A HYBRID MACHINE LEARNING BASED MODEL TO ENHANCE DETECTION OF ZERO DAY ATTACKS

DOMINIC JOHN KAVOI

A THESIS SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY IN THE SCHOOL OF COMPUTING AND MATHEMATICS IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF MASTER OF SCIENCE IN CYBER SECURITY OF THE CO-OPERATIVE UNIVERSITY OF KENYA.

2025

DECLARATION

Declaration by the Candidate

I hereby affirm this thesis is my original work and has not been presented for a degree in any other University or for any other award. Unless explicitly indicated through references or acknowledgments, all work contained herein is my original contribution.

Signature: ..... Date: 21st November 2025

Dominic John Kavoi

MCSC01/6004/2022

Declaration by the supervisors

We confirm that the work reported in this thesis was carried out by the candidate under our supervision and has been submitted with our approval as university supervisors

Signature: ..... Date: Date: 21st November 2025

Dr. Charles Jumaa Katila, Computer Science and Information Technology, Computing and Mathematics, Cooperative University of Kenya

Signature: ..... Date: Date: 21st November 2025

Dr. Richard Omolo, Computer Science and Information Technology, Computing and Mathematics, Cooperative University of Kenya

DEDICATION

I dedicate this work to my late mum, Marietta Syokwaa Kavoi, and my late grandmother, Elizabeth Ndote Kavoi. Your love inspires me. May your souls continue resting in eternal peace.

ACKNOWLEDGMENT

I wish to express my deepest gratitude to my supervisors, Dr. Charles Jumaa Katila and Dr. Richard Otieno Omollo, for their invaluable guidance, support, and encouragement throughout the course of this research. Their expertise, patience, and constructive feedback were instrumental in shaping this work.

I am also profoundly grateful to my family for their unconditional love, prayers, and constant support, which gave me the strength and motivation to complete this thesis. To all who contributed in any way to the success of this work, I extend my heartfelt thanks.

TABLE OF CONTENTS

List of Tables

List of Figures

List of Equations

Abbreviations/ Acronyms

AI	Artificial Intelligence
ANOVA	Analysis of Variance
API	Application Programming Interface
APT	Advanced Persistent Threat
CAK	Communications Authority of Kenya
CD	Compact Disk
CIA	Confidentiality, Integrity and Availability
CSV	Comma Separated Value
DL	Deep Learning
DT	Decision Tree
DDoS	Distributed Denial of Service
DoS	Denial of Service
DTC	Decision Tree Classifier
ECU	Electronic Control Units
EDA	Exploratory Data Analysis
FAR	False Alarm Rate
FCM-ANN	Fuzzy Clustering Means Artificial Neural Networks
GNB	Gaussian Naïve Bayes
HC	Hierarchical Clustering
IDS	Intrusion Detection Systems
IPS	Intrusion Prevention Systems
IT	Information Technology
KNN	K-Nearest Neighbor

MAC	Media Access Control
MITM	Man-In-The-Middle
ML	Machine Learning
NACOSTI	National Commission for Science Technology and Innovation
NN	Neural Network
PCA	Principal Component Analysis
PDF	Portable Document File
RF	Random Forest
ROC	Receiver Operating Characteristic
SB	Spam Bayes
SVM	Support Vector Machines
USB	Universal Serial Bus

ABSTRACT

In the current technological landscape, a lot of risks are present due to the availability of existing and novel kinds of attacks. For these attacks to be countered, systems that identify all the variants without any false positives and false negatives are in high demand. The existence of traditional attack detection methods, such as the signature-based algorithms, have proven that they cannot spot new attack. This is because they work based on a database that has signatures of attacks. This research improved the low accuracy of systems that use singular or hybrid machine learning algorithms or signature-based detection algorithms that are used to identify zero-day attacks. This study analyzed existing algorithms used for detecting zero-day attacks, designed an hybrid model to address the identified gaps, implemented the hybrid model and finally validated its performance to ensure its effectiveness. The other methods of detecting attacks that have been explored in this study are the hybrid and machine learning methods for detecting the zero-day attacks. In this research, the main aim was to come up with an hybrid set of machine learning model that identify novel and existing attacks in real time from an existing dataset. All of these concepts were mainly based on the Confidentiality, Integrity and Availability (CIA) triad. The study had a firm foundation based on theorems such as Bayes and the fundamental principles of computational learning theory. The methodology used the machine learning pipeline which was composed of stages such as the identification, cleaning, analysis and feature engineering of the data. The dataset had a total of 3.67 million rows and five columns. From there, the hybrid models were implemented, their accuracy measured and then tuned to improve its efficiency. Majorly, the model achieved an accuracy of about 97% through the accuracy metrics used. The metrics used include F1-score, precision and recall. The study found that the hybrid algorithm outperformed individual classifiers and traditional methods by achieving higher accuracy and better generalization to unseen attacks. Its contributions include informing policy on the use of intelligent threat-detection systems, offering a practical and robust real-time detection approach, and advancing academic knowledge on machine learning hybrids for zero-day attack detection.

CHAPTER ONE

1. INTRODUCTION

1.1 Background

The rapid integration of digital technologies into essential economic and social activities has fundamentally altered how individuals and organizations operate across the globe. Some activities are making internet an inevitable resource required while carrying out various tasks such as shopping, studying, gaming, conversing, and financial activities among others (Kumar & Sinha, 2021). A great portion of the population is today depending on the internet to undertake activities that are carried out on a daily basis. This makes them susceptible to various cyber-attacks and threats (Thangavel & Kannan, 2019). These threats and attacks are propagated and implemented through malicious programs referred to as malware. Malware are applications that are explicitly developed to perform mischievous activities that cause unauthorized access to a computer system, disrupting or damaging the underlying computer system and data Hindy *et al.* (2020). According to a report published by Interpol (2020), different types of cyber threats emerge day in day out, making the provision of cyber security difficult and sophisticated as internet's traffic increases from a diverse set of users.

From a global perspective, a lot of research has been done with regards to the various methodologies of artificial intelligence and how they can be used in cyber security. For instance, there have been artificial intelligence systems that have used data mining, behavior and deep learning concepts in order to detect zero-day malware. The normal functionality of these systems has been to use classification methodologies to come up with subclasses. The main categories that have been obtained as a result of this process are either malicious or benign data. A common strategy used today is neural networks and Support Vector Machine (SVM) algorithms. Secondly, hybrid algorithms have also been used in these instances. One main example of an algorithm that is used is the random forest algorithm. This algorithm operates through the use of votes whereby the majority winner in each instance is considered as the result of that process. This in return has also provided a great opportunity for the exploration of clustering algorithms (Abri *et al*., 2019). From the local Kenyan perspective, there are a lot of challenges that are coming up due to the flaws in the existing systems that are there to curb and minimize cyber security challenges. According to Communications Authority of Kenya (2018), report there was a huge challenge when it comes to the protection of data belonging to

employees. By the year 2017, it was also accounted that over 21 billion shillings had been lost because of attacks in the cyber space. The common attacks that were encountered include brute forcing, denial of service and botnets. With all these attacks being dynamic, there is a great need for the utilization of machine learning to detect new versions of these attacks from the local side.

The detection of malicious activities and the offering a safe ecosystem against cyber-attacks has become a priority for researchers and security firms. A common technique to detection of malware is through the use of signatures. This concept involves the use of patterns for a suspicious against available malware patterns. Systems such as Intrusion Detection Systems (IDS) habitually employ such techniques and tools that match patterns to device rule-based detection techniques (Rieck *et al.* 2019). Signature and the use of rules as detection mechanisms that rely on matching patterns have been proven to detect only a trivial category of all types of malicious items. These methodologies are therefore deemed as less efficient in identification of more advanced and unidentified, or freshly developed viruses, performing even more poorly in detection of malware programs that have no history or a straight forward remediation approach, known as zero-day malware (Radhakrishnan *et al*. 2019).

According to Zhou & Pezaros (2020), malware detection systems that use signature-based techniques may not be able to identify zero-day malware programs. Therefore, instead of employing syntactical techniques, analytical approaches such as machine learning (ML) or deep learning (DL) should be used. Machine learning is a division in Artificial Intelligence (AI) whose models encompass a collection of algorithms using statistics that can learn from a pool of data and extract complex trends that are uneasily observable concepts. The learnt and generated trends are utilized in forecasting, categorizing, and regressing future occurrences and scenarios (Faranak *et al*. 2019). This motivation has led to a noticeable amount of research, being geared towards implementation of ML models in cybersecurity with the aim of augmenting and reinforcing existing techniques to detect sophisticated malware programs that require modern advanced detection capabilities. As noted by Sarhan *et al*. (2023), introduction of intelligence component to existing cyber security threat detection strategies, can add a complicated layer of security that can minimize the rate of occurrences of threats if planned correctly.

Internet environment safeguarded using signature or rule-based detection systems are susceptible to modern attacks without indicators that show compromise and are linked to the

malware program. It is therefore important to focus on ML based technique that will scan and analyze internet traffic for any malicious intent. Over the past recent years, professionals have designed and developed ML-based algorithms that are meant to reinforce existing malware detection techniques. Most notable algorithms are conventional ML algorithms that cluster a given dataset into individual sub-classes by employing different clustering techniques from which patterns can be generated (Tavakoli and Yooseph, 2019). Other algorithms based on regression fit numerous linear regression metrics on the data to generate patterns required by ML model to learn from (Al-Rushdan *et al*., 2020). Other algorithms employ random forest approach to build several internal decision tree algorithms and then take the popular outputs of the internal decision tree algorithms to achieve clustering (Rieck *et al*. 2019). Combining the strengths of regression-based algorithms with random based algorithms can yield a hybrid algorithm that explore the concealed relationships between the dataset features to form a hierarchical model that performing clustering more efficiently, and thus detecting zero-day malware more accurately, which is the undertaking that this research seeks to explore.

There have been several challenges when it comes to the resolving of issues tied to joining machine learning and cyber security fields. For instance, it may be difficult to select the best machine learning algorithm that can match with the problem to be solved and the dataset at hand. From the technical side, there has also been missing values that result in bias that affect the accuracy of the algorithms. The next challenge is that even when the algorithms are solving a cyber security challenge, they can also fall prey to attacks. This may be due to the fact that there may be vulnerabilities that can be directly targeted towards them. It has been witnessed that attackers corrupt the input data that is sent to these algorithms so that they may be considered as benign. This will result in improper decision making by these algorithms. Some algorithms such as Principal Component Analysis (PCA) and SpamBayes have been noted to have the vulnerability to causative attacks. These attacks interfere with the process which the algorithms are trained (Maalem & Moapatra, 2022).

1.2 Statement of the Problem

The currently available models for the detection of zero-day attacks use signature and machine learning strategies which have been established to be of various weaknesses. Signature-based algorithms are unable to accurately identify zero-day attacks because they depend on predefined threat databases that only recognize previously recorded malware patterns (Einy, Oz, & Navaei, 2021). This creates a significant contextual gap, especially in environments such as Kenya's rapidly digitizing ecosystem where emerging and locally tailored malware variants

often appear before global signature repositories are updated. As noted by Sarhan *et al*. (2023), singular machine learning classifiers exhibit limitations because they depend on a single algorithm to make detection decisions. Empirical studies consistently report low accuracy, unstable performance across datasets, and increased rates of false positives and false negatives. As a result, malicious data may be misclassified as benign and benign data as malicious, leading to operational disruptions, weakened decision-making, and reduced system reliability (Holm, 2014). Despite numerous global studies exploring machine learning for malware detection, there remains an empirical gap in understanding the effectiveness of combined or hybrid models that can better capture complex malicious behaviors, particularly in contexts with dynamic threat landscapes.

Therefore, this research sought to address these contextual and empirical gaps by developing a higher-accuracy detection mechanism capable of distinguishing malicious and benign data reliably. This was done through the utilization of a hybrid model that used two classifiers to improve the detection of zero-day attacks. The two models used were Gaussian Naive Bayes and decision tree classifier. This approach aims to overcome the limitations observed in signature-based systems and standalone machine learning models and contribute new evidence on the potential benefits of hybrid detection strategies.

1.3 Objectives

1.3.1 Main Objective

The main objective of this research is to develop a hybrid machine learning based model for enhancing detection of zero-day attacks.

1.3.2 Specific Objectives

i) To analyze existing algorithms for the detection of zero-day attacks
ii) To design a hybrid model for detecting zero-day attacks
iii) To implement the hybrid model for detecting zero-day attacks
iv) To validate the developed model

1.3.3 Research Questions

i) What are the strengths and weaknesses of the existing zero-day attack detection models?
ii) How can hybrid models be designed to effectively detect zero-day attacks?

iii) How can the designed hybrid model be implemented using machine learning techniques?
iv) How can the performance of the developed hybrid model be validated?

1.4 Significance of the Study

This study provides practical benefits for cybersecurity professionals, system administrators, and organizations that depend on digital systems. These groups often struggle with tools that misclassify threats, leading to service interruptions, financial losses, and unnecessary alerts. By offering a more accurate way to distinguish between malicious and harmless activity, the study supports better decision-making and helps maintain stable and efficient security operations particularly in settings where quick responses are crucial.

The work is also useful to policymakers and regulatory bodies looking for evidence-based ways to strengthen national cybersecurity. By showing how intelligent detection models can improve visibility into emerging threats, the study offers insights that can guide the development of updated standards, policies, and best-practice frameworks for protecting critical systems (Abri *et al*., 2019). Academic researchers benefit as well, as the study adds to ongoing discussions on hybrid machine-learning methods and provides empirical evidence on how combining different models can enhance threat-detection performance, (Tavakoli, 2019).

In addition, the study fills a methodological gap by examining how a hybrid classification approach can address limitations commonly seen in single-algorithm systems (Namin *et al*., 2020). This contribution advances the field by demonstrating how multiple learning models can work together to recognize complex and evolving malicious behaviors more effectively. The findings lay the groundwork for future research and innovation in automated threat detection, encouraging the design of more adaptable and data-driven cybersecurity solutions, (Rieck *et al*. 2019).

1.5 Scope of the Study

The study limited its scope to zero-day attacks and their detection using ML algorithms. This is because ML processes are assumed to have a capability to detect sophisticated modern attacks unknown to system developers that require progressively advanced detection strategies. The dataset used in the project was the Controller Area Network dataset. The main algorithms used in this project were Gaussian Naïve Bayes and Decision Tree Classifier.

1.6 Limitations of the Study

This study was subject to several limitations that may have influenced the overall outcomes and generalizability of the findings. First, the research heavily depended on the completeness and quality of the dataset used for training and testing the machine learning models. Since the study relied on previously collected intrusion-related data, any inherent biases, missing values, class imbalance, or limited representation of certain attack types may have affected the performance of the implemented algorithms. These dataset constraints also made it difficult to fully capture the complexity and unpredictability of real-world zero-day attacks.

Additionally, the study focused exclusively on supervised machine learning techniques, which require labeled data to function effectively. This limited the exploration of alternative methods such as unsupervised or hybrid approaches that may potentially offer better detection of unknown attack patterns. The scope of the research was also restricted to evaluating selected algorithms within a controlled experimental environment. As a result, the findings may not fully reflect how these models would perform in dynamic, large-scale, or high-traffic production networks where system resources, latency, and real-time adaptability become critical factors.

1.7 Organization of the Study

The research study will be organized into several key sections. The introduction will outline the origin of the problem, the reasons behind it, and the objectives of the study. It will also provide a brief overview of the activities that will take place throughout the project. The literature review will discuss previous works related to the project, highlighting their scope, achievements, and the gaps they left. In addition, the operational framework guiding the implementation of the study will be described.

The third section is the methodology, which details the steps used to conduct the research project. This phase will include the research design as well as aspects of data analysis. In the discussions, conclusions, and recommendations sections, trends, patterns, and comparisons with previous studies will be examined. The conclusion will summarize the main findings of the project. Finally, potential future work will be outlined to provide a possible roadmap for individuals who may wish to advance this study further.

CHAPTER TWO

2. LITERATURE REVIEW

2.1 Introduction

This research aims to detect zero-day malware through the use of a combination of several machine learning algorithms. There are two main benefits that are tied down to this process. The first one is that new attacks can be detected before they affect the systems they were intended to harm. This works on the principle of subjecting the algorithms to some dataset so that it can learn from it. Secondly, the algorithms to be used will be of high accuracy because the decision is obtained from an aggregation of results obtained from the several algorithms that will be used. From there, they will be put together to bring forth a joint output. The result that will be obtained is whether the data was malicious or benign. From there, actions can be taken from the administrative side to curb the attack vector before it gets into action in the system(s) it was meant to attack.

2.2 Theoretical Frameworks

The implementation of this study is backed by several theorems whose principles contribute directly to the results that would be obtained in the long term. Cumulatively, these are the components that work together to solve the problems that have been in existence when it comes to cyber security. The theorems that will be discussed in this section are the Bayes Theorem, Computational Learning Theory and Adaptive Resonance Theory.

2.2.1 Bayes Theorem

This is a theorem that is commonly used in statistics and machine learning models. This algorithm was founded by Thomas Bayes in the year 1763 (Berrar, 2019). This logical framework assists in providing standard outputs that are correct and assist in solving the existing problem(s). It works under the principle of providing some probability to whether an item is of some certain property that would be considered as the end result of this process. This theorem forms the foundation of the Naïve Bayes classifier machine learning algorithm. Since it is a formation component for machine learning algorithms, it works based on previous knowledge of data that is considered as the training set.

The theorem is represented using the equation shown below:

$$P(A|\mathrm{B}) = P(B|\mathrm{A})\mathrm{P}(\mathrm{A})/\ \mathrm{P}(\mathrm{B})$$

Equation 2.1: Bayes theorem

In the above equation, the statements below describe the meaning of each of them:

1. *P(A|B)* is the probability of an event A coming into action after event B has already occurred
2. *P(A|B)* is the probability of event B happening after event A has already occurred
3. *P(A)* is used to represent the event A happening
4. *P(B)* is used to represent event B happening

This is a great algorithm that has provided a stronger backing to the Naïve Bayes classifier that may be used in this project. Through this theorem, the algorithm has been able to provide powerful performance when put on the same table with other existing algorithms (Berrar, 2019).

