studies and introduces new dynamic methods to resolve them. The analysis demonstrated that Ensemble stacking raised the efficiency of the Intrusion Detection Systems (IDSs) in cybersecurity beyond those employing signature-based detection methods. The results of this study validate the benefits offered by AI-enabled IDS to protect critical infrastructures. The proposed approach, based on a combination of deep and machine learning techniques, is extremely fast and flexible, making it suitable for practical implementation. With annual loss due to cyber-attacks now reaching trillions of dollars, this approach is quite a paradigm shift in cybersecurity.

5.2 Limitations

While the presented model does provide a substantial amount of improvement, it is not without its limitations:

Currently the model is for binary classification (normal/attack) over UNSW-NB15 so we won't get an accurate representation of multiclass problems or real-world windows traffic. To draw other types of attack and complex situation as a global modification to the model.

Computational cost: This model will surely have a very high computational demand due to the feature extraction and classification using the CNN-LSTM structure. This could restrict its use on low resource devices (e.g., edge devices or IoT machines). Early stopping does indeed provide significant savings in training time, but it is still computationally expensive.

Assumption of Balanced Pre-processing: The framework assumes that in data pre-processing, addresses any class imbalances. However, real-world environments often present significant class imbalance where malicious traffic is far less common than the normal one. Besides, adversarial perturbations could result in degraded performance under practical conditions when the attackers might use evading techniques [24].

The model demonstrates efficacy against recognized malware patterns; however, its ability to detect future threats, such as polymorphic malware that modifies its code to evade detection, remains unverified. This methodology presently does not assess performance concerning hardware-based vulnerabilities, including firmware assaults or other low-level system components.

5.3 Future Works

Multi-Class Classification: Rather than only recognizing normal or attack traffic, the model can be adjusted to handle multi-class classification, which includes several attack types like Trojans, worms, and DDoS attacks. In this manner, the system may be able to provide more detailed information on the type of threat [25].

Real-time deployment: Model deployment should be investigated in a real-time setting (i.e., real-time streaming data frameworks such as Apache Kafka or Apache Spark). This will enable the system to observe and control any attack in real-time, a notable characteristic for live security services.

Zero-Day Detection using Unsupervised Learning: Also, the inclusion of unsupervised learning it will further improve the optimization by detecting zero-day attacks which are not previously known. Those solutions like autoencoders or anomaly detection models can be trained to learn new attack patterns that do not have any signature associated with it.

Explainable AI: Systems based on explainable AI techniques will help the system be more interpretable. This rule will offer the justification of the prediction made by the model and this can be useful to allow security analyst's trust in the model decisions and to ease finding false positives or wrong predictions [26].

Federated learning: The federated learning methodology may be adopted for the preservation of privacy, permitting training across distributed Windows systems without sharing private information. This may enable groups to collaborate on developing IDS systems while remaining privacy-preserving.

The mechanisms of counter-attack can be included in the future is a pressing issue. The model will update these strategies with automated responses (traffic redirection, firewall rules update) thus ultimately allowing it to recognise the attack and take counter-measures in real-time.

In summary, the hybrid IDS system proposed in this research is an important step forward for cybersecurity in general and especially its capability of detecting malicious software that targets Windows environments. The model combines the state-of-the-art machine and deep learning models using CNN-LSTM architecture with ensemble approaches (Random forest, LightGBM, XGBoost and CatBoost). This blend of ingredients results in optimally effective security that scales to grow with any organization, and we push through challenges faced by point solutions -such as data imbalances, overfitting and high rates of false positives — to reach the uncommon height of 97.4% accuracy for detecting complex threats (e.g., ransomware and fileless attacks).

With the rising trend of cyber-attacks, it is fundamental to revolutionize this hybrid IDS architecture in order for Windows system to become more secure and resilient. The targeted next enhancements will then result in more adaptive and proactive IDS, enabling companies to anticipate the changing treat and protect critical infrastructure.

REFERENCES

[1] R. a. L. S. Sinha, "Study Of Malware Detection Using Machine Learning," *ANVESAK ISSN: 0378-4568, Impact Factor: 6.20,* vol. 51, pp. 145--154, 2021.

[2] F. J. a. O. A. a. A. G. I. a. A. T. K. Akinshola-Awe, "Framework for the detection and classification of malware using machine learning," *Dutse Journal of Pure and Applied Sciences,* vol. 10, pp. 177--186, 2024.