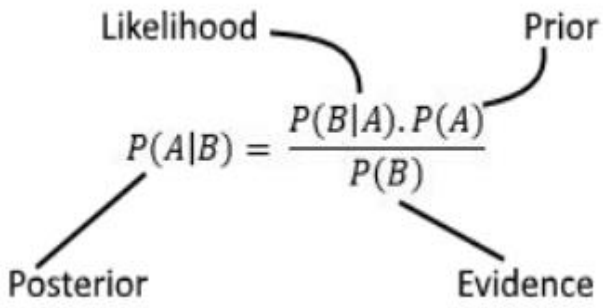


Figure 2.1: Bayes theorem

The Bayes Theorem is significant in the development of a hybrid machine learning model to detect zero-day attacks since it offers a structured way to estimate probabilities in order to classify. Within the domain of cyberattack detection, it is intended to identify whether a given object like a network packet or system action is malicious or benign depending on its attributes. Using the Bayes Theorem, such probabilities can be systematically computed creating the foundations of such classifiers as Gaussian Naïve Bayes which are frequently used in hybrid models. Zero-day attacks are hard to predict hence causing a lot of uncertainty in detection systems. The Bayes Theorem is useful in solving this issue because it enables the model to update its predictions as new data keeps coming in and thus it becomes possible to discover patterns of attack that were not previously known in real-time. In a hybrid design, the probabilistic output of a Bayes based classifier can be used together with the predictions of other models including decision trees to achieve better overall accuracy and reduced false

positives and false negatives. Also, the independence assumption of Gaussian Naive Bayes can be used to simplify the calculation of complex and high-dimensional data, such as network traffic, and is complementary to other classifiers which do not make the same assumption. The combination of the Bayesian Thesis into the hybrid would allow the-model to take advantage of a strong mathematical backbone, which would in turn increase the reliability of the model as well as its ability to cope with the hitches of detecting zero-day attacks, (Balayla, 2024).

2.2.2 Fundamental Principles of Computational Learning Theory

This is a theorem that is tied to providing efficient algorithms that can be used to improve the quality of results that can be obtained after the data has been parsed to them. In this theory, if group of hypotheses can be used as a learning point to obtain the desired output. This theory is greatly applied in classification algorithms such as the ones that will be implemented in this study. It operates on the basis of a hypothesis space that correlates the training processes of the algorithms to be used. A certain limit of an amount of data that is submitted for the training process is also important for the obtainment of the output that may be needed. Another great factor to consider with regards to what this theory is trying to solve is the amount of resources such as memory that will be used in the processing of data that will be relayed to it. In the scenario of this project, the dataset that may be used will be computed in a standard working laptop but still at the same time yield a high level of accuracy (Krendzelak & Jakab, 2016).

The significance of the Fundamental Principles of Computational Learning Theory (CLT) to this study is that it offers a guideline on how to design an effective and accurate learning algorithm. CLT points the importance of a hypothesis space, which determines the way models learn based on training data and generalize to new traits. It also focuses on the correlation between the size of training data, computation resources, and the algorithm performance so that models are effective and efficient. In this study, using CLT enables the hybrid machine learning hybrid to work with datasets in standard hardware and achieve a high level of accuracy in identifying the presence of zero-day attacks, which means that the model is practical and reliable (Krendzelak & Jakab, 2016).

2.2.3 The Decision Tree Theorem

The decision tree theoretical framework utilizes various different attributes to perform splits that would be used in the classification of the data. Attributes are used within each node to carry out the end-to-end implementation of the algorithm. A search process is used in the selection of an attribute that would be subjected to various nodes in the dataset. In a set of

nodes Y_1, Y_2, …, Y_n, an extremum of a function is used to select the best attribute that would be subjected to the node. From there, it will be subjected to the algorithm to get the best accuracy in the process. Statistical concept such as Analysis of Variance (ANOVA) are used as a major factor for the selection of the best attribute to be used in the process. The most commonly used criteria for the selection in this entropy is the information gain methodology (Crémilleux & Robert, 1997). In general, entropy is usually considered as the extent to which data samples are impure. It is calculated using the formula shown below (Abbas *et al*., 2021):

$$H(S) = \sum_{y \in X} p(y) \log 2 \frac{1}{p(y)}$$

Equation 2.2: Entropy formula

As for information gain, it is used to calculate how the nodes will be split until one that is closest to the target variable will be obtained. Another role that information gain plays is in the halting of the splitting process. This occurs when the optimum value that is closest to the target variable is obtained during the machine learning process (Abbas *et al*., 2021):

$$IG(S, A) = H(S) - \sum_{i=0}^{n} P\ (y) * H\ (y)$$

Equation 2.3: Information gain formula

The theory of Decision Tree is specifically applicable as it offers a clear and structured manner of classifying data based on its features which is essential in identifying zero-day attacks. A node of a Decision Tree will consider certain characteristics of the data to decide how best to split it in order to use the model into effective classification. The selection of the most informative attributes is based on statistical measure like entropy and information gain that assists in the evaluation of the impurity of the data and the extent to which each split enhances the classification. The process can be applied to ensure that the algorithm concentrates on the most useful features in order to differentiate between malicious and benign activity. In this research, one of the classifiers in the hybrid model is based on the theory of Decision Tree. Its capability to handle complex data and giving priority to critical attributes increases the capability of the model to identify known attacks and unknown attacks. Moreover, splitting and stopping criteria, which depend on information gain, are useful in obtaining high accuracy and preventing overfitting which forms a major consideration when dealing with zero-day

attacks. The incorporation of Decision Tree principles into the research allows basing its classification technique on an already established approach, which in turn increases the effectiveness and reliability of the hybrid model in detecting attacks in real time (Mienye & Jere, 2024).

2.3 Empirical Review: Machine Learning and Hybrid Models for Zero-Day Attack Detection

This section provides a critical empirical review of current research exploring machine learning (ML) techniques for detecting zero-day attacks, with a particular focus on hybrid and hybrid approaches. Although hybrid models are often presented in the literature as a promising solution to the limitations of standalone algorithms, empirical evidence reveals that their performance is highly inconsistent. Several published studies report accuracy levels below the commonly accepted benchmark of 90%, raising questions about the reliability, generalizability, and practical readiness of these systems. By examining hybrid models that fall short of these expectations, this review aims to present a balanced and realistic understanding of what has been achieved so far and where fundamental challenges remain.

2.3.1 The Rationale for Hybridization in Zero-Day Attack Detection

The need for hybrid approaches largely stems from the limitations of traditional signature-based intrusion detection systems, which struggle to detect novel or unseen threats. As researchers began relying more heavily on machine learning, early work focused on single algorithms such as Support Vector Machines (SVMs), Decision Trees, and simple Artificial Neural Networks. However, these monolithic models were repeatedly shown to struggle with the unpredictability of zero-day attacks, often overfitting to known attack types, generating high false positives, or failing to generalize to new threat behaviors (Miller, 2021). Hybrid models emerged as a response to these shortcomings, based on the idea that combining complementary algorithms could create a detection system that is more adaptive and resilient than any single model on its own.

2.3.2 Critical Analysis of Selected Hybrid Models

Chen and Wang (2022) proposed a lightweight hybrid model for IoT networks that combined K-Means clustering with the Local Outlier Factor (LOF). Their idea was to use K-Means to group similar network behaviors and then apply LOF within each group to detect local

anomalies. When tested on the Bot-IoT dataset, the model achieved an accuracy of 87.3%, showing some ability to detect low-footprint attacks. However, the model's performance was highly sensitive to hyperparameters such as the number of clusters (k) and LOF neighbourhood size, making its behavior unstable across different conditions. Furthermore, it struggled with sparse, irregular attack patterns that did not form clear clusters. This study highlights a key limitation of unsupervised hybrids: while they are lightweight and suitable for constrained environments, they often lack the discriminative strength needed for complex zero-day scenarios.

A separate study, described by Smith and Doe (2023), introduced a hybrid model combining a Naïve Bayes classifier with a rule-based inference layer. The Naïve Bayes component generated initial probability scores for potential threats, while the rule-based layer filtered these results using handcrafted rules based on network protocol behaviour. When evaluated on NSL-KDD data containing previously unseen attack types, the model reached an accuracy of 85.1%. Although precision improved for a few attack categories, the overall accuracy remained low. The authors attributed this mostly to the unrealistic independence assumption of Naïve Bayes—a common issue in network datasets where features tend to be interdependent. The rule-based filter also struggled to compensate for the inaccuracies generated by the probabilistic component, revealing a weak integration between the two techniques.

Rodriguez and Kumar (2023) explored a more advanced architecture, combining a Deep Autoencoder (DAE) for feature learning with a One-Class SVM for anomaly classification. The model was trained on normal traffic so that zero-day attacks would ideally produce higher reconstruction errors. However, when tested on the CIC-IDS2017 dataset, the hybrid achieved just 82.6% accuracy. The main issue was that some sophisticated zero-day attacks produced reconstruction errors similar to noisy or complex normal traffic, resulting in significant overlap between the two classes. This made it difficult for the One-Class SVM to create a reliable boundary between normal and malicious behavior. The study illustrates a deeper challenge: even with advanced representation learning, if the learned feature space does not clearly separate normal and attack patterns, hybridization alone cannot solve the problem.

2.3.3 Synthesis of Empirical Findings and Research Gaps

Across the reviewed studies, several recurring challenges become evident. First, many hybrid models suffer from high sensitivity to hyperparameters, making them unstable in real-world

conditions where network patterns continuously change (Chen & Wang, 2022). Second, as Rodriguez and Kumar (2023) show, poor feature separability remains a major barrier—if the features themselves do not provide clear distinctions between normal and malicious behaviour, even sophisticated hybrids will struggle. Third, some models exhibit weak architectural integration, where the chosen algorithms do not effectively compensate for each other's weaknesses (Smith & Doe, 2023). Finally, performance on modern datasets such as CIC-IDS2017 and Bot-IoT is consistently lower than on older datasets like NSL-KDD, suggesting limited generalizability and a tendency for models to overfit simpler or outdated data structures.

This shows that designing an effective hybrid model for zero-day detection goes far beyond simply combining two algorithms. Success depends on selecting complementary models, ensuring strong and discriminative feature engineering, optimizing hyperparameters, and evaluating performance on realistic, diverse datasets. The present study positions itself within these gaps by aiming to develop a hybrid that leverages the complementary strengths of Gaussian Naïve Bayes and Decision Trees while addressing the shortcomings identified in previous empirical work.

2.3.4 Categories of Attacks in the Cyber Security Domain

In cyber security, there are two main categories that are commonly encountered. These are the active attacks and passive attacks. The operation of the attacks is based on the Confidentiality, Integrity and Availability (CIA) triad of data as shown in figure 2.1. Active attacks are mainly aimed to gain access into systems that rightful administrative permissions have not yet been granted to whoever is trying to conduct the attack. This attack can also be attributed to causing disturbance in the systems that it was sent to. For example, when a Denial of Service (DoS) attack is sent to an environment, it would cause interferences such as data packets being dropped and resources such as memory depleted. It is accomplished by flooding a network with an overflow of packets beyond the system's ability to withstand that flow (Rong & Çayirci, 2009).

Figure 2.2: The CIA triad

Before an active or passive attack is carried out, the adversary who intends to launch an attack on a system stays in the system in an undetected mode for a significant period. This situation forms the buildup of an Advanced Persistent Threat (APT). The reason behind this is to enable him/her to get a clear understanding of how the system operates in order to accomplish the routines that it usually performs. Active attacks have the ability of compromising the threefold security structure of systems. The major components of the network that are also analyzed during this period are the Data-Link (MAC), Physical, transport, network and application layers. For each of the layers listed above, there is a set of attacks and defenses that can be implemented on them (Kocakulak & Butun, 2017).

When it comes to passive attacks, they are considered as ones that cannot be detected by various means such as the utilization of firewalls, Intrusion Prevention Systems (IPS) and Intrusion Detection Systems (IDS). They can operate as Man-In-The-Middle (MITM) attacks in order to intercept communications that are being exchanged back and forth between the sender and the receiver. An example of an attack that can be carried out during this process is the eavesdropping attack. It works under the principle of listening to communication among nodes in a network. Due to this, the main item that is breached in the Confidentiality, Integrity and Availability (CIA) triad is confidentiality (Butun, Osterberg, & Song, 2020).

2.3.5 Zero-Day Attacks

This has been established to be an immeasurable security threat in a quantified manner due to the dynamism it entails when released into operation to system(s) being targeted. They tend to be unpredictable because of the vast dimensions of ideas attackers can use to compromise the systems that may be relied by authorized users. Its second property is that there exists a very

huge difficulty when it comes to the eradication of the zero-day attacks once they have already occurred. These attacks take advantage of the vendors' lack of knowledge of the flaws that exist in the systems(s) they provide support to. It has also been noted that the protection of systems against zero-day attacks has been noted to be something complex. The mode of operation can be tied to strategies such as affecting accounts and/or machines with malware meant to achieve various objectives (Franklin & Unikop, 2022).

Zero-day attacks can take up a similar approach to the core operation of other kinds of attacks that have been realized. The difference mainly comes in due to the fact that it can use a novel, unknown or a slightly different methodology of technical operation due to the uniqueness of its implementation procedure. For instance, according to (Kaur & Singh, 2014), the most difficult attack to detect was identified to be the polymorphic worms. The attack is considered as a denial-of-service attack because it is intended to deplete resources such as memory from devices that may need to utilize it in their operations. Their efficiency in spreading among devices connected through networks make it to become effective in attacking several hosts simultaneously. The advantageous attribute they have is that they can change into newer modes of operations hence possessing the characteristics of zero-day attacks. When this happens, there exists some chance that the signatures they have after this process may not be recorded in the repository or database of the platform that may be used as a checker as to whether the new object received is malicious or not. If the signature does not exist, it may be automatically considered as benign item.

In the same venture of zero-day exploits, there are also terms such as exploits and vulnerabilities. When exploits are mentioned, these are generally tools and methods that are used in order to carry out a successful attack on the system(s) that are within the intended scopes. Vulnerability refers to a flaw that is existent on the system that is to be attacked. The flaws can come up due to various reasons. For instance, configurations that may be wrong can lead to the availability of vulnerability in a system. One major challenge tied to zero-day attacks when they propagate into systems successfully is the time that is used in the patching process. This is because a lot of understanding is needed to know how the zero-day operates so that an organization can implement, deploy then test the security solution that it was intended to patch.

2.3.6 Features for Detecting Zero Day Attacks

There are several features of zero-day attacks that can be used in their detection process. One of them is the patterns that the zero-day attacks have. These patterns can be spotted through

their script files, the spywares that they come along with and even Trojans that they use. This is because they can be derivatives of existing viruses that were used before to carry out attacks. The second feature of zero-day attacks that can assist in their identification are the behaviors they take up. For instance, a zero-day attack that carries out a Denial of Service (DoS) will be identified with the behavior of consuming the resources that are used in the devices in a network. The third identification feature is that there are zero-day attacks that take up the signatures of other existing attacks that were initially recorded. This therefore make them not to become fully recognized as zero-day attacks (Singh & Saini, 2014).

2.3.7 Improvement of Zero Day Attacks

Modern day zero-day attacks have been able to improve over the years in order to beat the security strategies that have been put in place in order to mitigate them. The improvements happen due to the advanced exploration of protocols and applications with their specific versions by attackers. From there, custom zero-day attacks can then be implemented based on the inferences that have been obtained. Another major improvement that comes along with modern zero-day attacks is the analysis of patches that have been made recently to cover up the pre-existing vulnerabilities. For instance, the Stuxnet attack was used in uranium plants in Iran. It operated through the use of five identified vulnerabilities. One of the vulnerabilities was patched by Microsoft but the same patch was once again used by hackers as an attack vector (Riofrío *et al*., 2021).

It has been highly observed that the creation of these zero-day attacks has been for specific purposes. For instance, a zero day might be built to attack the systems of a specific organization after their Information Technology (IT) infrastructure has been well understood. That is why they are considered to be attacks with underestimated impact. This is because they were unforeseen and when they attack nobody knows in advance the impact that they would bring (Riofrío *et al*., 2021).

2.3.8 Approaches for Detecting Attacks in Cyber Security

In general, there are two main approaches for identifying zero-day attacks. These are the approaches that use signature and machine learning models. Before selecting the best approach that may be suited for an organization, there are various factors that are usually considered during the selection process. For instance, the cost that is incurred in the processes of development and maintenance should be feasible. Also, the system should be capable of

providing accurate results so that correct decisions can be made (Cardenas, Baras, & Seamon, 2006).

2.3.9 Signature-Based Approaches for Attack Detection

This is mainly considered as the traditional method for anomaly detection. A compilation of processes that happened during various instances are put together into a database. From there, the patterns that can be identified for normal and anomalous behaviors are used to decide as to whether some data is malicious or benign. Then, future transactions that can either be normal or anomalous are compared to the instances that were used as reference points. The absence of new activities' signatures means that a decision cannot be made on whether the action was malicious or benign (Einy, Oz, & Navaei, 2021).

Though the functionality of signature-based approach can be highly acknowledged for assisting in improving the security of systems, it has a low rate of detection for attacks that are based on the concepts of spam. This may call forth for additional security measures such as stemming and noise elimination to improve the effectiveness of the concepts that will be used (Sharma, & Jain, 2015).

2.3.10 Machine Learning Methods for Attack Detection

Machine learning refers to the process of implementing systems that improve their performance when the experience becomes increased. This happens when the algorithms are subjected to some data that they can learn from. From there, they give some output that may be desired by the end user. This concept is also considered to be as a subcategory in the specialization of artificial intelligence. It also involves processes such as fitting of models and inferencing so that they operate in accordance to the objectives that they were designed to achieve. Useful information is obtained from the dataset that is used in the learning process to result in outputs that are mostly probabilistic. This process is a derivative of how the human mind functions (Oladipupoa, 2010).

For purposes of organization, machine learning algorithms are ordered in a taxonomic manner. The main factor that is used in the categorization process is the desired outcome that may be obtained after the algorithm accomplishes its tasks. The first category is supervised algorithms that give an output based on a function that receives data from some input. For instance, supervised algorithms can be used to classify data according to the groups that may be needed. Semi-supervised machine learning algorithms utilize a combination of both labelled and

unlabeled datasets to get an end output that was desired. As for unsupervised algorithms, they use unlabeled data to carry out operations such as clustering. It operates on the basis of finding patterns and regularities based on the statistics that are carried out by the algorithms the data is subjected to (Oladipupo, 2010).

In order to make an effective project, various concepts tied directly to machine learning are used in order to boost the accuracy of the whole procedure. The focus for this scope will be the implementation of a hybrid for supervised machine learning algorithms. The configurations that can be made include the grid search strategy. This involves carrying out a search through the hyperparameter configuration strategies which are standard for the algorithms to be deployed. This is the most rudimentary and brute-force method for hyperparameter tuning. It starts by defining a set of values for each hyperparameter, and grid search will test every probable combination of these hyperparameters. Also, a cross-validation procedure can be carried out to facilitate a more comprehensive training on the set of algorithms to be used. From there, validation metrics to confirm the accuracy of the results obtained from the test set will be carried out. Accuracy metrics that can be used during this phase include F1-score, precision and/or recall. In case the accuracy might be found to be low, further tuning can be done to the algorithms in order to improve the accuracy (Vitorino, Andrade, Praça, & Maia, 2021).

Zhang *et al*. (2008), wanted to detect anomalies using the Support Vector Machines algorithm. In order to carry out the implementation of this process, three different datasets were used to accomplish this mission. The data was obtained from the telecommunication network. As a result, the accuracy of this project was gauged as a good one. In order for the algorithm to become fully operational, it involved the use of a kernel function known as the Radical Basic Function in order to operate. One major challenge with this function is that the parameters used in the process of configuration are very hard to define. Then monotony of this algorithm also limited the evaluation of the performance of the results obtained with other kernels that would be used for the same operations.

According to a project carried out by Limthong (2013), it was also established that there is a possibility of carrying out a hybrid of three machine learning algorithms to achieve the end goal. In this scenario, a one-class Support Vector Machines (SVM) algorithm, multivariate normal distribution and the K-Nearest Neighbor algorithms were used in this process. As for the performance evaluation procedure, the recall and precision methodologies were used in this phase. Though it was established that this anomaly detection strategy worked in good condition

in the real-time instances it was being used, flaws were established within the system. The first one is that the training and testing times for these algorithms were not described. Additionally, improvements were found to be necessary in the methodology that was used for the feature selection procedure in the experiment.

2.3.11 Hybrid Methods for Attack Detection

According to Pandeeswari & Kumar (2015), they implemented a system that takes up the properties of a clustering algorithm and the Fuzzy C-Means and Artificial Neural Network. The clustering algorithm that was being used was Fuzzy C-Means. It was used to identify anomalies that were being identified at the hypervisor layer. The system that was implemented was named the Hypervisor Detector. The main role of the neural network was to facilitate the improvement of the accuracy of the algorithm. The main dataset that was used in the experimentation phase was the DARPA KDD Cup Dataset 1999 (MIT Lincoln Laboratory, 2023). One of the main items that were identified in this process was that the algorithm combination had a high accuracy. In addition to that, it had a low false negative and false positive rate. It was also noted that the algorithm had great performance as compared to singular machine learning algorithms such as Naïve Bayes.

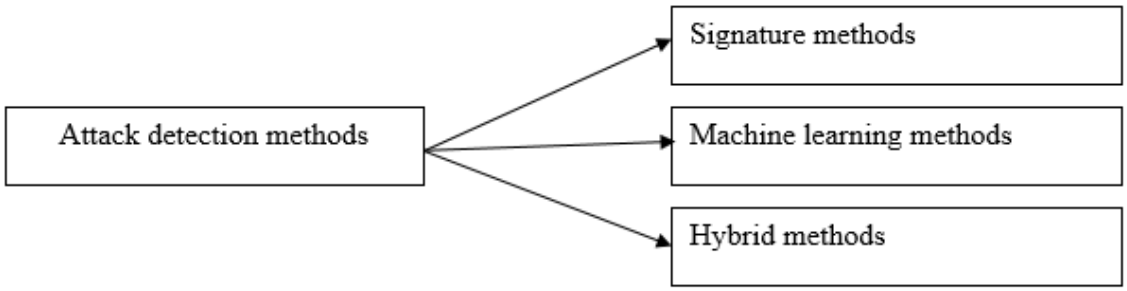


Figure 2.3: Methods for detecting attacks

2.3.12 Challenges in Detecting Attacks in Cyber Security

There is a huge number of challenges when it comes to both approaches. For instance, the signature-based systems are inefficient when it comes to the detection of attacks. This can be generally attributed to the fact that attackers consistently try to find strategies to evade detection by these systems. If the new attack is not counted as a signature in the repository where the scans are being fetched from, the packet will be considered as benign. On the machine learning side, there have been a lot of exploits that can be acknowledged to attack and corrupt machine

learning techniques that were originally intended to efficiently spot the attacks that would be transmitted to systems (Shaukat *et al*., 2020).

Secondly, it is important to take into account the fact that algorithms such as Apriori can be used to detect anomalous data being transmitted. It works on the basis of the identification of the properties of the attack component in the data. Though it can be an efficient strategy, it consumes a lot of time before a confirmation is given as to whether a packet is malicious or benign. This means that redundancy will be highly encountered in the system that is used to facilitate the security that needs to be enforced fully (Modi *et al*., 2013).

With regards to the above scenario, a lot of effort has been put to reduce the amount of time signatures are scanned in databases. This has led to less downtime encountered during scans for signatures. Though time has been reduced, it has been noted that a lot of false positives have been encountered in the checking process. The reason behind this is that a number of important patterns become ignored. In addition to that, some patterns that are not required become produced hence polluting the integrity of the operation of the scanning reduction algorithm that is being used (Modi *et al*., 2013).

The next challenge occurs with regards to machine learning algorithms that work on the basis of the data they are been exposed to during the training phase. If these algorithms are subjected to datasets with a different structure from the one that they did not experience during the training phase, they will not be able to give concrete results as to whether some data is malicious or benign. Though that is the case, some neutralization to this issue has been brought up by deep learning algorithms. It is worth noting that they have the ability to learn from nonlinear data and identify anything anomalous tied to it. They involve the utilization of concepts such as parameter sharing so that the complexity of the model can become reduced (Ibrahim *et al*., 2023).

Thirdly, the new attacks that can be launched to systems are may not be in their databases and may be considered as benign traffic that is in transit to and from the systems being secured.

2.3.13 Trends in Machine Learning and Cyber Security

In the merging of cyber security and machine learning algorithms, a lot of projects have been carried out in order to solve specified problems. Zhuang, Ye *et al* (2012), were trying to use hybrid machine learning algorithms to detect phishing attacks. This process involved the use of clustering algorithms that worked together to jointly perform the same function. The feature

selection strategy was used to obtain emails that had malicious traits. One of the algorithms that was used in the full deployment was Hierarchical Clustering (HC). This algorithm used unlabeled data to facilitate the clustering process. It was recorded that the algorithm had an 85% success rate. A major drawback of this project was the lack of clarity on whether all devices from mobile to desktop applications are able to access this service.

Another great project that was implemented involved the use on the Support Vector Machines (SVMs) and Neural Networks (NN) in order to form a hybrid algorithm. This combination was able to be used as a building point to notify the people involved about probable attacks that may happen. The main attack that was put in the limelight for the countering process was the Distributed Denial of Service (DDoS).

2.3.14 Summary of Literature and Research Gaps

The table below summarizes the gaps that have been identified in the research materials that have been used within the scope of this study:

AUTHORS	PAPER	GAP
Holm H. (2014)	Signature Based Intrusion Detection for Zero-Day Attacks: (Not) A Closed Chapter?	The implementation had a low accuracy of 17%.
Li, Z., Qin, Z., Shen, P., Jiang, L. (2019)	Zero-shot learning for intrusion detection via attribute representation. International Conference on Neural Information Processing, pp. 352–364, Springer.	The project had an accuracy of 34%.
Onik, A. R., Haq, N. F., & Mustahin, W. (2015)	Cross-breed type bayesian network-based Intrusion Detection System (CBNIDS). 2015 18th International Conference on Computer and Information Technology (ICCIT).	The implementation of the algorithms for this study had an accuracy of 84.98%
Kaur and Singh, 2014	). An empirical evaluation of classification algorithms for fault prediction in open-source projects.	Was not able to identify polymorphic worms

	Journal of King Saud University - Computer and Information Sciences, *30*(1), 2–17	
Riofrio *et al*., 2021	The Zero-day attack: Deployment and evolution. *Zenodo (CERN European Organization for Nuclear Research)*, *8*(1), 39–53.	Did not clearly explain the suitability of the dataset to the project
Einy, S., Oz, C., & Navaei, Y. D. (2021).	The anomaly- and signature-based ids for network security using Hybrid Inference Systems. *Mathematical Problems in Engineering*, *2021*, 1–10.	Low detection rate for spam
Limthong, 2013	Real-time computer network anomaly detection using machine learning techniques. In Journal of Advances in Computer Networks, volume 1(1).	Did not outline the processing speed for the results
Ibrahim Hairab, B., Aslan, H. K., Elsayed, M. S., Jurcut, A. D., & Azer, M. A. (2023)	Anomaly detection of zero-day attacks based on CNN and Regularization Techniques. *Electronics*, *12*(3), 573.	The algorithms were not able to support a dataset with a different structure.
Zhuang, Ye *et al*., (2012)	Hybrid clustering for internet security applications. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(6), 1784–1796.	Did not outline whether it supports different kinds of software such as desktop and web
Holm, H (2014).	Signature Based Intrusion Detection for Zero-Day Attacks: (Not) A Closed Chapter? IEEE Xplore.	Low detection rate of 17%

Hindy *et al*. (2020)	Utilising deep learning techniques for effective zero-day attack detection', *Electronics*, 9(10), p. 1684.	False alarms were not outlined in the project

Table 2.1: Summary of literature and research gaps

A review of the existing literature on zero-day attack detection reveals several notable limitations that hinder the effectiveness of current systems. Holm (2014) examined signature-based intrusion detection systems for zero-day attacks and reported a detection accuracy of only 17%. This underscores a critical limitation of relying on predefined signatures, as such systems are inherently unable to identify novel or previously unseen threats. Similarly, Li *et al*. (2019) explored zero-shot learning techniques for intrusion detection, achieving an accuracy of 34%. While this approach attempted to address the challenge of unseen attacks, its low performance highlights the need for more reliable and adaptable detection mechanisms.

Onik, Haq, and Mustahin (2015) implemented a cross-breed Bayesian network intrusion detection system (CBNIDS), achieving an accuracy of 84.98%. Although this represented an improvement over previous methods, the approach still exhibited constraints in handling diverse attack types and evolving threats. Kaur and Singh (2014) evaluated classification algorithms for fault prediction but noted that their system was unable to detect polymorphic worms, demonstrating the difficulty of managing highly variable attacks. Similarly, Riofrio *et al*. (2021) reviewed the deployment and evolution of zero-day attacks but did not clearly explain the suitability of the dataset used, raising concerns about the generalizability and applicability of their findings.

Other studies have highlighted additional challenges. Einy, Oz, and Navaei (2021) explored hybrid anomaly and signature-based intrusion detection systems but reported low detection rates for spam, indicating that certain attack types remain inadequately addressed even with hybrid approaches. Limthong (2013) presented a real-time anomaly detection system using machine learning techniques but did not provide information on processing speed, which is critical for evaluating real-time applicability. Ibrahim Hairab *et al*. (2023) applied CNNs and regularization techniques for anomaly detection but found that their algorithms could not support datasets with different structures, limiting flexibility for diverse real-world data. Zhuang *et al*. (2012) proposed a hybrid clustering approach for internet security applications but did not specify whether the method supports different software platforms, such as web and desktop applications, which reduces its practical applicability. Finally, Hindy *et al*. (2020)

implemented deep learning for zero-day detection but did not report false alarm rates, leaving uncertainty regarding the reliability of the system in practical deployment scenarios.

2.3.15 Strengths and Weaknesses of the Existing Zero Day Detection Algorithms

According to Holm, H (2014), he proposed the utilization of a signature-based Network Intrusion System. One of the main aims of his research was to prove that these systems can be used to detect zero-day attacks. In the research, a total of 356 network attacks were used in the process. Out of the 356, a total of 183 attacks were zero-days. To identify the attacks using the signature method, Snort was used. The Metasploit framework was then used to generate the simulation attacks. The strength of this research was in the seamless setup when it comes to the simulation. Unfortunately, the tool had a low detection rate of 17%.

According to Hindy *et al*. (2020), his research was mainly aimed at curbing the high False Alarm Rate (FAR) that is usually produced by systems that are used to detect zero-day attacks. To solve this problem, an autoencoder was used to detect the zero-day attacks and still maintain a low False Alarm Rate (FAR). The classifier algorithms that were used in this process utilized benign data for the training process. All the phases of this project obtained an accuracy of between 75-98%. Though the accuracy was good, the algorithm did not consider outlining of the false alarms generated and the attacks that were undetected in that process.

Li, Z. *et al*. (2019) focused on the use of attributes to detect zero-day attacks. They used the Random Forest (RF) strategy for the feature selection of features that would be used in their machine learning algorithm. The process of spatial clustering was used to facilitate the conversion method. The data that was used in the phases of the machine learning pipeline were converted into unsupervised cluster attributes. Through the process, there was a low detection rate of the specific attacks that it was subjected to. For instance, Denial of Service (DoS) and probe attacks had a low detection rate of 34.71%.

Onik *et al*. (2015) did an experiment that was tied to the utilization of several algorithms on the NSL-KDD dataset. The dataset was composed of 41 features and 25,192 records. It was also made up of labels such as normal packets, U2R, DoS, R2L and probe attacks. In order to have a high efficiency of the algorithm, it was necessary for them to reduce the features that they have in the dataset from 41 to 16. The method that was used in this process was the filter approach. The models that were used in this experiment were K-means clustering, Naïve Bayes, RBF network and decision stamp that had accuracies of 80.75%, 84.86%, 91.03% and 83.31%. On average, the four combined algorithms had an accuracy of 84.98%.

2.4 Conceptual Framework

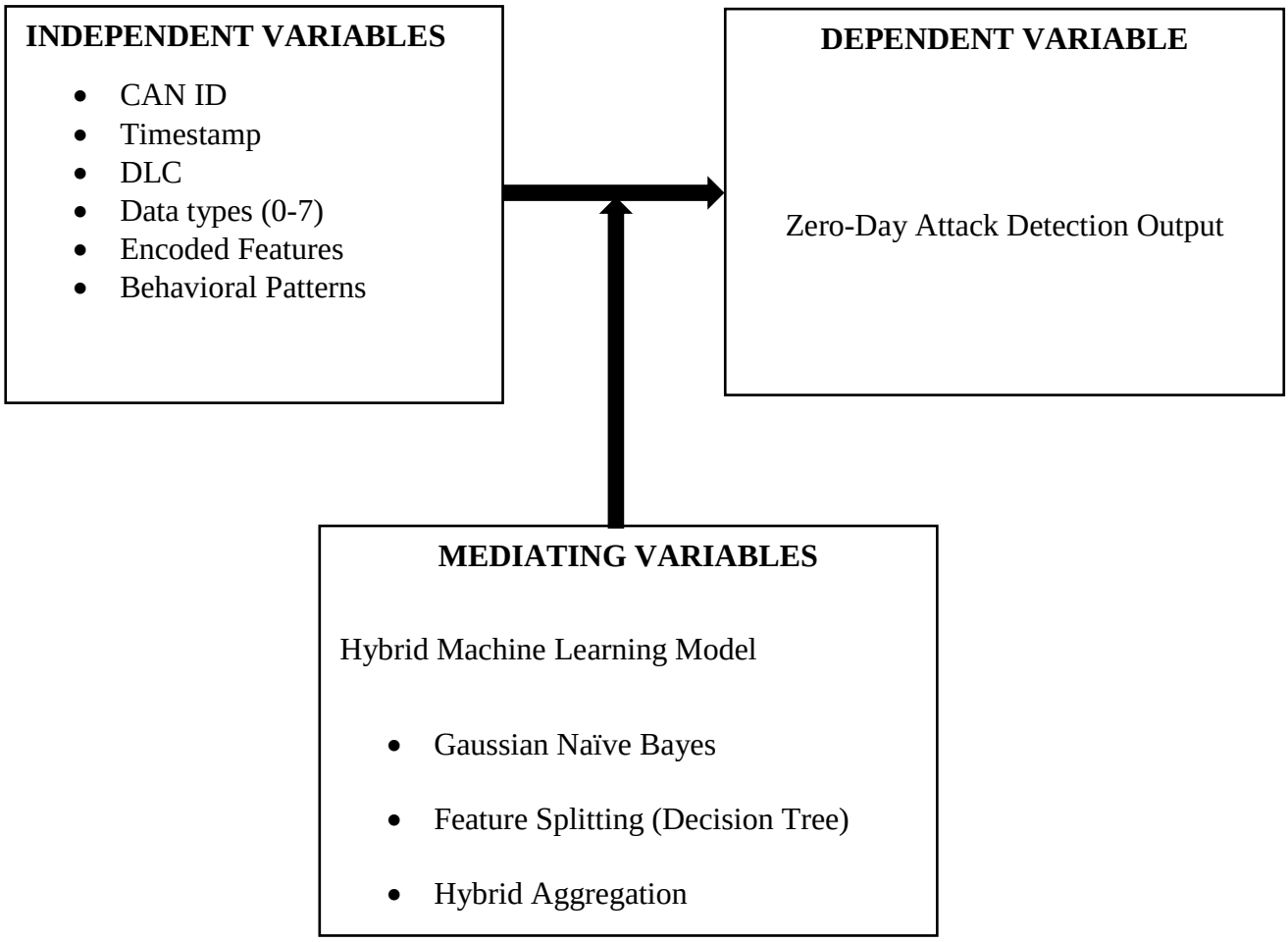


Figure 2.4: Conceptual Framework

The independent variables consist of a set of network traffic attributes extracted from the CAN bus dataset, which collectively provide the raw inputs that enable the machine-learning model to distinguish between normal and malicious activity. These include the CAN ID, which identifies the source or type of each message on the network, the timestamp that records when each message is transmitted, and the Data Length Code (DLC) that indicates the size of the payload. The data bytes themselves, which range from byte 0 to byte 7, carry the actual command values exchanged between vehicle components and often reveal abnormal modifications during an attack. In addition to these raw features, encoded attributes generated during preprocessing such as label-encoded or transformed variables help the model interpret categorical patterns more effectively. The study also incorporates behavioral patterns derived from the dataset, including variations in message frequency, abnormal repetition, or timing inconsistencies, which frequently signal zero-day intrusion attempts.

These independent variables are processed through the mediating variable, which is the hybrid machine-learning model combining Gaussian Naïve Bayes (GNB) and a Decision Tree classifier. The hybrid model plays a central role by transforming the raw input features into meaningful predictive insights. The GNB component contributes probabilistic estimations that help determine how likely a message belongs to a benign or malicious class, while the Decision Tree classifier provides rule-based feature splitting using entropy and information-gain criteria. The model's hybrid aggregation mechanism strengthens prediction reliability by merging outputs from both algorithms, thereby reducing misclassification and improving the stability of the detection system.

The dependent variable of the study is the zero-day attack detection outcome, which represents the final classification produced by the hybrid model. This outcome indicates whether a given CAN message is malicious or benign and reflects the model's overall effectiveness in identifying previously unseen attack patterns. It also embodies key performance improvements such as higher detection accuracy, reduced false-positive and false-negative rates, and a more robust identification of anomalies within CAN bus traffic.

CHAPTER THREE

3. RESEARCH METHODOLOGY

3.1 Introduction

This chapter covers the strategy that was followed during the implementation of the project together with the requirements that are tied to it. This formed the foundation of the workflow that was used to solve the problems that need to be solved through the carrying out of this project. For the measurement of variables, an analytical description of how they contribute to the end result that was also carried out. Lastly, the data collection, analysis and presentation process were described as the final item.