[3] R. a. A.-K. A. Hilabi, "Windows operating system malware detection using machine learning," *Bulletin of Electrical Engineering and Informatics,* pp. 3401--3410, 2024.

[4] N. a. S. J. Moustafa, "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Information Security Journal: A Global Perspective,* vol. 25, pp. 18--31, 2016.

[5] N. Z. a. S. A. a. K. O. Gorment, A recent research on malware detection using machine learning algorithm: Current challenges and future works, 2021.

[6] N. A. a. O. O. E. a. M. S. a. O. J. a. D. Azeez, "Windows PE malware detection using ensemble learning," in *Informatics*, 2021, p. 10.

[7] R. a. P. L. Chiwariro, "Malware detection and classification using machine learning algorithms," *Int J Res Appl Sci Eng Technol,* vol. 11, pp. 1727--1738, 2023.

[8] N. a. S. J. Moustafa, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 military communications and information systems conference (MilCIS),* 2015, pp. 1--6.

[9] S. a. R. F. A. a. R. A. a. G. W. a. Y. W. Rosyada, "Enhancing XGBoost Performance in Malware Detection through Chi-Squared Feature Selection," *Jurnal Sisfokom (Sistem Informasi dan Komputer),* vol. 13, pp. 396--402, 2024.

[10] D. a. o. Hemanth, “Improved malware detection for IoT devices using random forest algorithm comparing with decision tree algorithm,” 2022.

[11] N. a. S. J. a. C. G. Moustafa, “Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks,” *IEEE transactions on big data,* vol. 5, pp. 481--494, 2017.

[12] A. Al-Marghilani, “Comprehensive analysis of iot malware evasion techniques,” *Engineering, Technology \& Applied Science Research,* vol. 11, pp. 7495--7500, 2021.

[13] M. a. L. S. a. M. N. a. P. M. Sarhan, “International Conference on Big Data Technologies and Applications,” *Springer,* pp. 117--135, 2020.

[14] Y. B. Parmar, Windows Portable Executor Malware detection using Deep learning approaches, 2020.

[15] D. a. o. Hemanth, “Improved malware detection for IoT devices using random forest algorithm comparing with decision tree algorithm,” 2022.

[16] M. a. K. D. a. I. S. a. B. S. a. A. M. Azeem, “Analyzing and comparing the effectiveness of malware detection: A study of machine learning approaches,” *Heliyon,* vol. 10, 2024.

[17] V. a. M. A. a. B. I. Verma, “International Journal of Performability Engineering,” *Analyzing and classifying malware types on windows platform using an ensemble machine learning approach,* vol. 20, p. 312, 2024.

[18] E. G. V. a. C. J. G. Enriquez, “Dynamic malware analysis using machine learning-based detection algorithms,” *Interfases,* pp. 119--138, 2024.

[19] U. a. R. K. H. K. a. C. K. Divakarla, “A novel approach towards windows malware detection system using deep neural networks,” *Procedia Computer Science,* vol. 215, pp. 148--157, 2022.

[20] J. a. A. M. Note, “Comparative analysis of intrusion detection system using machine learning and deep learning algorithms,” *Annals of Emerging Technologies in Computing (AETiC),* vol. 6, pp. 19--36, 2022.

[21] N. a. A. F. Selamat, “Comparison of malware detection techniques using machine learning algorithm,” *Indones. J. Electr. Eng. Comput. Sci,* vol. 16, pp. 435--440, 2019.

[22] M. A. M. a. A. Z. a. A. F. A. H. a. S. K. A. M. a. H. H. S. Yusof, “Detecting malware with classification machine learning techniques,” *International Journal of Advanced Computer Science and Applications,* vol. 14, 2023.

[23] M. A. M. a. A. Z. a. A. F. A. H. a. S. K. A. M. a. H. H. S. Yusof, “Detecting malware with classification machine learning techniques,” *International Journal of Advanced Computer Science and Applications,* vol. 14, 2023.

[24] N. Z. a. S. A. a. K. O. Gorment, “A recent research on malware detection using machine learning algorithm: Current challenges and future works,” in *International Visual Informatics Conference*, Springer, 2021, pp. 469--481.

[25] O. a. U. S. a. A. T. a. S. S. a. A. D. A. a. A. M. a. B. A. a. A. R. Khalid, “n insight into the machine-learning-based fileless malware detection,” *Sensors,* vol. 23, p. 612, 2023.

[26] F. O. a. O. A. A. a. E. O. O. Gbenga, “Towards optimization of malware detection using extra-tree and random forest feature selections on ensemble classifiers,” *The International Journal of Recent Technology and Engineering (IJRTE),* vol. 9, pp. 221--212, 2021.