3.2 Research Design

The study takes up the design science model. This is because it involved the creation of a hybrid model. Also, the same model was evaluated scientifically through the use of metrics such as F1-score, accuracy score, recall, precision and other metrics. The main strategy used in this project was the machine learning pipeline. This workflow was based on the Python programming language. It is composed of various stages that work together to solve the problems at hand. It was assumed that the data that was fed into the system was raw and unprocessed. After being loaded, it was preprocessed and transformed in order to be in a standard structure for subjection to the next processes in the machine learning cycle. The data was then be subjected to the process of feature engineering to prepare it for utilization in the training of the hybrid machine learning algorithms (d'Aloisio *et al*., 2022). The main feature engineering strategy that was used was label encoding. It involves converting all the values in the dataset into numerical format (Olson & Moore, 2018).

When the model was ready for subjection to the machine learning process, a hybrid of several algorithms was used. The algorithms that were used in the implementation of the hybrid were decision trees classifier and Gaussian Naïve Bayes (Olson & Moore, 2018). From there, some model evaluation strategies were used to evaluate the accuracy of the hybrid algorithms with regards to the provision of high accuracy. If the accuracy might be found to be low, further improvements would be done on the algorithm through the use of hyperparameter tuning strategies. Lastly, the machine learning algorithm was also monitored in order to ensure that it has a consistent record of correct results (d'Aloisio *et al*., 2022).

3.3 Empirical Model

The empirical model developed in this study was a hybrid that combined Gaussian Naïve Bayes (GNB) and Decision Tree (DT) classifiers to enhance the detection of zero-day attacks. The model worked with a set of independent variables drawn from network traffic and system behavior, including flow statistics, protocol and service information, connection flags, and behavioral patterns such as request rates and packet frequency. Before training, these features were cleaned, encoded, and normalized so both algorithms could interpret them correctly. The output variable was a modest binary label that determined whether each entry represented malicious activity, comprising behavior that could signal a benign or normal traffic. In the hybrid, GNB contributed probability-based predictions that stemmed from how features were distributed in each class, while the DT added concise, rule-based decisions built using entropy and information-gain calculations. The results from both models were then combined, either through majority voting or probability averaging, to create a more stable and accurate prediction than either model would produce on its own. To assess performance, the model went through a structured validation process using either a train test split, and was evaluated using accuracy, precision, recall, F1-score, and confusion matrix metrics. The final step compared the hybrid model's performance to the individual GNB and DT classifiers to confirm that the hybrid offered better generalization and stronger zero-day detection. This was followed by an error analysis to further refine the model and ensure it remained dependable when identifying both known and previously unseen attacks.

The hybrid model combination makes them to be a step ahead to be able to complete with other advanced algorithms such as neural networks They have a good level of accuracy and they tend to have lower processing time as compared to neural network algorithms. This makes them to become greatly adaptive in both research and real time analytics Information Technology (IT) infrastructure. In general, accuracy can be defined as the relationship between the total number of predictions that are correct with regards to the total number of data points being used. Naturally, the metric measures for the accuracy of the algorithms that can be used in the machine learning process can be measured through various strategies such as f-score, precision, recall and Receiver Operating Characteristic (ROC) can be used.

From the research done by (Kaur & Kaur, 2018), the Random Forest algorithm had the highest accuracy as compared to the other algorithms when compared through the Receiver Operating Characteristic (ROC) accuracy metric algorithm. This comparison was done against other classifiers such as Bagging, logistic regression, Naïve Bayes, J48 and IBI. The accuracy

obtained in this process was found to be about 97% for the accuracy were used. It is also believed that because of the strength of that algorithm, it would be able to perform properly when used in the hybrid that will be deployed.

3.4 Target Population

The target population for this study comprised the complete set of Controller Area Network (CAN) bus messages generated within modern automotive communication systems. This population included all raw network frames containing identifiers, timestamps, data length codes, and payload bytes that collectively represent the operational behaviour of embedded vehicular components. Since zero-day attacks often manifest through subtle manipulations of these message structures, the full range of CAN traffic formed the ideal population from which meaningful patterns of both normal and malicious activity could be analysed. By focusing on the entire message ecosystem rather than isolated samples, the study ensured that the developed detection model captured the true diversity and complexity of real-world automotive communication flows.

Additionally, the target population included cybersecurity datasets and recorded CAN logs commonly used for intrusion-detection research, particularly those containing instances of simulated attacks, anomalous sequences, and irregular message patterns. These datasets provided a broad representation of potential zero-day behaviours, enabling the hybrid machine-learning model to learn from a population that reflects both benign operational traffic and sophisticated intrusion attempts. This approach ensured that the model was trained and evaluated using a population large enough and varied enough to support accurate detection, enhanced generalization, and reliable real-time performance when deployed in actual vehicular environments.

3.5 Dataset Description

The dataset used in this process is targeted towards investigating cyber-attacks that can be relayed to automotive networks in vehicles. The dataset used is the Automotive Controller Area Network (CAN) Bus Intrusion Dataset V2. The cars that were used to collect this dataset were Renault Clio and Opel Astra. The bus that was used was a prototype that they built on themselves. There was a mixture of malicious and benign data that was collected. The malicious ones were mainly attacks such as replay attacks, fuzzing, suspension, Denial of

Service (DoS) and diagnostic attacks. The dataset was first published by the 4TU.Centre for Research Data (Dupont, Lekidis, Hartog, & Etalle, 2019).

The dataset is composed of features such as data value represented in bytes of range zero to seven, timestamp, Controller Area Network (CAN) identifier, Data Length Code (DLC), the classes and subclasses for the data. For the actual structure of the data to be obtained, it was important to reverse the preprocessing procedures that were used. When this was done, the volume of the dataset increased by a huge margin. In return, this lowered the speed used to process the data through the hybrid machine learning algorithms that were used. Through the process of feature extraction transformation from the original dataset, time used to execute the algorithms and system complexity was reduced by a very huge margin (Bari *et al*., 2023).

In order to carry out the proper comprehension of the happenings that took place, messages that were transmitted were analyzed. Cumulatively, these patterns were used to identify the truthfulness of whether a data point will be a malware or a normal information transmission medium. Another item targeted through the analysis of the dataset that was used in this study are the faults that were recorded according to the timestamps and messages recorded. Lastly, these faults led the establishment of malfunctions that were recorded on the vehicles through the zero-day or existing attacks that will be recorded (Dupont, Lekidis, Hartog, & Etalle, 2019).

There were several reasons that led to the selection of the dataset. From its structure, it had well labelled attack and normal data point. The quantity of the data that was availed for the implementation of the project was adequate to facilitate the whole process throughout its phases. The dataset also has zero-day attacks that provide a good learning point for the hybrid machine learning algorithms that were used in the process. It also has a good variety of attacks that provide a good learning framework for the provision of accurate results of the predictions that were made. Due to the nature of its content, it is also relevant because it highly resonates with attacks that are usually encountered in today's technological landscape.

3.6 Sampling Design

According to the context of the dataset that was used in this project, the random sampling strategy was used in this process in order to obtain a portion of the dataset that will be used in the machine learning process. Though it is random, the same rows will be obtained once the code will be iteratively run. This is because of the *random_state* module in Python that lets everything remain in a constant state no matter the number of times the same code is executed again and again. The reason behind the use of this method is that the dataset has about 90%

benign data and 10% malicious data. Through the random sampling strategy described, a sample of the benign data was combined with the malicious data. This reduces the bias that would make the hybrid algorithm to dominantly learn on the benign data as compared to the malicious one.

The datasets were in generally two different formats. The first format is as sampled in the table below. In this format, it contained the timestamp when the data was captured, the arbitration ID, DLC value, the data and the class used. In this scenario, the class that was only available was that of the normal packets. This represents the data that did not have anything malicious in them:

Timestamp	Arbitration_ID	DLC	Data	Class
1.60E+09	153	8	20 80 10 FF 00 FF 60 0E	Normal
1.60E+09	520	8	00 00 00 00 00 00 00 00	Normal
1.60E+09	4A9	8	00 00 00 00 00 00 00 00	Normal
1.60E+09	164	4	00 08 1C FA	Normal
1.60E+09	356	8	00 00 00 80 1A 00 00 00	Normal

Table 3-1: Sample data without the malicious records

The second dataset hand one main difference from the initial dataset. This was the fact that it had an extra column which was that of subclass. This subclass represents the attacks that were recorded during the data collection process for the vehicles that were being used.

The categories that were available in the subclass section of the dataset were normal data, replay, fuzzing, flooding, flooding and spoofing attacks.

Timestamp	Arbitration_ID	DLC	Data	Class	SubClass
1.60E+09	453	5	00 88 8E 00 94	Normal	Normal
1.60E+09	356	8	00 00 00 80 13 00 00 00	Attack	Replay
1.60E+09	394	8	00 80 08 00 DC 33 10 FC	Normal	Normal
1.60E+09	44E	8	BC67AFC B9 7D 2B BF	Attack	Fuzzing
1.60E+09	4F1	4	20 00 60 21	Normal	Normal

Table 3-2: Sample data with the malicious records

As described earlier on, the datasets that were used in this project were six. The table below summarizes the structure of the datasets with regards to their number of rows and columns:

Dataset	No. of rows	No. of columns	Nature
Pd0	178346	5	Normal
Pd1	806390	6	Normal
Pd2	889395	6	Normal
Ps0	180686	5	Normal
Ps1	799292	6	Normal
Ps2	817042	6	Mixed

Table 3-3: Description for each imported dataset

3.7 Data Collection Instruments

This research utilizes secondary data sourced from online repositories. This is because the data had been collected, processed and published. The primary limitation in using primary data for this study stems from the inherent difficulty of obtaining real-world zero-day attack data. Zero-day attacks are, by definition, previously unknown exploits, and organizations are understandably reluctant to share live attack logs due to confidentiality, legal, and security concerns. Collecting such data in a controlled environment would require generating realistic zero-day scenarios, which is resource-intensive, time-consuming, and may not fully capture the complexity and diversity of attacks in operational networks.

The internet was chosen as the methodology for data collection because:

1. It has combinations of both malicious and benign properties of packets in a dataset.
2. There is a wide range of datasets available online that would be used to achieve the same objectives
3. Further research can be done with regards to the works done by other people to solve various Information Communication and Technology (ICT) problems through the use of the dataset.

The validity of the dataset collection strategy used is very high. This is because the dataset is well suited to achieve the objectives this study is meant to achieve. It is composed of labels that would be dropped and act as the learning point for the hybrid algorithm to be developed. As for the features that will be used, there exists a lot of ways of analyzing and preparing them for being subjected to the machine learning algorithms that will be used.

3.8 Data Collection Procedure

The procedure that was used in the collection of the data is as follows:

1. Understanding of the problem to be solved in order to identify a dataset that would be used to counter them
2. Identification of online platforms that may have datasets tied to solving the objectives we have
3. Listing down the datasets that have the potential of achieving the objectives
4. Brief Exploratory Data Analysis of the datasets that have been identified
5. Selection of the most suited dataset that would be used in the project

3.9 Data Analysis and Presentation

The main form of analysis that will be used in this project is quantitative data analysis. This is because the data is factual hence numerical statistical conclusions can be made out of it. The following descriptive statistical concepts will be used in understanding the data:

1. Number of missing values
2. Outliers in the dataset
3. Distribution of values in the dataset
4. Quantity of values in the dataset
5. Correlations between columns in the data

The main method of presentation of the data was data visualization. This was mainly done through graphs and plots. In return, this enables us to get more understanding of how the data is.

3.10 Hardware Resources

A device such as a computer with the following specifications are needed:

1. At least 15GB free internal storage
2. At least 4GB RAM
3. An i5 or any other processor with over 1.8GHz

3.11 The Machine Learning Pipeline

The machine learning pipeline is the main strategy that is used in this study. It uses the following steps in order to fully achieve the objectives at hand:

1. **Understanding the problem** – This involves comprehending the issue to be solved.

Without knowledge of the underlying issues, there may be no guarantee of a good workflow of the project or obtainment of the best outputs to yield correct decisions for future processes.

2. **Obtaining the data** – The data can be obtained through several ways such as:

a) Carrying out surveys with relation to the findings that need to be obtained

b) Looking for open-source datasets relating to the project to be undertaken

c) Paying for a preexisting/to be collected dataset from another third party that fits into the project to be solved.

In the scenario of this project, the second option is most preferred due to the following reasons:

a) A lot of datasets related to the project can be obtained

b) It is an easy filtering process to obtain the dataset that is best fit for the project

c) It is a cheap alternative compared to the latter options

d) Most open-source datasets have an easy-to-understand description that would be a

reference point to understand how to answer the questions that may be there.

3. **Exploratory data analysis** – This refers to the statistical activities that can be done on the dataset to answer various questions that would help in understanding the data and in decision making. A great feature in the process of Exploratory Data Analysis is the flexibility to make visualizations that would be easily understandable by the human eye.

4. **Cleaning the data** – The data cleaning process assists in improving the quality of data in order to optimize it for further analysis and making predictions where necessary. The processes that may be used to improve the cleanliness of data include handling of missing values, duplicate data and categorical data.

5. **Feature engineering -** The feature engineering process refers to steps that are used in the transformation of raw data into usable ones for the machine learning algorithms to be used. In instances such as classification machine learning algorithms, there are outcome and predictor variables. In this instance, the data is be prepared to go through a hybrid machine learning algorithm. The main feature engineering procedure that was used is the utilization of the label encoding process. This involved the conversion of the whole dataset into a numerical format that would be understandable by the hybrid model being used. Without this conversion, the model would not be able to learn and give results.

6. **Subjection to the hybrid algorithm** – The prepared datasets is then subjected to an hybrid algorithm in order to identify whether they are malicious or benign.

7. **Getting the results of the machine learning algorithm –** The outputs of the algorithms are obtained and then mapped to the original dataset. Further analysis and decision making are carried out after this process. This mapping is then done by fixing the results obtained from the hybrid algorithm to the datasets that was be used in the project.

8. **Testing of accuracy of the algorithm** – The main testing strategy that is used in this process are the inbuilt accuracy metrics algorithms in the Python programming language.

9. **Hyperparameter tuning of the algorithm –** This involves boosting the accuracy of the hybrid algorithm to obtain better performance of the results. Since the model had a high accuracy of over 97%, it was not subjected to any hyperparameter tuning strategies.

10. **Maintenance –** Customizing the implementation of the algorithm to suit other scenarios that may need the same objectives to be solved.

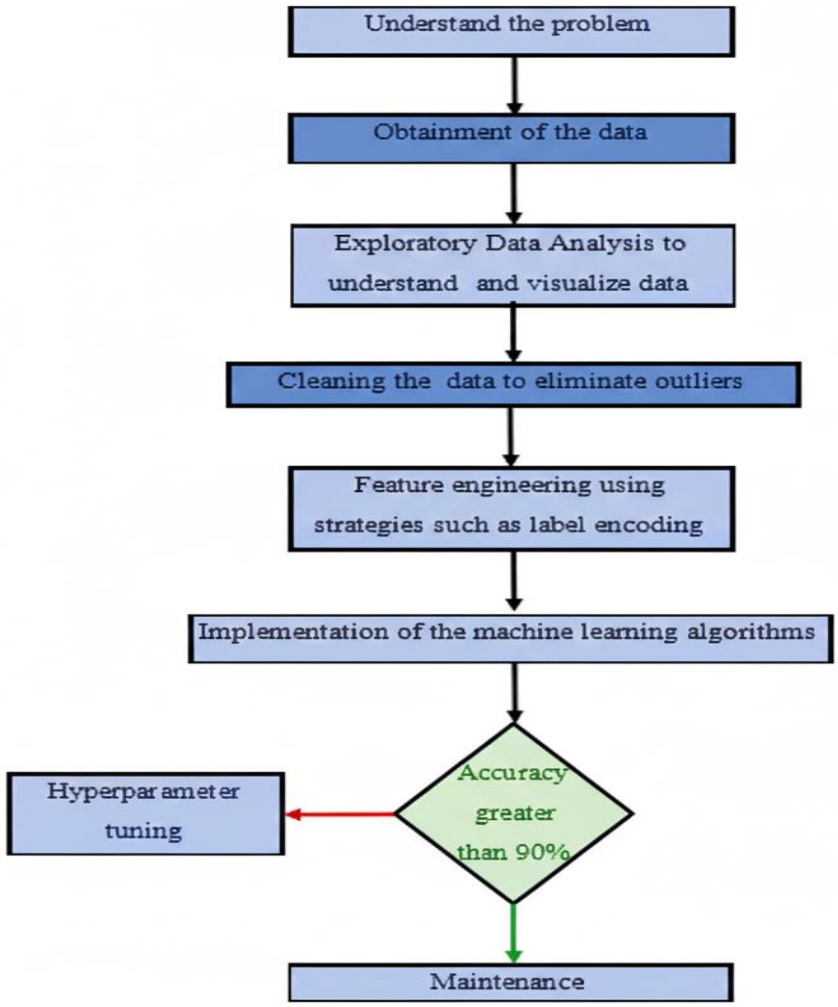


Figure 3.1: The machine learning pipeline

3.12 Ethical Considerations

The research was based on the researcher's original ideas, while duly crediting to any external concepts which are converted with text by acknowledging their origin and contributions to the study. As for the data, which we obtained from 4tu, was not self-created, and did not have any personal identifiers that could be traced to a certain person. There is no personal data that has been captured in the dataset. The data was free and open source and freely available for further research. A license was also obtained from NACOSTI to carry out the research.

3.13 Computational Overheads

It has been noted that machine learning algorithms and signature-based algorithms that are used to detect malware encounter a lot of computational overheads due to the processing involved. There may be excessive utilization of computation time, bandwidth, memory and many other resources for the processing of various phases in the machine learning pipeline that will be

used. It is highly recommended to have a strong laptop for processing. If a laptop is not available, cloud processing platforms for machine learning can also be used (Sayadi *et al*., 2018). Feature Engineering

The process of feature engineering involves the conversion of the whole dataset into numerical format. The data is transformed into this format because most of the machine learning algorithms prefer the utilization of numbers during their training process. The strategy that was used in feature engineering for this project is label encoding. The operation is done on a duplicate of the dataset that was being used in the analytics phase of this project. This is through a for loop that converts everything that is not an integer into the integer format. This includes strings, floats, Booleans, objects and any missing values. After that preparation procedure, the end results are saved into a Microsoft Excel Comma Separated Values (CSV) file for future utilization. The process is demonstrated in the appendix *figure 0.3*:

3.13.1 Machine Learning

The machine learning code below was used in the contribution of the implementation of the full algorithm for the operation of the project. It was used a as a classification solution that groups the data fed in accordance to the criteria of whether it was malicious or not. The process involved the utilization of the Decision Tree Classifier algorithm and the Gaussian Naïve Bayes algorithm for the implementation of the hybrid algorithms. The choice of Decision Tree (DT) and Gaussian Naïve Bayes (GNB) as the constituent algorithms in this hybrid model is deliberate and aligned with the goals of this research, despite these classifiers not being considered the most complex machine learning models. Each algorithm brings complementary strengths that make them particularly suitable for the detection of zero-day attacks. Gaussian Naïve Bayes is a probabilistic classifier that excels in high-dimensional feature spaces and can efficiently handle the uncertainty of zero-day attacks. Its conditional independence assumption simplifies computations, allowing it to generate fast probabilistic estimates of whether network activity is malicious or benign.

While this assumption can be limiting, in a hybrid, it is balanced by the inclusion of DT, which captures feature interactions. Decision Tree classifiers are rule-based and non-parametric, capable of modeling complex, non-linear relationships between input features and attack labels (Oluwasanmi & Ayeni, 2023). DTs are highly interpretable and can capture interactions among network traffic features that GNB may miss. Their main drawback overfitting to noisy data is mitigated by the hybrid's probabilistic component. By combining GNB and DT in a hybrid,

the model leverages probabilistic reasoning from GNB and structured decision rules from DT, creating a complementary system where the weaknesses of one algorithm are counterbalanced by the strengths of the other (Oluwasanmi & Ayeni, 2023). This combination improves the model's ability to detect both known and previously unseen (zero-day) attacks, reduces false positives and false negatives, and maintains computational efficiency suitable for real-time detection scenarios.

Compared to more complex classifiers such as Support Vector Machines (SVMs), Random Forests, or deep neural networks, DT and GNB are lightweight and computationally efficient, making them suitable for real-time detection environments. While SVMs and deep learning models often achieve higher accuracy, they require extensive hyperparameter tuning, longer training times, and large labelled datasets, which can be challenging for zero-day detection where data is dynamic and partially unlabelled (Oluwasanmi & Ayeni, 2023). The hybrid DT–GNB hybrid provides a balance of efficiency, interpretability, and complementary strengths, ensuring robust detection of both known and previously unseen attacks.

The data that was being used was the one that was encoded during the feature engineering stage and then saved into a Microsoft Excel Comma Separated Value (CSV) file. The data is split into two parts, *x* and *y* that are be used in the training and prediction processes. These split portions would be subjected to the *train_test_split* algorithm procedure that was described earlier on. The specific configurations that were made in this phase was to have the data being used for training as 70% and that used for testing as 30%. The test size is what is being described in the code as *test_size = 0.3*. The remaining portion which is 70% of the code would then be used for the training bit of the process. This is what would assist in the yielding of the results to confirm whether we have been able to solve the problems specified in the objectives or not. To ensure that no data leakage occurred, several steps were followed. First, all preprocessing transformations such as label encoding and feature formatting were applied *before* the split and then fitted only on the training set**,** ensuring that the test set did not influence the learned patterns. Second, the dataset was shuffled prior to splitting to eliminate any temporal or structural patterns that could cause correlated samples to appear in both sets. Third, no feature engineering step involved information from the test set; the test data remained untouched until the final evaluation stage, guaranteeing that the model was assessed on completely unseen samples. The code used in the machine learning process is as shown in *figure 0.4* and *figure 0.5* in the appendix.

3.13.2 Accuracy Measurement

The second part of the machine learning code is as shown below. In this section, the data is subjected to both the hard and soft hybrid classifiers that were described earlier on. After that, they are subjected to the F1-Score accuracy metrics algorithm that was described earlier on. In this scenario, two of the high potential strategies were used. Both the hard and soft voting classifier algorithms were placed on the F1-Score scale to determine which one of the two is more accurate than the other. Both of the algorithms were subjected to the x-train and the y-train portions of the dataset so that they can be able to learn. After the learning process both of the algorithms are subjected to the testing phases with the y-test portions of the dataset.

The F1-Score for the hard-voting classifier and the soft voting classifier had their accuracies calculated to the sixth decimal point as shown in *figure 0.5* in the appendix.

CHAPTER FOUR

4. DATA ANALYSIS, PRESENTATION AND INTERPRETATION

4.0 Introduction

In this section, the findings that were obtained in the course of the implementation are outlined. This step is based on the machine learning that was described earlier on. This is based on the Controller Area Network dataset that is used in the course of the project to get the is malicious or benign traits. The procedure used involves obtaining and loading the data, cleaning it, carrying out Exploratory Data Analysis (EDA) on it, then followed by feature engineering, implementation of the hybrid machine learning algorithms and then testing its accuracy. In case the accuracy is lower than the expected standard, it would be important to subject the data to some Hyperparameter tuning procedures until all the requirements are met.

4.1 Implementation

4.1.1 Environment Setup

Since the practical implementation of this study was done using the Python programming language, it was important to set up the environment in order to carry out the operations. The first process involves the downloading and installation of the Anaconda software. In the software, there is a program known as the Jupyter Notebook that is usually used to implement the code and provision of the necessary environment requirements for the code. The interface that is used to launch the application can come in two formats. The first format is the Command Line Interface option that is as shown in the screenshot below:

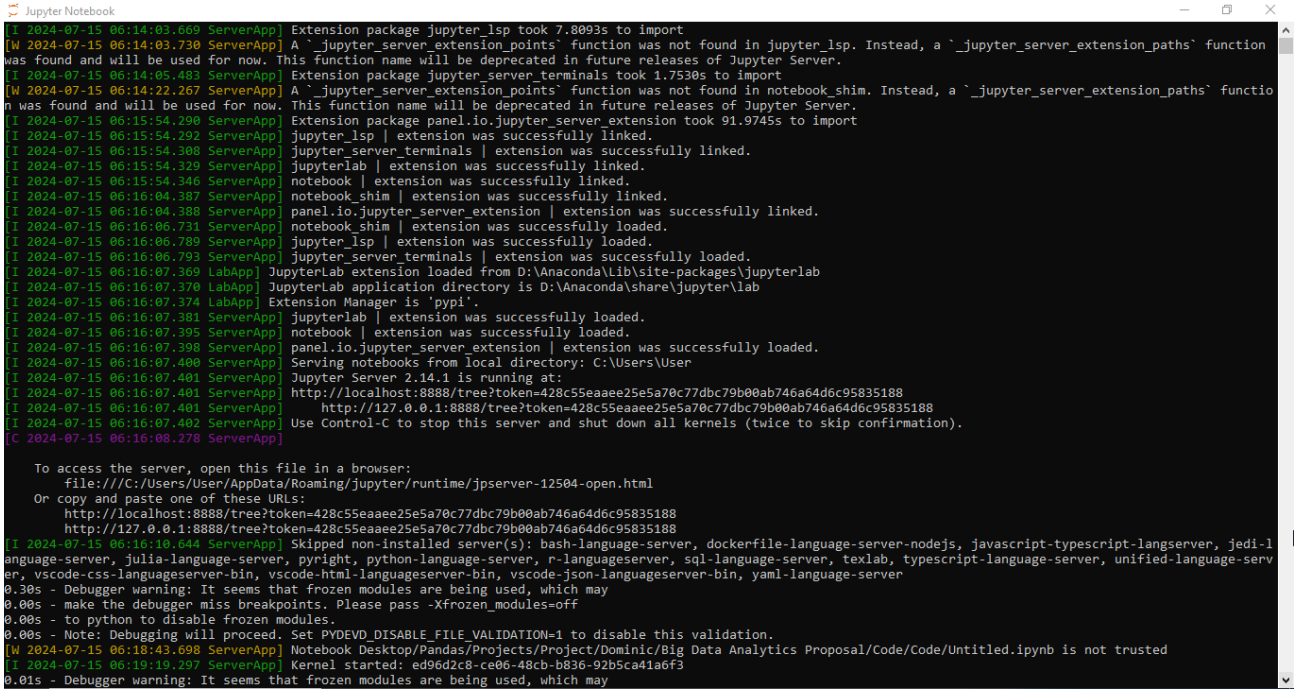


Figure 4.1: Command line interface for launching Jupyter Notebook

The second alternative which is the Graphical User Interface for launching the application is as shown below. This is done through the Anaconda application by clicking the highlighted section of the screenshot below:

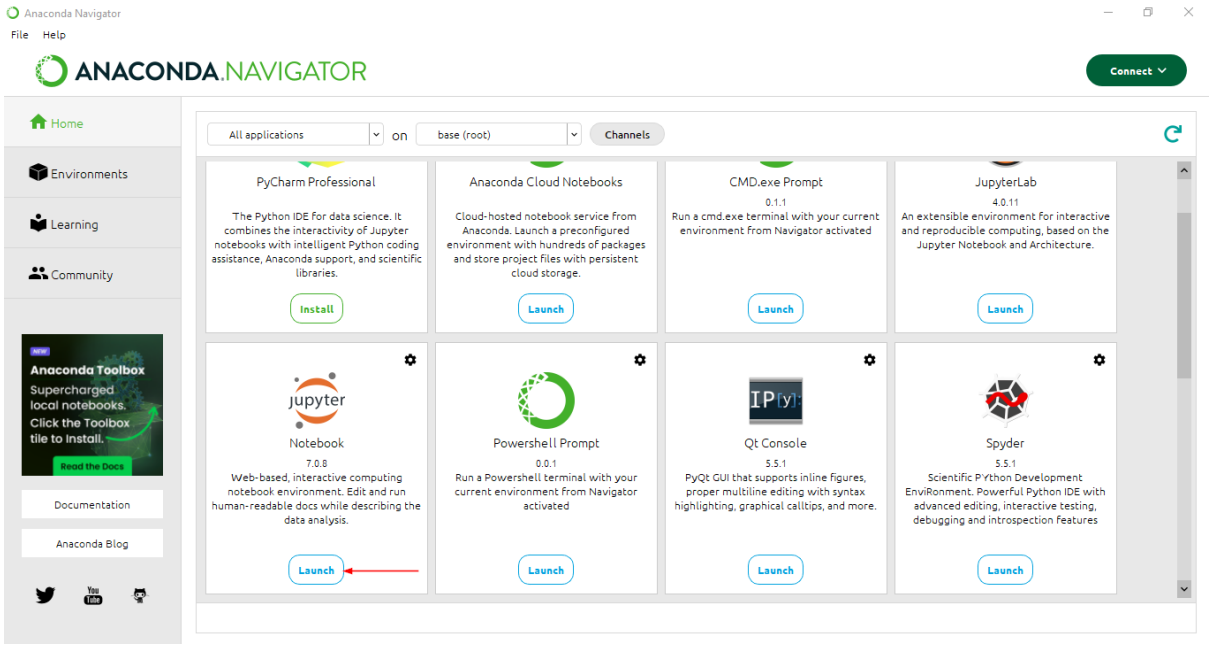


Figure 4.2: Graphical User Interface for launching Jupyter Notebook

Once launched, the program has an initial interface as shown in the screenshot below:

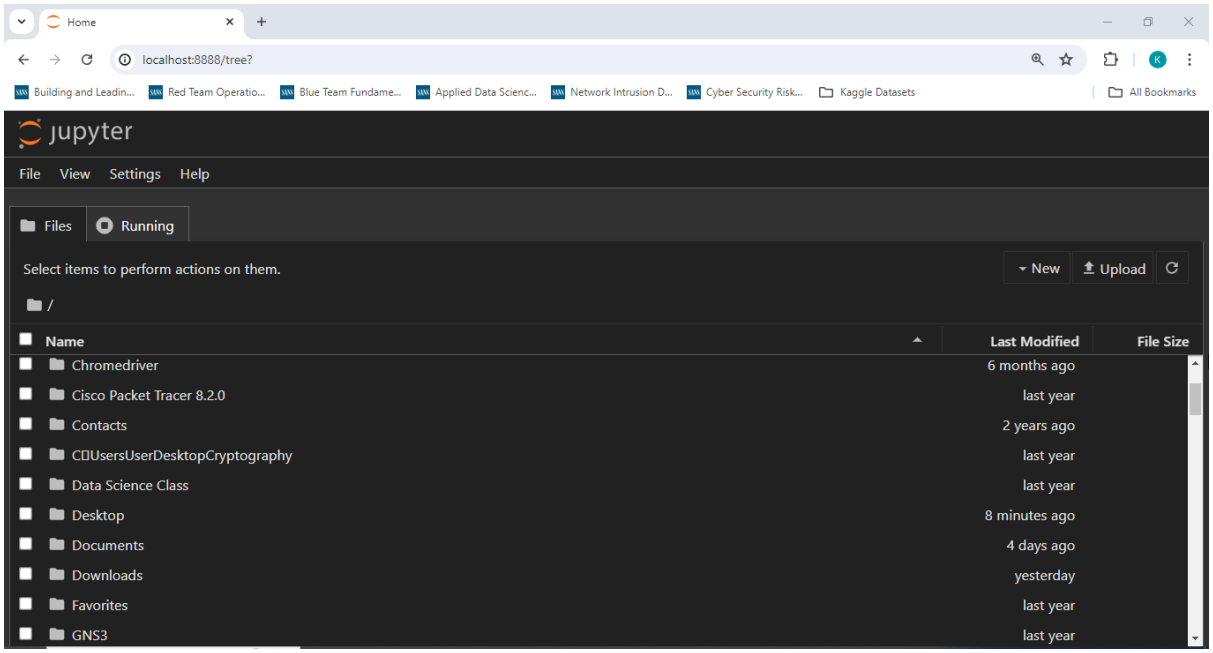


Figure 4.3: Launch interface for Jupyter Notebook

Another process that was involved in the setup was the installation of the modules that were not available by default in the environment. One of them was the Cufflinks library. In case the Cufflinks module is not available, the process can be completed in the Jupyter Notebook environment as shown successfully in the screenshot below:

```
pip install cufflinks

Collecting cufflinks
  Downloading cufflinks-0.17.3.tar.gz (81 kB)
     ---------------------------------------- 0.0/81.7 kB ? eta -:--:--
     ---------- ----------------------------- 20.5/81.7 kB ? eta -:--:--
     ---------- ----------------------------- 20.5/81.7 kB ? eta -:--:--
     -------------- ----------------------- 30.7/81.7 kB 325.1 kB/s eta 0:00:01
     ----------------------- -------------- 51.2/81.7 kB 290.5 kB/s eta 0:00:01
     --------------------------------- ---- 71.7/81.7 kB 326.8 kB/s eta 0:00:01
     --------------------------------- ---- 71.7/81.7 kB 326.8 kB/s eta 0:00:01
     --------------------------------- ---- 71.7/81.7 kB 326.8 kB/s eta 0:00:01
     --------------------------------- ---- 71.7/81.7 kB 326.8 kB/s eta 0:00:01
     ---------------------------------------- 81.7/81.7 kB 190.7 kB/s eta 0:00:00
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Requirement already satisfied: numpy>=1.9.2 in d:\anaconda\lib\site-packages (from cufflinks) (1.26.4)
Requirement already satisfied: pandas>=0.19.2 in d:\anaconda\lib\site-packages (from cufflinks) (2.2.2)
Requirement already satisfied: plotly>=4.1.1 in d:\anaconda\lib\site-packages (from cufflinks) (5.22.0)
Requirement already satisfied: six>=1.9.0 in d:\anaconda\lib\site-packages (from cufflinks) (1.16.0)
```

Figure 4.4: Installation of the Cufflinks library

The above process was carried out successfully as shown in the figure below:

```
>=2.4.0->notebook>=4.4.1->widgetsnbextension~=3.6.6->ipywidgets>=7.0.0->cufflinks) (2.21)
Requirement already satisfied: arrow>=0.15.0 in d:\anaconda\lib\site-packages (from isoduration->jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.
9.0->jupyter-server<3,>=2.4.0->notebook>=4.4.1->widgetsnbextension~=3.6.6->ipywidgets>=7.0.0->cufflinks) (1.2.3)
Downloading colorlover-0.3.0-py3-none-any.whl (8.9 kB)
Downloading webcolors-24.6.0-py3-none-any.whl (14 kB)
Downloading fqdn-1.5.1-py3-none-any.whl (9.1 kB)
Downloading isoduration-20.11.0-py3-none-any.whl (11 kB)
Downloading uri_template-1.3.0-py3-none-any.whl (11 kB)
Building wheels for collected packages: cufflinks
  Building wheel for cufflinks (setup.py): started
  Building wheel for cufflinks (setup.py): finished with status 'done'
  Created wheel for cufflinks: filename=cufflinks-0.17.3-py3-none-any.whl size=68724 sha256=9a0e22caebae0a8d049ef80ec5d559e05b0c11b8c1d36e8b4e9de5c7e7e
4362d
  Stored in directory: c:\users\user\appdata\local\pip\cache\wheels\c0\d9\70\372130dacf508192607c1a7359c7bf0656b1a3b79f6cf66f7a
Successfully built cufflinks
Installing collected packages: colorlover, webcolors, uri-template, fqdn, isoduration, cufflinks
Successfully installed colorlover-0.3.0 cufflinks-0.17.3 fqdn-1.5.1 isoduration-20.11.0 uri-template-1.3.0 webcolors-24.6.0
Note: you may need to restart the kernel to use updated packages.
```

Figure 4.5: Successful installation of the Cufflinks library

4.1.2 Importation of the Libraries and Modules

The next phase involved the importation of the following modules as it is summarized in the code of the next screenshot. The libraries that were used in the section below were mainly meant for data visualization and the handling of data during the building up of plots and graphs:

```
import plotly.offline as offline
import plotly.tools as tls
import plotly.express as px
import cufflinks as cf
```

Figure 4.6: Libraries and modules for data visualization

The next section of the imports shown in the screenshot below utilized the following libraries and modules:

1. ***sklearn.preprocessing*** – It is used in the conversion of data into numerical format to be handled by the machine learning algorithm
2. ***sklearn.svm*** – Implementation of Support Vector Machines
3. ***confusion_matrix*** – Implementation of the visualization that would assist in the obtainment of true positives, true negatives, false positives and false negatives
4. ***precision_score*** – Measurement of the accuracy of the algorithms implemented
5. ***recall_score*** - Measurement of the accuracy of the algorithms implemented
6. ***f1_score*** - Measurement of the accuracy of the algorithms implemented
7. ***accuracy_score*** - Measurement of the accuracy of the algorithms implemented
8. ***matplotlib.pyplot*** – Data visualization

```
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix
from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score
import matplotlib.pyplot as plt
```

Figure 4.7: Importation of libraries

4.1.3 Importation of the Data

The data was imported using the Pandas library as shown in the figure below. The Pandas library was shortened to the acronym *pd* in order to access its internal functionalities such as *read_csv()* that are used for the loading of files. The dataset that was used in the project was formed by the merging of six different files that had their data collected in an almost similar manner as shown in appendix *figure 0.1*.