[27] O. a. U. S. a. A. T. a. S. S. a. A. D. A. a. A. M. a. B. A. a. A. R. Khalid, “An insight into the machine-learning-based fileless malware detection,” *Sensors,* vol. 23, p. 612, 2023.

[28] S. S. a. P. K. K. hmad, “A Novel Machine Learning Framework for Analyzing Performance of Different Prediction Models by Using Automatic Malware Detection (AMD) Algorithm,” *Apex Journal of Business and Management,* vol. 1, pp. 11--20, 2023.

[29] A. a. K. P. a. B. A. a. J. S. Kamboj, “Detection of malware in downloaded files using various machine learning models. Egyptian Inform. J. 24 (1), 81--94 (2023)”.

[30] A. a. B. A. a. R. S. a. M. I. M. Djenna, “Artificial intelligence-based malware detection, analysis, and mitigation,” *Symmetry*, vol. 15, p. 677, 2023.

[31] F. O. a. O. A. A. a. E. O. O. Gbenga, “Towards optimization of malware detection using extra-tree and random forest feature selections on ensemble classifiers,” *The International Journal of Recent Technology and Engineering (IJRTE)*, vol. 9, pp. 221--212, 2021.

[32] M. Kumar, “Scalable malware detection system using big data and distributed machine learning approach,” *Soft Computing*, vol. 26, pp. 3987--4003, 2022.

[33] “Comparative analysis of intrusion detection system using machine learning and deep learning algorithms,” *Annals of Emerging Technologies in Computing (AETiC)*, vol. 6, pp. 19--36, 2022.

[34] A. a. K. A. A. a. A. M. a. T. A. a. J. A. a. K. M. A. U. a. K. S. Ijaz, “Innovative machine learning techniques for malware detection,” *Journal of Computing \& Biomedical Informatics*, vol. 7, pp. 403--424, 2024.

APPENDICES

Appendix A

Exploratory Data Analysis (EDA)