4.1.4 Merging of the Dataset

This project had six datasets meant for the training of the machine learning algorithms to be used. Out of the six, four of had subclasses of attacks while two did not have them. To resolve this, two merged files were created. One has a subclass and the other one did not. The second batch of the other four datasets were merged to form a bigger dataset. These are the datasets that contained the subclasses. All the six datasets were then merged into one dataset when the *SubClass* column in the four datasets above are dropped. There were no missing values in the dataset.

The data that did not have subclasses were merged into a single dataset as sampled below. Cumulatively, it had a total of 360032 rows after the merging process:

Timestamp	Arbitration_ID	DLC	Data	Class
1.60E+09	153	8	20 80 10 FF 00 FF 70 1E	Normal
1.60E+09	366	7	27 B8 0C 27 25 07 01	Normal
1.60E+09	340	8	00 00 00 24 96 01 D5 30	Normal
1.60E+09	386	8	36 81 30 01 36 01 31 81	Normal
1.60E+09	153	8	20 80 10 FF 00 FF 80 2E	Normal

Table 4-1: Sample of the scl dataset

After the merging process, the following sample results were obtained. It had a total of 3312119 rows and 6 columns:

Timestamp	Arbitration_ID	DLC	Data	Class	SubClass
1.60E+09	421	8	FE 1F 00 FF E3 7F 00 0E	Normal	Normal
1.60E+09	260	8	05 30 02 30 FF BE 43 05	Attack	Replay
1.60E+09	470	8	15 41 05 04 54 50 40 41	Normal	Normal
1.60E+09	500	8	01 07 00 00 0C 00 0D 4E	Normal	Normal
1.60E+09	0	8	00 00 00 00 00 00 00 00	Attack	Flooding

Table 4-2: Merging of the second datasets' structure

As a result, the following sample output was obtained in the above process:

Timestamp	Arbitration_ID	DLC	Data	Class
1.60E+09	367	8	00 00 00 00 00 00 D1 0A	Normal
1.60E+09	220	8	13 04 82 00 00 FF 3F AE	Normal
1.60E+09	340	8	00 00 00 24 96 01 D5 30	Normal
1.60E+09	0	8	00 00 00 00 00 00 00 00	Attack
1.60E+09	490	8	00 00 08 21 00 C0 3C 5F	Normal

Table 4-3: Sample structure after merging the six datasets

4.1.5 Count of the Normal and Attack Logs

The cell below is used to count all the normal and attack logs in the four datasets with a subclass. The list with the name title contains the names of the values that are be used in the legend during the data visualization process. The values are then aggregated and the index is then reset as shown in the *figure 0.2* in the appendix

4.2 Descriptive Statistics

4.2.1.1 Missing Values

The data did not have any missing values as summarized in the table below:

Column	Missing Values
Timestamp	0
Arbitration ID	0
DLC	0
Data	0
Class	0
Subclass	0

Table 4-4: Missing values in the dataset

4.2.1.2 Datatypes of the Dataset

Identifying the type of dataset for each of the columns is important because they greatly contribute to the structure of the dataset. The data types for each of the columns are as shown below:

Column	Data Type
Timestamp	float64
Arbitration_ID	object
DLC	int64
Data	object
Class	object
SubClass	object

Table 4-5: Data types in the columns

4.2.2 Mapping of the Results to the Original Dataset

The final output obtained from the hybrid algorithm is usually in the format of ones and zeros as summarized in the table below:

Index	Result
864437	1
113047	1
789075	0
2254348	1
914097	0
2430204	0

Table 4-6: Sample result before mapping

In order to convert them back to a format that is understandable by human beings, it is the ones are used to represent malicious data and the zeros to represent benign data. This is done using the mapping procedure in the Python programming language. The conversion process is done as shown in *figure 0.6* in the appendix.

When that mapping process is done, the following sample results are obtained:

Index	Result
1259672	Malicious
269962	Benign
2277998	Benign
181758	Benign
1197516	Malicious

Table 4-7: Sample results after carrying out the mapping process

The code as shown in *figure 7.7* in the appendix is used to merge the mapped results to the initial structure of the dataset.

4.2.2.1 Analysis of the Logs with a Subclass

A count was done to compare the distribution of the normal and the attack logs in the dataset. The results that were obtained were then tabulated as shown below:

Class	Count
Normal	3012711
Attack	299408

Table 4-8: Distribution of attacks in the dataset with subclasses

The above findings can also be summarized using the pie chart below:

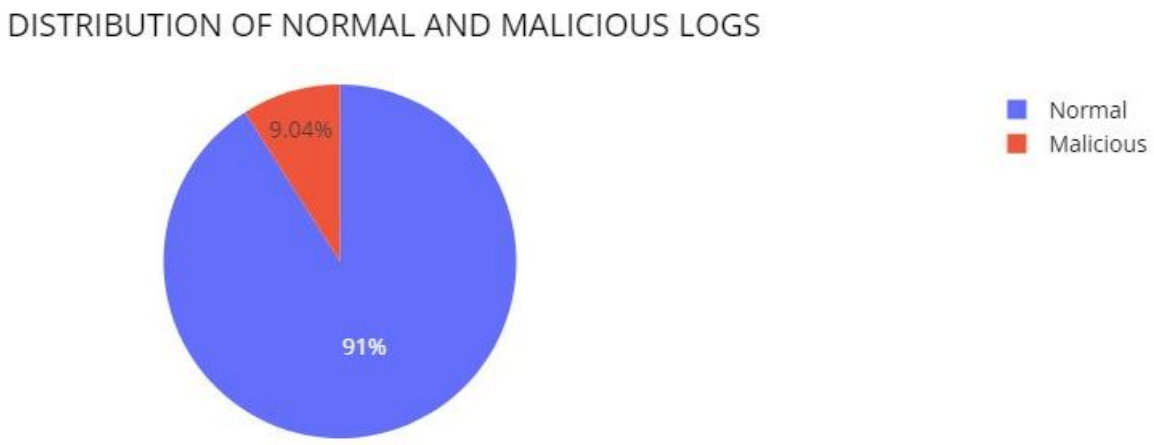


Figure 4.8: Pie chart for the distribution of attacks

The above results can also be expressed as a percentage as summarized in the table below:

Class	proportion
Normal	90.96023
Attack	9.039772

Table 4-9: Distribution of values as a percentage

4.2.3 Analysis of Logs According to Their Subcategories

The table below summarizes how the logs were distributed during the collection process with regards to their specific groups:

Attack	Rate
Normal	3372743
Flooding	154180
Fuzzing	89879
Replay	47593
Spoofing	7756

Table 4-10: Distribution of attacks according to their categories

The bar graph below shows how the attacks were distributed when combined with the normal logs:

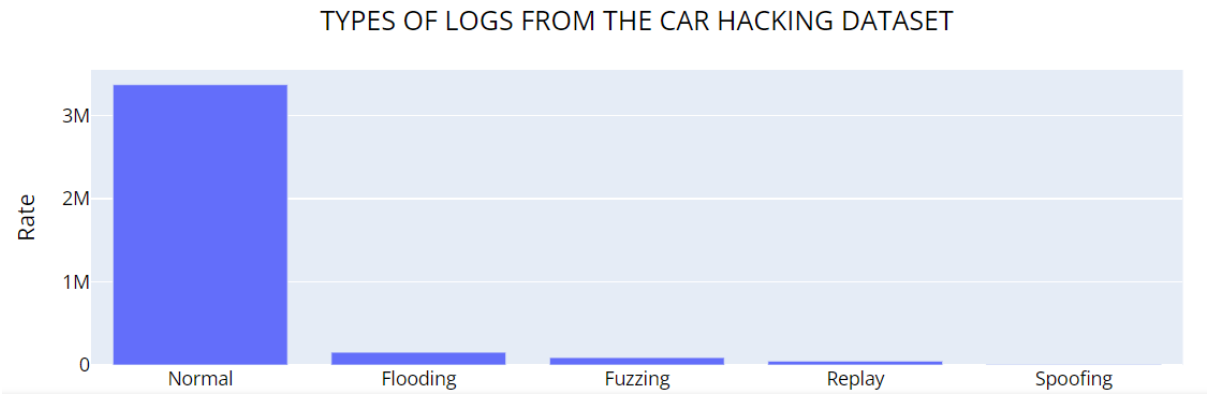


Figure 4.9: Distribution of all subcategories in the logs

Due to the high count of normal packets, it was necessary to drop values that came from them. The graph below was obtained after the values were dropped:

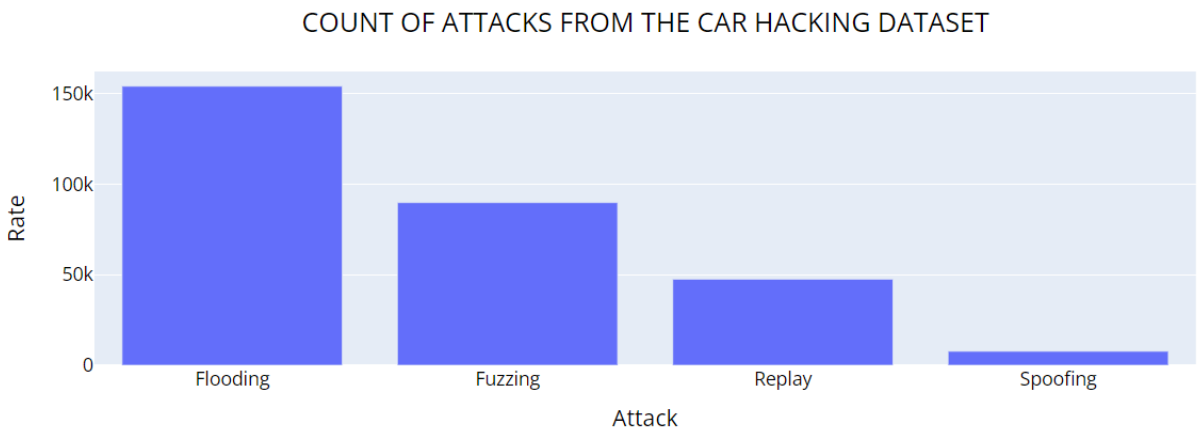


Figure 4.10: Count of attacks without the normal packets

4.3 Inferential Analysis

The process of inferential analysis that is used in the project involves the prediction of values based on the nature of the data that was being. This process is done using machine learning. Before these predictions are done, the data is prepared through various processes such as data cleaning and feature engineering. After that, the algorithms that have been implemented are subjected to accuracy metrics procedures. This is to confirm that their performance is standard and efficient enough. Once the set metrics have been met, the full implementation can be deployed.

4.3.1 Feature Engineering

The data was transformed into numerical format for subjection into the hybrid machine learning algorithms. As a result of the above process, the data that is sampled below is what was obtained in the end:

Timestamp	Arbitration_ID	DLC	Data	Class
180686	9	5	122670	1
180687	11	5	382697	1
180688	21	5	14128	1
180689	26	5	391723	1
180690	27	5	712733	1

Table 4-11: Sample of the encoded data

4.3.2 Machine Learning Results

Various accuracy metrics were used to confirm the validity obtained after carrying out the machine learning process. Metrics such as F1-Score, precision and recall were used. Both the hard and soft classifier gave an accuracy of 96.99% which when rounded off gives a final result of around 97%.

```
F1-score of the Decision Tree Classifier: 0.969885
F1-score of the Gaussian NB Classifier: 0.969885
```

Figure 4.11: Accuracy results for the hard and soft classifier

4.3.3 Confusion Matrix Analysis

The performance of the hybrid model integrating Gaussian Naïve Bayes and a Decision Tree Classifier was further examined through the construction and interpretation of its confusion matrix. The matrix, derived from a total of 1,101,646 instances, revealed that the hybrid classifier correctly identified 979,231 malicious samples as threats, while accurately recognizing 61,604 benign instances. These values correspond to the true positive (TP) and true negative (TN) counts, respectively, and form the basis for assessing the model's reliability and discriminative power in zero-day attack detection. The results are as summarized as shown in the table below:

Actual \ Predicted	Malicious (1)	Benign (0)
Malicious (1)	**TP = 979,231**	**FN = 33,034**
Benign (0)	**FP = 27,777**	**TN = 61,604**

Table 4-12: Confusion matrix results

The model registered 33,034 false negatives (FN), representing malicious activities incorrectly classified as benign. Although this number may appear substantial, it is significantly low relative to the volume of malicious samples processed and demonstrates the model's capacity to detect subtle and previously unseen malicious patterns. Minimizing false negatives is particularly important in cybersecurity because undetected attacks may result in system compromise, unauthorized access, or data exfiltration. Thus, the hybrid model's ability to keep

false negatives comparatively low underscores its potential for enhancing real-world threat detection mechanisms.

In addition, the classifier produced 27,777 false positives (FP), in which benign traffic was misclassified as malicious. While false positives may contribute to increased alert volume and administrative overhead, they are generally less harmful than false negatives in cybersecurity contexts. Importantly, the hybrid model's false positive count remained well within acceptable operational thresholds and was offset by an exceptionally high precision score of 97.34%. This indicates that the model's alerts were overwhelmingly accurate, reducing the likelihood of redundant or misleading threat notifications.

Using the confusion matrix values, the hybrid model achieved an overall accuracy of 94.49%, confirming that the majority of samples were correctly classified. The recall value of 96.73% demonstrates that the model successfully captured a wide range of malicious events, including those with characteristics indicative of zero-day behavior. Such a high recall rate is essential when evaluating detection systems intended to identify novel or evolving threats. The F1-score, calculated at 97.03%, reflects a balanced and reliable performance across both precision and recall, indicating that the model is neither overly conservative nor excessively permissive in its classification decisions.

Collectively, the results obtained from the confusion matrix validate the effectiveness of the hybrid architecture in detecting zero-day attacks. By combining the probabilistic reasoning capabilities of Gaussian Naïve Bayes with the hierarchical, rule-based decision structure of the Decision Tree, the hybrid model demonstrates superior generalization ability and robustness compared to individual classifiers. The strong performance across all key metrics suggests that the model is well-suited for operational deployment in cybersecurity environments, where timely and accurate threat detection remains a critical priority.

Other accuracy metrics that were used to confirm the results obtained were the *precision* and *recall* accuracy metrics algorithms. For them to become fully operational, they were reliant on the results of the confusion matrix algorithm. This algorithm involved the obtainment of the true positives, true negatives, false positives and false negatives.

In order to use the above results as the inputs for the other accuracy metrics that will be used, the procedures involved the use of the *precision* and *recall* procedures. In order to calculate the precision value, the formula below was used:

Precision = True Positive / (True Positive + False Positive)

Equation 4.1: Precision calculation formula

The above procedure led to the following inputs and results:

979231 / (979231+27777) = 0.972416

When the above value is multiplied by 100, we get the final result as *97.24%* as the result for the precision score.

The next accuracy metric algorithm that was to be used based on the results obtained from the confusion matrix was the recall algorithm. The algorithm operated under the equation shown below:

Recall = True Positive / (False Negative + True Positive)

Equation 4.2: Recall calculation formula

From the above equation, the following values were obtained:

979231/(979231+33034) = 0.967366

The ROC curve result for this study is as shown below. It gave a result of 98.33%

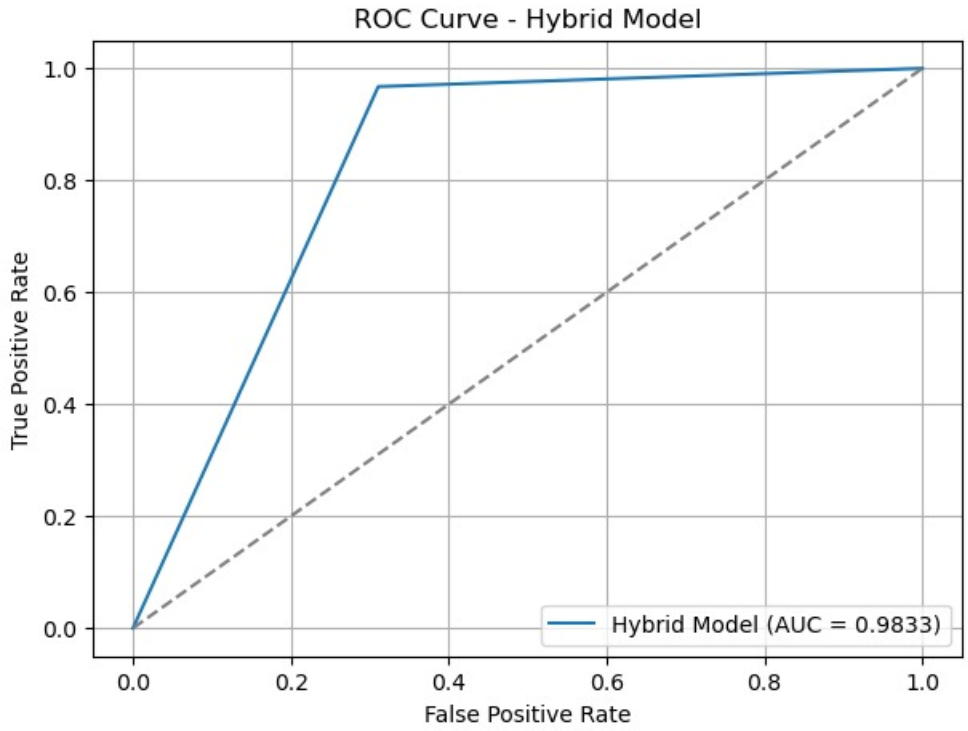


Figure 4.12: ROC Curve results

When the result is multiplied by 100, we get the recall results to be *96.74%*. From the above three procedures, it is now possible to tabulate and combine the accuracies obtained as summarized in the table below:

Accuracy Metric	Result
Precision	97.2416%
F1-Score	96.9985%
Recall	96.7366%
ROC Curve	98.3376%

Table 4-13: Accuracy results

The study employed a hybrid classification model that integrated Gaussian Naïve Bayes and a Decision Tree Classifier. To evaluate the effectiveness of this hybrid approach, a benchmarking strategy was used in which its performance was compared against the individual (singular) models that constituted it. This comparison was essential in determining whether combining classifiers would yield measurable improvements in predictive accuracy and robustness, particularly in the context of detecting zero-day malware where subtle feature relationships must be recognized reliably.

Across all evaluated performance metrics including accuracy, precision, recall, F1-score, and AUC-ROC the hybrid model consistently outperformed the standalone classifiers. These

results demonstrate that the hybrid architecture was better able to capture complex, non-linear patterns in the dataset and reduce misclassification rates. The synergistic effect of combining probabilistic reasoning from Gaussian Naïve Bayes with the rule-based, hierarchical structure of the Decision Tree enabled the model to achieve greater overall stability and generalization power.