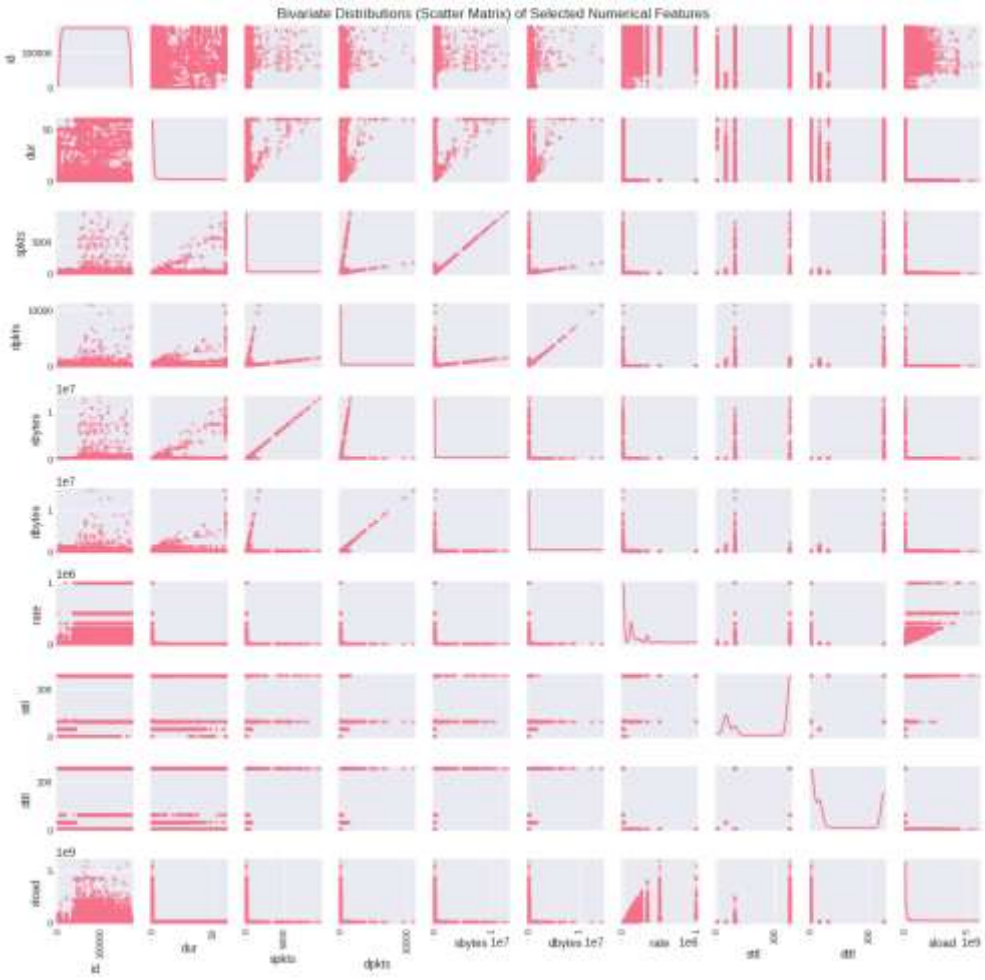


Figure: Appendix A.1

This is an illustration of a scatter plot matrix, or pair plot, showing the relationships between several numerical features in a training dataset. In each off-diagonal cell, we have displayed a scatter plot of two features and on the diagonal the distribution of a feature – in many cases histograms or density plots. The matrix consists of such variables as id, dur (duration), spkts(source packets) dpkts(destination packets) sbytes, dbytes sload and R sttl,dttl,rate. The scatter plots show the relationship between any two

variables, and can be used to identify patterns in a data set that might include clusters or outliers.

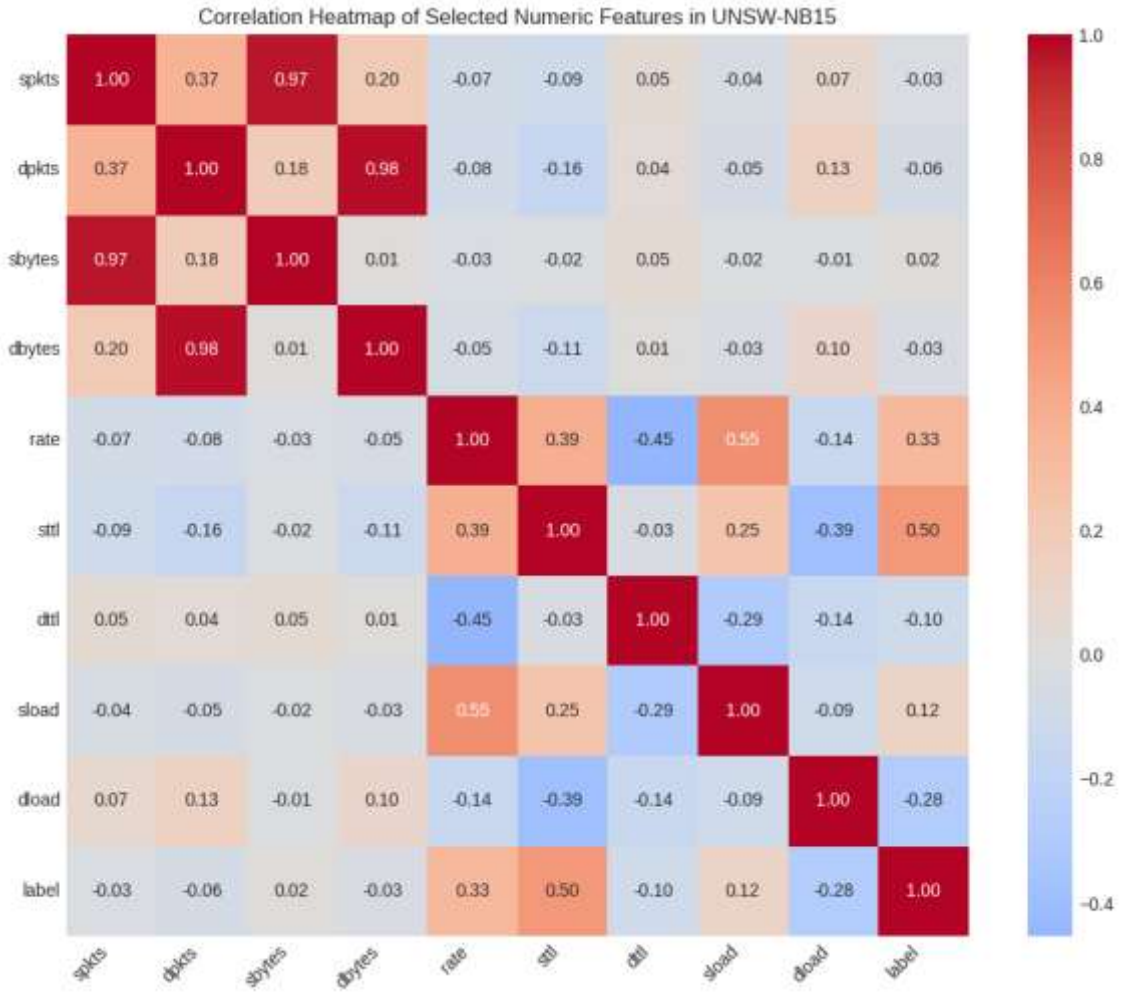


Figure: Appendix A.2

This is the output of the correlation heatmap, which displays how numerical features in UNSW-NB15 are related. Each value of the grid corresponds to the correlation coefficient between pairs of variables, varying between -1 and 1 (perfect negative and positive relationship respectively). The color of the emitter reveals the level of these correlations: red corresponds to positive and blue to negative. The high positive correlations are seen between such features as spkts (source packets), sbytes (source

bytes) and rate, dttl(destination TTL) etc. and low negative correlations are observed in the case of rate, ctflw & dttl. The heatmap facilitates comprehension of the interrelations between features to be useful in subsequent analysis, or feature selection.