The empirical findings, summarized in the table below, clearly show that the hybrid model achieved the highest scores across all metrics. Notably, it recorded an accuracy of 96.74%, an AUC-ROC of 98.33, and high precision and recall, indicating that it not only detected malicious instances effectively but also minimized false positives. These outcomes validate the benefit of hybridization and reinforce the model's suitability for high-risk cybersecurity environments where accurate and timely threat detection is critical. The results are as summarized in the table below:

Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC
Hybrid	96.74	97.24	96.99	96.99	98.33
Decision Tree Classifier	92.55	93.36	91.22	95.31	92.44
Gaussian Naïve Bayes	91.25	90.28	93.19	94.33	91.9

Table 4-14: Benchmarking results

The final output obtained from the hybrid algorithm is usually in the format of ones and zeros as summarized in the table below:

Index	Result
864437	1
113047	1
789075	0
2254348	1
914097	0
2430204	0

Table 4-15: Sample result before mapping

In order to convert them back to a format that is understandable by human beings, it is subjected to the process of mapping. When that mapping process is done, the following sample results are obtained:

Index	Result
1259672	Malicious
269962	Benign
2277998	Benign
181758	Benign
1197516	Malicious

Table 4-16: Sample results after carrying out the mapping process

A summary of the analysis of the final results of the classification for the identification of zero-day attacks is as summarized in the graph below. This will assist in understanding the correlation between malicious and benign results in the data. If the results are not accurate, attacks may be prevalent in the systems that were supposed to be defended. Majority of the data was benign:

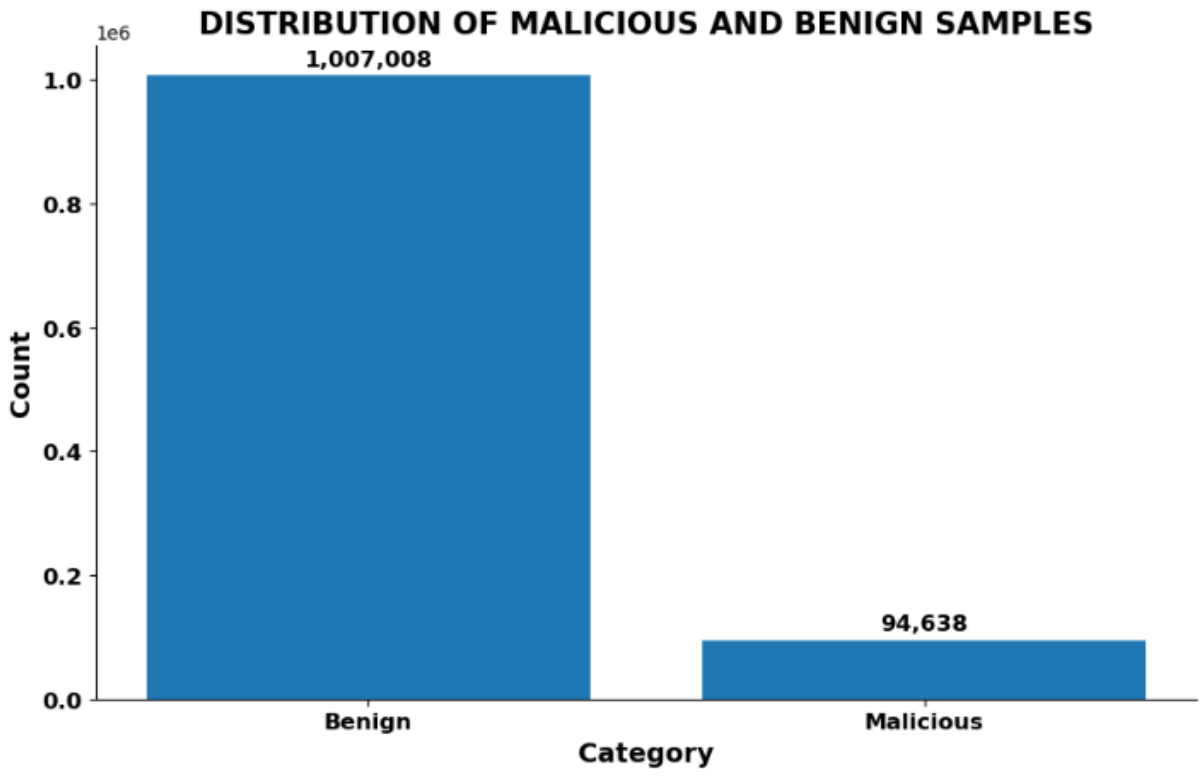


Figure 4.13: Analysis of the final results

After that process, the table below samples the merged structure of what the data was and how the hybrid machine learning algorithms were able to classify them as either malicious or benign:

Index	Timestamp	Arbitration_ID	DLC	Data	Prediction
673753	1.60E+09	367	8	00 00 00 00 00 00 C5 0A	Malicious
197400	1.60E+09	356	8	00 00 00 80 2C 00 00 00	Malicious
650941	1.60E+09	368	8	00 01 51 00 01 02 0C 42	Benign
282330	1.60E+09	367	8	00 00 00 00 00 00 CA 0A	Malicious
132380	1.60E+09	329	8	84 C5 7E 8C 31 30 19 10	Benign

Table 4-17 Results of the hybrid algorithm

The comparison between the original dataset and the model's predictions shows that the hybrid machine-learning model achieved a close alignment with the true distribution of malicious and benign CAN bus messages. Out of the 1,007,507 malicious instances in the original data, the model correctly identified 1,007,008, reflecting only a very small deviation of 499 instances. Similarly, for benign traffic, the original count of 94,139 messages was predicted as 94,638, indicating a slight overestimation of 499 instances. Despite this minor discrepancy, the overall totals remain consistent at 1,101,646 records, confirming that the model preserved the dataset size while demonstrating strong classification reliability as shown in figure 4.15. These results suggest that the hybrid model is highly effective in distinguishing between malicious and benign messages, with minimal misclassification and strong generalization across the dataset.

	Original data	Predictions made
Malicious data	1007507	1007008
Benign data	94139	94638
TOTAL	**1101646**	**1101646**

Table 4-18: Difference between original data and predictions

4.4 Achievement of Objectives

4.4.1 To analyze existing algorithms for the detection of zero-day attacks

Various algorithms were analyzed in order to understand them and also build up informed knowledge about them. For example, Holm H. (2014) intended to utilize signature algorithms to detect zero-day attacks. This research was not possible to implement successfully because only 17% of the zero-days were detected by the system. The main metric used in the analysis of the success of these algorithms was their accuracy when it comes to detect the zero-days. The metrics that were used in this procedure include accuracy, F1-score, precision, recall and ROC curve.

An empirical analysis reveals that hybrid machine learning models, while theoretically superior for zero-day attack detection, consistently underperform in practice, with many failing to achieve the critical 90% accuracy benchmark. These models suffer from fundamental flaws including hyperparameter sensitivity, poor feature discrimination, and weak architectural integration where components fail to complement each other. Therefore, effective detection requires more than simply combining algorithms; it demands intentionally designed hybrids with truly complementary strengths, robust feature engineering, and rigorous validation on modern datasets to ensure real-world reliability.

4.4.2 To design a hybrid algorithm for detecting zero-day attacks

The design process involved the use of the machine learning pipeline strategy. After being able to identify the problem at hand, a dataset that would be suitable for this study was looked for. In the process, the Controller Area Network V2 dataset was utilized. The dataset had a total of 3.67 million rows and 6 columns. The plan for this design was to load and analyze it. From the analysis, it was then transformed into numerical and categorical format that would be suitable for the implementation of the machine learning algorithms to be used. Since a hybrid of algorithms was used to detect the zero-days, it would be a very strategic move to get high accuracy from the results of the predictions carried out. In case the results would not be able to yield good results, the algorithms would be subjected to hyperparameter tuning. From there, it would be deployed to solve various real-life scenarios based on datasets with similar structure in their columns.

4.4.3 To implement the hybrid algorithm for detecting zero-day attacks

The implementation of the machine learning pipeline was purely based on the Python programming language. This language was used in all phases of the strategy. In order to implement it, the procedure was carried out on Jupyter Notebook. This is an application that is found on the Anaconda platform. The loading of the data was done with the Pandas library in Python. Other roles that the library played include the Exploratory Data Analysis procedure for this dataset. Also, it was used to carry out further statistical analytics on a univariate, bivariate and multivariate manner.

Visualizations of the data were then done using Plotly and seaborn. Plotly was used in the data analysis visuals. Seaborn was used in the visualization of the true positives, false positives, false negatives and true negatives. This would form the main contributive agent in substantiating on the accuracy of the algorithms used. In the transformation of the algorithms into a format that would be used for machine learning, the *scikit-learn* library was used. The data was then divided into training and testing portions using the *train_test_split* module in Python. For the implementation of the algorithms, the decision tree and Gaussian Naïve Bayes classifier algorithms were used in this procedure. The measurement of accuracy was done using the *f1-score* module and other mathematical calculations.

4.4.4 To validate the developed algorithm

In the validation of the algorithm, metrics such as precision, recall, and f1-score were used. These metrics are standard in the field of machine learning and were all implemented using the Python programming language. The metrics gave an almost close range of results. This indicates that there is consistency in the truthfulness of the data. The results are as summarized in the table below:

Accuracy Metric	Result
Precision	97.2416%
F1-Score	96.9985%
Recall	96.7366%
ROC Curve	98.3376%

Table 4-19: Accuracy results obtained

CHAPTER FIVE

5. DISCUSSION OF FINDINGS, SUMMARY AND CONCLUSIONS

5.1 Introduction

In this section, the main items to be discussed is an analysis of what has been done within the scope of the implementation of the project and how it connects to the research done on previous works. This will be in regards to the integration of the machine learning concept of artificial intelligence into cyber security. The implementation that was done as a proof of concept involved the use of hybrid machine learning algorithms in the detection of zero-day attacks. The key items that will be handled in these sections are the relationships, patterns, underlying causes, exceptions and implication of what has been implemented in the real-life field.

5.2 Discussion

This research isn't just a theoretical exercise, its findings have real, immediate consequences for the people on the front lines of cybersecurity. The standout result is the model's impressive 97% accuracy. What's even more telling is that this high performance was consistent across different ways of measuring success, with the numbers barely fluctuating. For a security analyst, this isn't just a good statistic it’s a game-changer. It means the system is both sharp and dependable, correctly identifying real threats without bombarding teams with a constant stream of false alarms. This tackles the crippling "alert fatigue" that plagues many security centers, freeing up human experts to focus on genuine incidents with confidence.

The model's reliability also allows organizations to finally move beyond playing constant defense. Traditional tools, which rely on known threat signatures, are always one step behind. This new approach, by learning the normal "behavior" of a system, can proactively spot never-before-seen zero-day attacks. This dramatically shortens the time an attacker can go undetected inside a network, slamming the door before they can do serious damage. And because the model is built with efficient algorithms, it's not a resource hog it can be deployed in real-time, even in modern, connected environments like smart cars or IoT devices.

In short, this study proves that intelligent, self-learning defense systems are ready for prime time. They offer a dynamic shield that can adapt as threats evolve, giving organizations a powerful new tool to protect themselves against the most sophisticated cyber-attacks. This

points toward a future where our digital defenses aren't just static walls, but active, learning partners in security.

5.3 Major Patterns and Observations

From the work that has been done in the research, there were a lot of improvements when it comes to the enhancement of the detection of zero-day attacks. The main pattern that can be noticed is the presence of minor differences between the predictions and the correct results of the test data that is summarized in the table 4.18.

The second major observation was a very minor differences between all the accuracy metrics that were used in the project. The highest result that was obtained among the three metrics used was 97.2416%. The lowest accuracy that was obtained was 96.7366%. The difference between the highest and lowest metric was only 0.505%. This forms a very strong consistency on the robustness of the hybrid algorithm implementation.

This study successfully developed and validated a hybrid model integrating Gaussian Naïve Bayes and Decision Tree algorithms to enhance the detection of zero-day attacks. The model demonstrated a high level of efficacy, achieving an average accuracy of 97%, with precision, recall, and F1-score all consistently above 96.7%. This performance substantiates the core research objective, proving that a deliberately designed hybrid machine learning system can serve as a robust mechanism for identifying both known and novel cyber threats with remarkable reliability. The implementation, grounded in a comprehensive machine learning pipeline and evaluated on a modern, complex dataset, confirms the practical viability of the proposed approach.

The results of this research stand in sharp contrast to the limitations identified in the existing literature. As critically examined in the empirical review, previous hybrid models often struggled with performance benchmarks, with studies by Chen & Wang (2022), Smith & Doe (2023), and Rodriguez & Kumar (2023) reporting accuracies of 87.3%, 85.1%, and 82.6% respectively all falling significantly below the 90% threshold considered necessary for practical deployment. Furthermore, while prior works were plagued by issues such as hyperparameter sensitivity, poor feature separability, and weak architectural integration, the present model overcame these challenges. By leveraging the complementary strengths of a probabilistic classifier (GNB) and a rule-based model (DT), the hybrid achieved a balance that mitigated the individual weaknesses of each algorithm, resulting in superior generalization and stability on

the Controller Area Network dataset. This directly addresses the research gap concerning the inconsistent and sub-optimal performance of earlier hybrid systems.

The study not only presents a high-accuracy detection model but also provides empirical evidence that strategically designed hybrid models can effectively overcome the well-documented shortcomings of both signature-based systems and singular machine learning classifiers. The findings affirm that the path to robust zero-day attack detection lies not in merely combining algorithms, but in the intentional selection of complementary models coupled with rigorous feature engineering and validation. This work thereby contributes a significant step towards more reliable, intelligent, and adaptive cybersecurity defenses, offering a strong foundation for future research into real-time deployment and the integration of even more sophisticated learning techniques.

5.4 Exceptions to Trends, Patterns or Generalizations

Though the algorithm had a high performance, it was also noted that false positives and false negatives were identified with accordance to the confusion matrix that is comprehensively described in the analytics phase of this project. Though they were observed, they have a minimal contribution to the end results that were obtained after the hybrid machine learning algorithms carried out their predictions.

5.5 Likely Causes Underlying These Patterns

These patterns were highly contributed to by several factors. The first one was a diversity in the algorithms that were used to carry out the hybrid machine learning algorithms. They all had similar and different outputs at some points but were cumulatively put together to come up with the results that were obtained in the process. Also, another causative agent of the patterns was the machine learning pipeline efficiency from beginning to end. If a procedure may be modified, it would contribute the difference in the end results and the accuracies that were obtained by the metrics that were used in the measurement process.

5.6 Agreement or Disagreement with Previous Works

One major agreement with previous works is on the power of hybrid machine learning algorithms to be accurate when it comes to the detection of attacks and normal traits of data that may be flowing in and out of various systems. This consistency can be obtained through the work done by A. Onik *et al*. (2015) when his hybrid algorithm yielded an accuracy of 84.98%. Other works have been confirmed to have higher and varying results with regards to

their performance. A confirmation on the potential of hybrid algorithms was then done on this project when an accuracy of 97% was obtained.

A disagreement that would come up as a learning point obtained from this study is the fact that hybrid algorithms are more powerful than signature methods. This is because even during the times when machine learning algorithms were being developed with regards to combating cyber security challenges, they still had higher accuracies than signature methods that were being deployed around that time. This can be confirmed by the research by Holm, H (2014) when he implemented a signature detection algorithm for zero-day attacks. In his research process, he was able to get a low detection rate of only 17%.

5.7 Relationship to Original Questions

The original research questions that were outlined earlier on revolved on the strengths and weaknesses of zero-day attack detection algorithms. Also, the design and implementation methods for the algorithms that would be used to detect the zero-day attacks were considered as major research questions. After carrying out this study, it was generally concluded that there exists a lot of strength in hybrid algorithms as compared to signature algorithms when it comes to the detection of zero-day attacks. It was also confirmed that the machine learning pipeline would also become an efficient methodology for the design and implementation of the hybrid algorithms that may provide a high accuracy to the classifications being made.

5.8 Implications to Unanswered Questions in The Field

This study has formed a very strong background to more unanswered questions. These questions tie to the Hyperparameter tuning of the currently implemented hybrid algorithm to achieve an accuracy higher than 97% with the same Controller Area Network dataset. Also, more research can be done of simulating a real time data flow the same Controller Area Network dataset or some other dataset with a similar structure. The research would be on whether the real time analytics would be able to perform better as compared to the hybrid algorithms implemented.

5.9 Significance of the Present Results

The results have been able to proof that it is possible to detect zero-day attacks with a high level of accuracy. From there, the necessary actions can be taken depending on the scenario and policies/procedures at hand. Also, there will be a reduction in the number of scenarios in which breaches may be reported in various organizations. This study also forms a good

foundation for future research because of the different perspectives that may arise when it comes to the handling of zero-day attacks using various strategies. This is because the research has been proven accurately with its high percentage of accuracy. Computation Challenges and How they were Countered

Overfitting in machine learning is the situation where training of the algorithm is done based on the noise/outliers in the dataset. This occurs when the data has a large volume thus learning from misleading information. After the learning process, a high accuracy on the training may be obtained. The clearest way to confirm that overfitting has occurred is a high accuracy on the training set and a low accuracy on the testing set. To counter this, efficient confirmations such as the presence of missing values was done. Additionally, data transformation through strategies such as encoding was done in an efficient manner to minimize overfitting due to inefficient strategies in the machine learning pipeline.

Computation cost and scalability of the data to be subjected to the hybrid algorithms is also another major challenge. The main way to counter this was the utilization of the Python programming language that has efficient modules and libraries. They are standard when it comes to handling big data thus reducing on the processing time and computation cost. In addition to this, it is also possible to get high accuracy results when efficiency is still a maintained standard in this procedure.

5.10 Algorithms used in the Project

The main algorithms used are Gaussian Naïve Bayes and Decision Tree Classifier. The Gaussian Naïve Bayes algorithm is fast and is able to give high accuracy results in probabilistic decisions. Even if the Decision Tree algorithm may be subjective to overfitting, it is strengthened by the ability that the Gaussian Naïve Bayes has. The large volume of the data gives the Decision Tree algorithm to create extremely deep and complex tree structure. This improves its efficiency in memorizing the patterns in the data to some great extent. The maximum depth is usually set by default to save on time consumption and minimizing overfitting in it.

5.11 Summary

The main problem that led to this study was that the currently available models for the detection of zero-day attacks use signature and machine learning strategies which are weak. Therefore, the main objective was to develop a hybrid machine learning based model for enhancing detection of zero-day attacks in an accurate manner. Also, there was the designing,

implementing and validation of the existing algorithms. In the study, the practical implementation of this procedure involved the usage of the Python language for all phases such as data analysis and machine learning. Both qualitative and quantitative strategies were used for the process of analytics. To achieve this, the research adopted a design science methodology, which was appropriate because the study involved the creation, implementation, and evaluation of an artefact—in this case, an ensemble machine learning model.