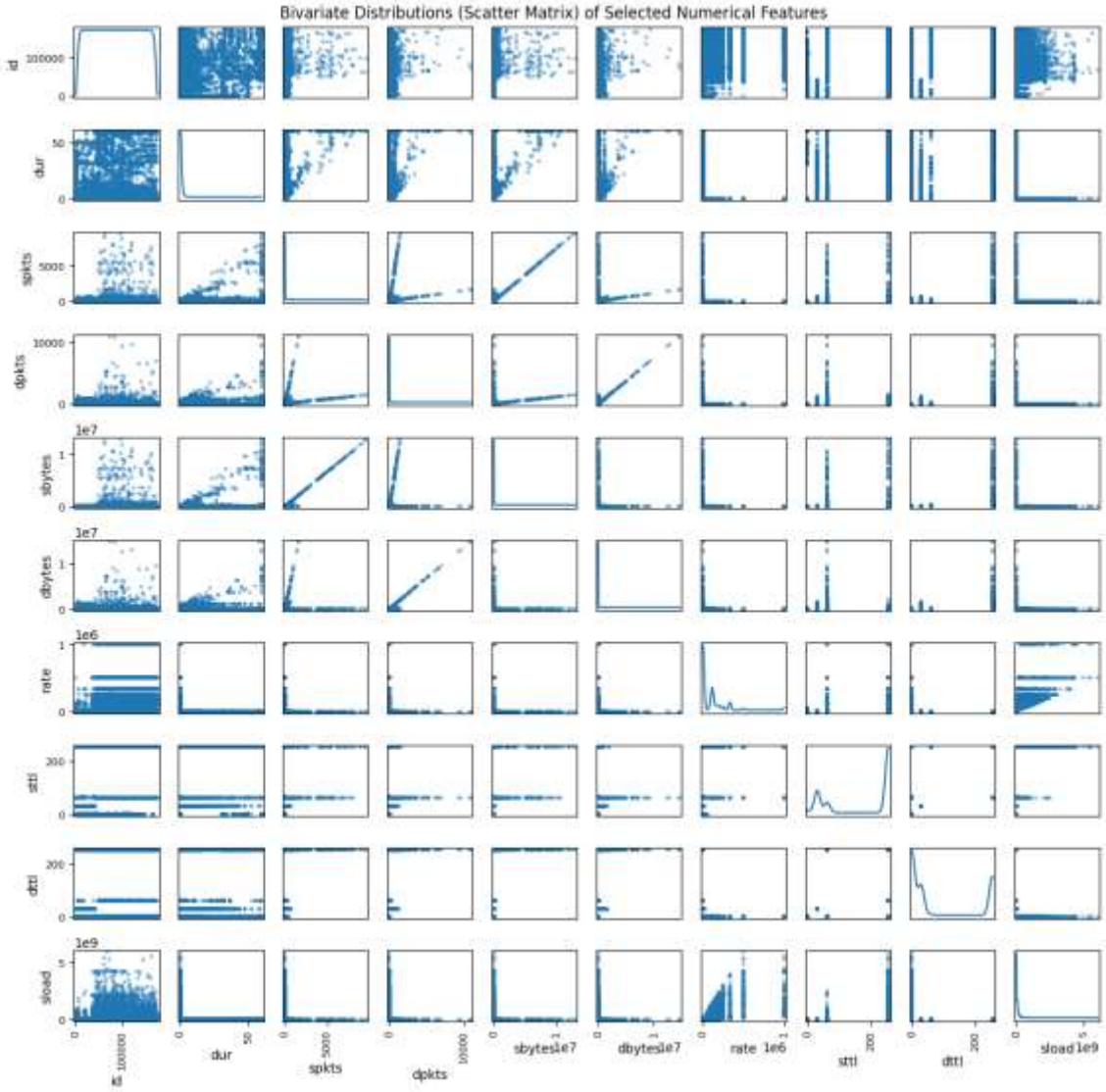


Figure: Appendix A.3

This is the scatter plot matrix, or bivariate distribution plot (i.e. plotting all of our continuous variables against each other). Each cell of the grid shows a scatter plot between two variables, while for the diagonal cells, it shows the distribution the one variable. The various attributes also include, for example, dur (duration), spkts (source

packets), dpkts (destination packets) and the like between pairs. These are plotted in different ways and densities, to exhibit us the correlation and distribution characteristic of these features. Lots of the plots reveal data that is sparse but some cluster in regions as being shown where they are situated from the information points.

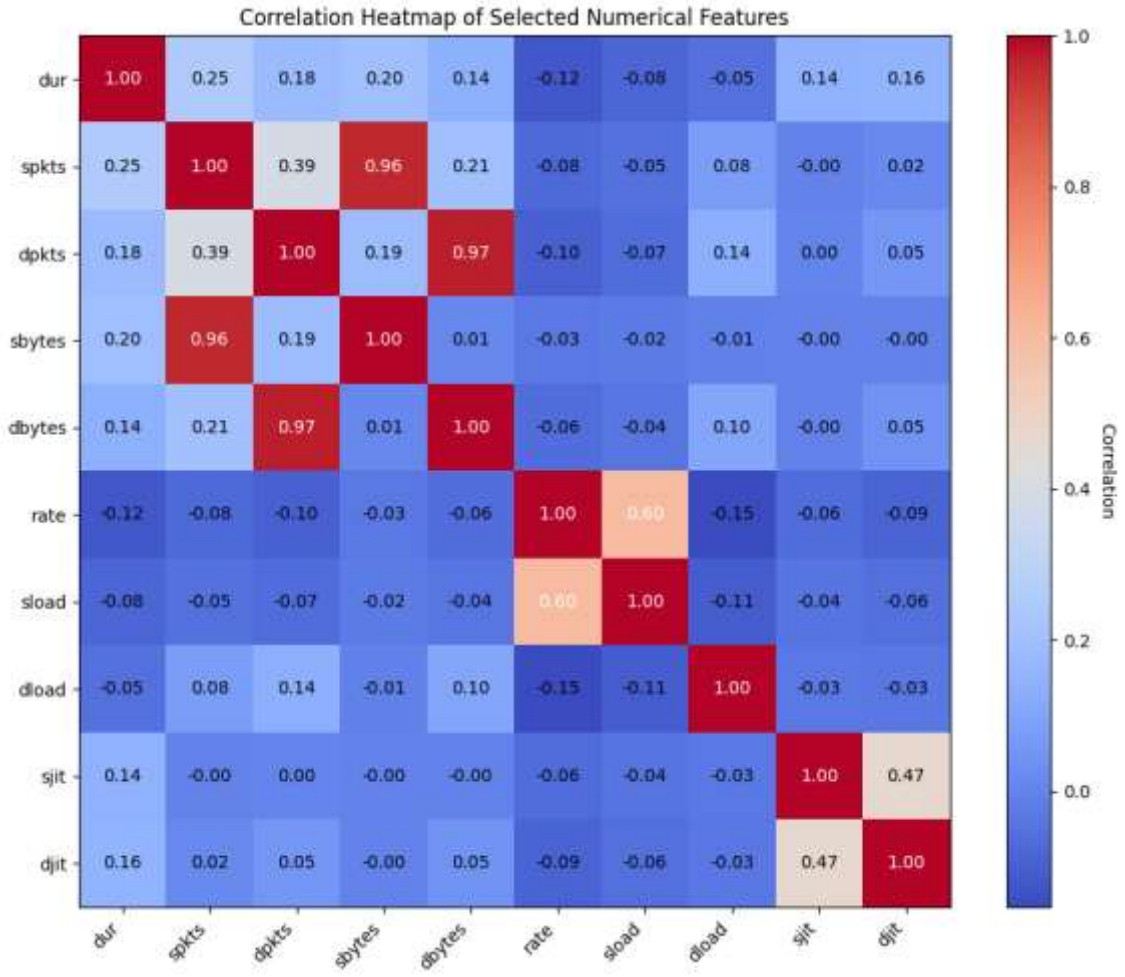


Figure: Appendix A.4

The above is a correlation heatmap of some numeric fields of a data frame. The number is a correlation coefficient between two variables and it ranges from -1 (for perfect negative dependence) to 1 (perfect positive dependence). The colour of dendrogram represents the magnitude of correlation where red is high positive correlation, blue is

weak or negative correlations. For instance, spkts(souce packets) has a strong positive relation to sbytes(source bytes), and rate and sload(system load) also have moderate positive relations. This heatmap is useful in analysing the dependencies and relationships between features based on which one can identify patterns & see if there are certain features that not mean much.

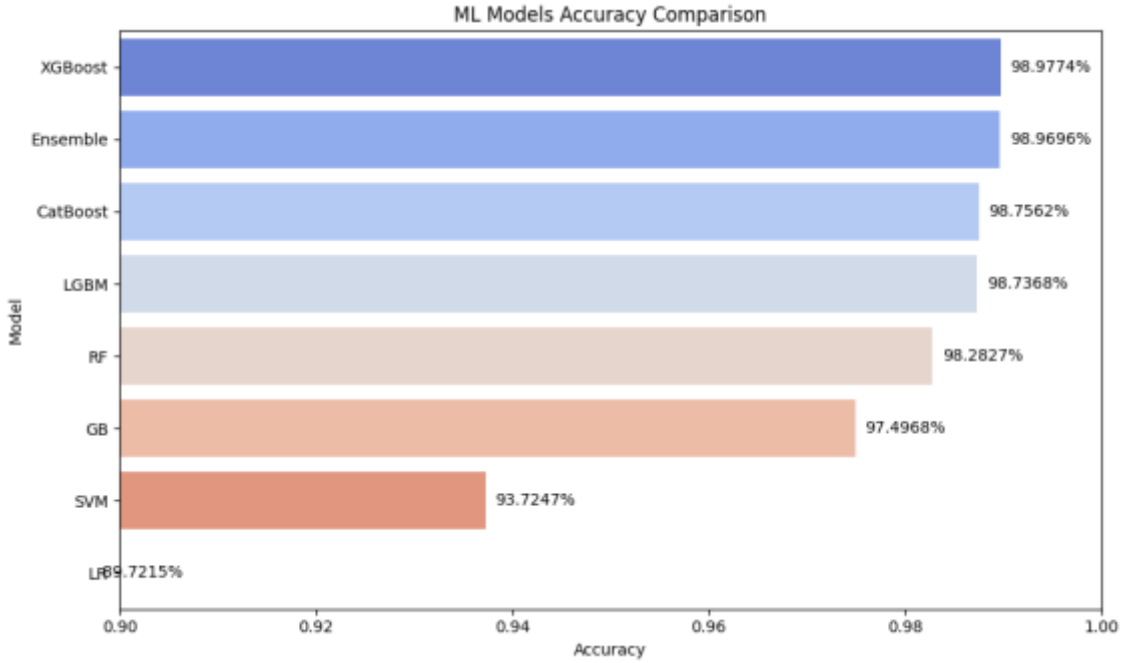


Figure: Appendix A.5

This bar graph presents an analysis between accuracies of various models such as XGBoost, Ensemble model, Cat Boost, LGBM, Random Forest (RF), Gradient Boosting (GB), Support Vector Machine and Logistic Regression. Models are ordered based on the accuracy; the XGBoost model with an accuracy of 98.9774 and Ensemble with an accuracy of 98.9696 are at the top of our list. Similarly, other also demonstrate high accuracy up to 98.28–98.76%. SVM and LR's accuracy is in lower than other models, being 93.72 % for SVM, 89.72 % for LR. Thus, this graph allows us to compare visually the performance of these models on the dataset.

Appendix B

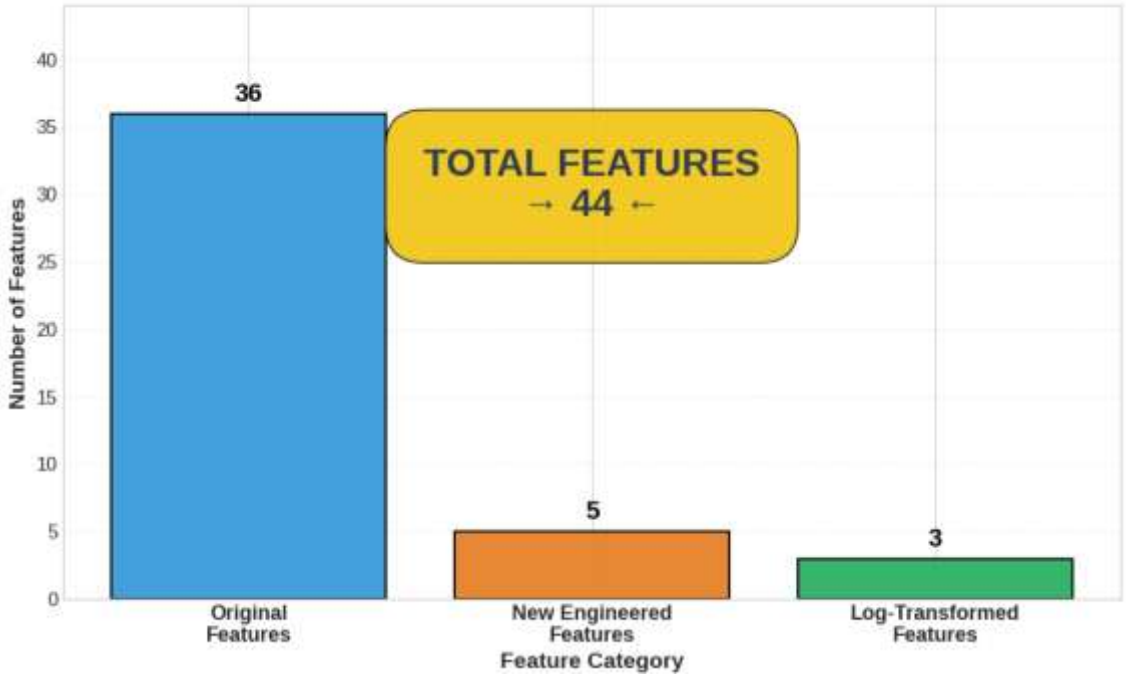


Figure: Appendix B.1

Original Features (36 features):

The majority of the features, totaling 36, are the original features derived directly from the raw data. These features form the foundational set for analysis and capture essential aspects of the network traffic, such as packet size, duration, and protocol type.

New Engineered Features (5 features):

To enhance the dataset, 5 new engineered features were introduced. These features include is_tcp, is_udp, total_bytes, byte_ratio, and pkt_ratio. They were specifically created to provide deeper insights into the network traffic, such as distinguishing between TCP and UDP connections, quantifying the total data exchanged, and analyzing the relative distribution of traffic between the source and destination.

Log-Transformed Features (3 features):

3 log-transformed features were created to address skewness in the data and minimize the impact of outliers. By applying a logarithmic transformation to dur_log, sbytes_log, and dbytes_log, the distribution of connection durations and data volumes was normalized, improving the robustness of the analysis.

Together, these features provide a comprehensive and balanced representation of network traffic, resulting in a total of 44 features used for the analysis. This enhanced feature set ensures a more effective and insightful examination of the network data.

Table Appendix B.1

Feature Name	Type	Description
is_tcp	NEW	Indicates whether the connection is TCP.
is_udp	NEW	Indicates whether the connection is UDP.
total_bytes	NEW	Total number of bytes transferred.
byte_ratio	NEW	Ratio of bytes transferred between source and destination.
pkt_ratio	NEW	Ratio of packets transferred between source and destination.
dur_log	LOG	Log-transformed duration of the connection.
sbytes_log	LOG	Log-transformed number of source bytes transferred.
dbytes_log	LOG	Log-transformed number of destination bytes transferred.

Several new features were engineered, and existing features were log-transformed to enhance the model's performance. The newly engineered features include is_tcp and is_udp, which indicate whether the connection uses TCP or UDP, respectively, helping to differentiate between reliable and unreliable protocols. The total_bytes feature quantifies the total data transferred during a session, while byte_ratio and pkt_ratio

measure the distribution of data and packets between the source and destination, respectively.

Log transformations were applied to the features dur_log, sbytes_log, and dbytes_log to normalize the distribution of connection durations and data transfer volumes. These transformations help mitigate the impact of outliers, improving the robustness of the model. Together, these engineered and transformed features enable more accurate analysis of network traffic patterns, bringing the total number of features used to 44.