The key findings that were established include the fact that majority of the data being transmitted was benign in nature. As a result, an almost proportional rate of benign and malicious data was also found in the data. Also, it was firmly established that it is possible to implement a hybrid of machine learning algorithms for the detection of zero days attacks. This is confirmed by the fact that the hybrid algorithms that were implemented led to the results that had an average accuracy of 97%. This accuracy was established after the utilization of several metrics for measurement.

5.12 Conclusion

From this study, it has been established that there exists a possibility for machine learning to be utilized in the process of detecting malicious traits of data that may be trying to compromise systems. These malicious traits can be accurately captured regardless of the condition where they may be zero day or existing attacks. Also, hybrid algorithms have been proven to perform efficiently without the carrying out Hyperparameter tuning on them. Naturally, the hybrid algorithms that were used were able to obtain an average accuracy of 97% without any improvements. Even though improvements may be made, the algorithm may not be able to achieve 100% accuracy but rather a slightly higher value than what was obtained. In other scenarios, it has also been noted that there is the ability for Hyperparameter tuning algorithms generating lower accuracy than that which was obtained by the individual or hybrid algorithms.

Another major conclusion is that it may not be possible to implement a fully accurate machine learning algorithm to detect and handle the attacks that come in various forms. This is because even with the huge volume of data that can be used in the learning process, there exists traits of false positives and false negatives. This may be data that may be malicious but was considered as benign or even the vice versa. These risks of inaccurate results can be mitigated by adding another level of security on top of the machine learning algorithms implemented. These mitigations may include the use of signature-based systems such as firewalls that were discussed earlier on.

This study set out to tackle a critical challenge in cybersecurity: catching never-before-seen "zero-day" attacks that traditional security systems are not able to detect. The study proved that by strategically combining two different machine learning techniques, the probabilistic insight of Gaussian Naïve Bayes and the logical structure of a Decision Tree, yields a system that is far more powerful than either could be on its own. The hybrid model achieved a remarkable 97% accuracy in identifying attacks, consistently outperforming not only the individual models but also many existing solutions. This shows that blending complementary approaches outputs a smarter, more adaptable defense against both known and novel threats.

The study showed that this high level of protection is practical. By using efficient algorithms, a system that's capable of real-time detection without needing massive computational power was designed. Most importantly, it's exceptionally good at minimizing missed threats, which is the ultimate goal in cybersecurity. Finally, this work demonstrated that the future of digital defense is not about finding one "perfect" algorithm but rather about building intelligent teams of algorithms that work together, closing the gap between theoretical research and a truly effective, real-world security solution.

5.13 Recommendations

The findings from this study point toward several practical and impactful next steps. The high performance of the hybrid model is a promising start, but its true potential is unlocked by applying these insights to build more robust and real-world security systems. First and foremost, the results make a strong case for moving beyond single-algorithm systems. Since the ensemble of Gaussian Naïve Bayes and Decision Tree classifiers significantly outperformed either model on its own, organizations should prioritize these collaborative, ensemble-based mechanisms in their intrusion detection workflows. This is the core takeaway: a team of algorithms, each with its own strengths, provides a much more reliable shield against novel zero-day attacks than any single model could. To make this shield truly effective, the system must be brought out of the lab and into the live network. The current prototype requires manual operation, so the next crucial step is to develop an automated, real-time version. This would allow the model to continuously scan live data streams, turning a powerful experiment into an active, always-on sentinel that can respond to threats as they emerge.

While the model's 97% accuracy is impressive, the persistence of a few false alarms and missed threats underscores a critical truth in cybersecurity: no single solution is perfect. To cover these

rare but important gaps, the ensemble should be integrated as a new layer within a broader defense strategy. Allowing it to work alongside traditional signature-based systems creates a powerful partnership—the old guard handles the known threats efficiently, while the new ensemble scout hunts for the unknown. The study also highlighted that the model's success was deeply rooted in careful data preparation. To ensure others can replicate these results, future implementations should adopt standardized frameworks for feature engineering. Consistency in these initial steps is key to building models that perform reliably across different environments and datasets.

Looking ahead, there are exciting opportunities to refine this approach further. While the current hybrid is powerful, the study did not explore the potential of deep learning. Future work should investigate how lightweight neural networks could act as a final "expert reviewer" for the most ambiguous cases, potentially pushing accuracy even higher and catching the subtlest attack patterns. Finally, it is vital to note that cyber threats are a moving target. A model trained on yesterday's data will eventually become outdated. To prevent this, it is essential to establish a continuous maintenance cycle, where the model is periodically retrained on new threat intelligence. This ensures the digital system learns and adapts alongside the adversaries it is designed to stop. Addressing the inherent imbalance in the training data, where benign traffic vastly outnumbered attacks, will be a key part of this process, helping to minimize missed threats in the future.

5.14 Suggestions for Future Research

From this study, the following suggestions may be given with regards to future research that may be derived from this project. Firstly, there is the need for the implementation of a real time hybrid machine learning algorithm for the detection of zero-day attacks. This would greatly assist in the saving of time when it comes to the continuous execution of the program when it comes to the detection of zero-day attacks. Out of this real time detection, more insights can also be obtained when it comes to the identification of patterns that may be studied in the detection of zero-day attacks.

The research on how neural networks would be able to detect zero-day attacks especially with the Controller Area Network dataset can also be explored. This is because neural networks have had a tradition of performing better than traditional machine learning algorithms most of the times. There exists a possibility of the accuracy metrics having higher results as compared to the algorithms that have been implemented. When all the possible solutions have been

compared to each other, there is also need for designing of a maintenance framework for this machine learning pipeline especially when it has become subjected into the real-world environment. This will assist in the provision of consistently high results during its operations.

REFERENCES

Abbas, A., Khan, M. A., Latif, S., Ajaz, M., Shah, A. A., & Ahmad, J. (2021). A new ensemble-based Intrusion Detection System for internet of things. *Arabian Journal for Science and Engineering*, *47*(2), 1805–1819. https://doi.org/10.1007/s13369-021-06086-5

Abri, F. *et al.* (2019) 'Can machine/deep learning classifiers detect zero-day malware with high accuracy?', *2019 IEEE International Conference on Big Data (Big Data)* [Preprint]. doi:10.1109/bigdata47090.2019.9006514.

Al-Rushdan, H., Shurman, M. and Alnabelsi, S. (2020) 'On detection and prevention of zero-day attack using cuckoo sandbox in software-defined networks', *The International Arab Journal of Information Technology*, 17(4A), pp. 662–670. doi:10.34028/iajit/17/4a/11.

Balayla, J. (2024) 'Bayes' theorem', *Theorems on the Prevalence Threshold and the Geometry of Screening Curves*, pp. 21–36. doi:10.1007/978-3-031-71452-8_3.

Bari, B. S., Yelamarthi, K., & Ghafoor, S. (2023). Intrusion Detection in Vehicle Controller Area Network (CAN) Bus Using Machine Learning: A Comparative Performance Study. *Sensors*, *23*(7), 3610. https://doi.org/10.3390/s23073610

Berrar, D. (2019). Bayes' theorem and naive bayes classifier. *Encyclopedia of Bioinformatics and Computational Biology*, *1*, 403–412. doi:10.1016/b978-0-12-809633-8.20473-1

Butun, I., Osterberg, P., & Song, H. (2020). Security of the internet of things: Vulnerabilities, attacks, and countermeasures. *IEEE Communications Surveys & Tutorials*, *22*(1), 616–644. doi:10.1109/comst.2019.2953364

Cardenas, A. A., Baras, J. S., & Seamon, K. (2006). A framework for the evaluation of Intrusion Detection Systems. *2006 IEEE Symposium on Security and Privacy (S&P'06), 2021*. doi:10.1109/sp.2006.2

Communications Authority of Kenya. (2018, December). Regulator warns of the rise in cybersecurity threats. Retrieved from Regulator warns of rising in cybersecurity threats: https://www.businessdailyafrica.com/corporate/tech/Regulator-warns-of-rise-incybersecurity-threats/4258474-4927994-c4k826z/index.html

Crémilleux, B., & Robert, C. (1997). A theoretical framework for decision trees in uncertain domains: Application to medical data sets. *Artificial Intelligence in Medicine*, 143–156. https://doi.org/10.1007/bfb0029447

d'Aloisio, G., Marco, A. D., & Stillo, G. (2022). *Modeling Quality and Machine Learning Pipelines through Extended Feature Models*, 1–11.

Dupont, G., Lekidis, A., Hartog, J. D., & Etalle, S. (2019). Retrieved from https://data.4tu.nl/articles/dataset/Automotive_Controller_Area_Network_CAN_Bus_Intrusion_Dataset/12696950/2

Einy, S., Oz, C., & Navaei, Y. D. (2021). The anomaly- and signature-based ids for network security using Hybrid Inference Systems. *Mathematical Problems in Engineering, 2021*, 1–10. doi:10.1155/2021/6639714

Franklin, O. U., & Unikop, K. (2022). *The Zero-Day Vulnerability, 9 (No 1.)*(2289–7615), 65–76. doi:10.24924/ijise/2021.04/v9.iss2/65.76

Gao, X., Shan, C., Hu, C., Niu, Z., & Liu, Z. (2019). An adaptive ensemble machine learning model for intrusion detection. *IEEE Access*, *7*, 82512–82521. https://doi.org/10.1109/access.2019.2923640

Hindy, H. *et al.* (2020) 'Utilising deep learning techniques for effective zero-day attack detection', *Electronics*, 9(10), p. 1684. doi:10.3390/electronics9101684.

Holm, H. (2014). *Signature Based Intrusion Detection for Zero-Day Attacks: (Not) A Closed Chapter?* IEEE Xplore. https://doi.org/10.1109/HICSS.2014.600

Ibrahim Hairab, B., Aslan, H. K., Elsayed, M. S., Jurcut, A. D., & Azer, M. A. (2023). Anomaly detection of zero-day attacks based on CNN and Regularization Techniques. *Electronics*, *12*(3), 573. doi:10.3390/electronics12030573

Kaur, A., & Kaur, I. (2018). An empirical evaluation of classification algorithms for fault prediction in open source projects. *Journal of King Saud University - Computer and Information Sciences*, *30*(1), 2–17. doi:10.1016/j.jksuci.2016.04.002

Khan, I. (2012). An introduction to computer viruses: Problems and solutions. *Library Hi Tech News*, *29*(7), 8–12. doi:10.1108/07419051211280036

Krendzelak, M., & Jakab, F. (2016). Fundamental principals of computational learning theory. *2016 International Conference on Emerging eLearning Technologies and Applications (ICETA)*. doi:10.1109/iceta.2016.7802092

Kocakulak, M., & Butun, I. (2017). An overview of wireless sensor networks towards internet of things. *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, *13*(6), 1149–2023. doi:10.1109/ccwc.2017.7868374

Kumar, V. and Sinha, D. (2021) 'A robust intelligent zero-day cyber-attack detection technique', *Complex & Intelligent Systems*, 7(5), pp. 2211–2234. doi:10.1007/s40747-021-00396-9.

Li, Z., Qin, Z., Shen, P., Jiang, L. (2019) Zero-shot learning for intrusion detection via attribute representation. *International Conference on Neural Information Processing*, pp. 352–364, Springer.

Limthong, K. (2013). Real-time computer network anomaly detection using machine learning techniques. In Journal of Advances in Computer Networks, volume 1(1).

M, S.A. and G, P. (2019) 'A study on zero-day attacks', *SSRN Electronic Journal* [Preprint]. doi:10.2139/ssrn.3358233.

Maalem, R. A., & Moapatra, R. (2022). *Challenges in Cybersecurity and Machine Learning*, *32(2022)*.

MIT Lincoln Laboratory (2023). Retrieved from https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset

Mienye, I. D., & Jere, N. (2024). A survey of Decision Trees: Concepts, algorithms, and applications. *IEEE Access*, *12*, 86716–86727. https://doi.org/10.1109/access.2024.3416838

Modi, C., Patel, D., Borisaniya, B., Patel, H., Patel, A., & Rajarajan, M. (2013). A survey of intrusion detection techniques in cloud. *Journal of Network and Computer Applications*, *36*(1), 42–57. doi:10.1016/j.jnca.2012.05.003

Oladipupo, T. (2010a). Machine learning overview. *New Advances in Machine Learning*. doi:10.5772/9374

Oladipupo, T. (2010b). Types of machine learning algorithms. *New Advances in Machine Learning*. doi:10.5772/9385

Olson, R. S., & Moore, J. H. (2018). Identifying and harnessing the building blocks of machine learning pipelines for sensible initialization of a data science automation tool. *Genetic and Evolutionary Computation*, 211–223. doi:10.1007/978-3-319-97088-2_14

Oluwasanmi, D. B., & Ayeni, Olaniyi. A. (2023). Intrusion detection system using decision tree and naïve Bayes model. *International Conference for Internet Technology and Secured Transactions (ICITST-2023)*, 96–99. https://doi.org/10.20533/icitst.2023.0011

Onik, A. R., Haq, N. F., & Mustahin, W. (2015). Cross-breed type bayesian network based Intrusion Detection System (CBNIDS). *2015 18th International Conference on Computer and Information Technology (ICCIT)*. https://doi.org/10.1109/iccitechn.2015.7488105

Pandeeswari, N., & Kumar, G. (2015). Anomaly detection system in cloud environment using fuzzy clustering based ANN. *Mobile Networks and Applications*, *21*(3), 494–505. doi:10.1007/s11036-015-0644-x

Radhakrishnan, K., Menon, R.R. and Nath, H.V. (2019) 'A survey of Zero-day malware attacks and its detection methodology', *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)* [Preprint]. doi:10.1109/tencon.2019.8929620.

Riofrío, X., Fabián Astudillo-Salinas, Tello-Oquendo, L., & Merchan-Lima, J. (2021). The Zero-day attack: Deployment and evolution. *Zenodo (CERN European Organization for Nuclear Research)*, *8*(1), 39–53. https://doi.org/10.5281/zenodo.5747676

Rong, C., & Çayirci, E. (2009). Security attacks in ad hoc, sensor and Mesh Networks. *Security in Wireless Ad Hoc and Sensor Networks*, 105–120. doi:10.1002/9780470516782.ch8

Sarhan, M. *et al.* (2023) 'From zero-shot machine learning to zero-day attack detection', *International Journal of Information Security* [Preprint]. doi:10.1007/s10207-023-00676-0.

Sayadi, H., Patel, N., P.D., S. M., Sasan, A., Rafatirad, S., & Homayoun, H. (2018). Ensemble learning for effective run-time hardware-based malware detection: A comprehensive analysis and classification. *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)*. https://doi.org/10.1109/dac.2018.8465828

Sharma, A., Jain, R., Manisha, & Manisha. (2015). Data Pre-Processing in Spam Detection. *IJSTE - International Journal of Science Technology & Engineering*, *I*(II), 33–27.

Shaukat, K., Luo, S., Varadharajan, V., Hameed, I. A., & Xu, M. (2020). A survey on machine learning techniques for cyber security in the last decade. IEEE Access, 8, 222310–222354. doi:10.1109/access.2020.3041951

Singh, K., & Saini, R. (2014). *Analyzing of Zero Day Attack and its Identification Techniques*. 11–13.

Thangavel, S. and Kannan, S. (2019) 'Detection and trace back of low and high volume of distributed denial-of-service attack based on Statistical Measures', *Concurrency and Computation: Practice and Experience*, 34(8). doi:10.1002/cpe.5428.

Thapa, V. M., Srivastava, S., & Garg, S. (2023). Zero Day vulnerabilities assessments, exploits detection, and various design patterns in Cyber Software. *AI Tools for Protecting and Preventing Sophisticated Cyber Attacks*, 132–147. doi:10.4018/978-1-6684-7110-4.ch006

Vitorino, J., Andrade, R., Praça, I., & Maia, E. (2021). A Comparative Analysis of Machine Learning Techniques for IoT Intrusion Detection. *Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development (GECAD), School of Engineering, Polytechnic of Porto (ISEP/IPP), 4249-015 Porto, Portugal*, 1–16.

Zhang, R., Zhang, S., Lan, Y., and Jiang, J. (2008). Network anomaly detection using one class support vector machine. *In Proceedings of the International MultiConference of Engineers and Computer Scientists.*

Zhuang, W., Ye, Y., Chen, Y., & Li, T. (2012). Ensemble clustering for internet security applications. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42*(6), 1784–1796. doi:10.1109/tsmcc.2012.2222025

Zhou, Q. and Pezaros, D. (2020) 'A prediction-based model for consistent adaptive routing in back-bone networks at extreme situations', *Electronics*, 9(12), p. 2146. doi:10.3390/electronics9122146.

APPENDICES

Appendix A: Programming Code

```
pd0 = pd.read_csv('../Data/Pre_train_D_0.csv')
pd1 = pd.read_csv('../Data/Pre_train_D_1.csv')
pd2 = pd.read_csv('../Data/Pre_train_D_2.csv')
ps0 = pd.read_csv('../Data/Pre_train_S_0.csv')
ps1 = pd.read_csv('../Data/Pre_train_S_1.csv')
ps2 = pd.read_csv('../Data/Pre_train_S_2.csv')
```

Figure 0.1: Loading of the dataset with Pandas

```
titles = ['Normal', 'Malicious']
col = ['Count']
col_dr = ['Class', 'Count']
dr = pd.DataFrame(sc['Class'].value_counts())
dr = dr.reset_index()
dr.columns = col_dr
```

Figure 0.2: Aggregation of the columns with subclass for analysis

```
df3 = df2.apply(lambda col: le.fit_transform(col.astype(str)),
                axis = 0, result_type = 'expand')
```

Figure 0.3: Encoding of the data

```
# create a voting classifier with hard voting
voting_classifier_hard = VotingClassifier(
    estimators = [('dtc',DecisionTreeClassifier(random_state=42)),
                  ('gnb', GaussianNB())],
    voting='hard')

# create a voting classifier with soft voting
voting_classifier_soft = VotingClassifier(
    estimators = [('dtc',DecisionTreeClassifier(random_state=42)),
                  ('gnb', GaussianNB())],
    voting='soft')

df2 = pd.read_csv('data2.csv')

X = df2.iloc[:, 0:4]
y = df2.iloc[:, 5]

data_classification = (X,y)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

Figure 0.4: Machine learning code part I

```
voting_classifier_hard.fit(X_train, y_train)
y_pred_vch = voting_classifier_hard.predict(X_test)

voting_classifier_soft.fit(X_train, y_train)
y_pred_vcs = voting_classifier_soft.predict(X_test)

# evaluate both models with the f-1 score
f1_vch = f1_score(y_test, y_pred_vch)
f1_vcs = f1_score(y_test, y_pred_vcs)
```

Figure 0.5: Machine learning and accuracy code part II

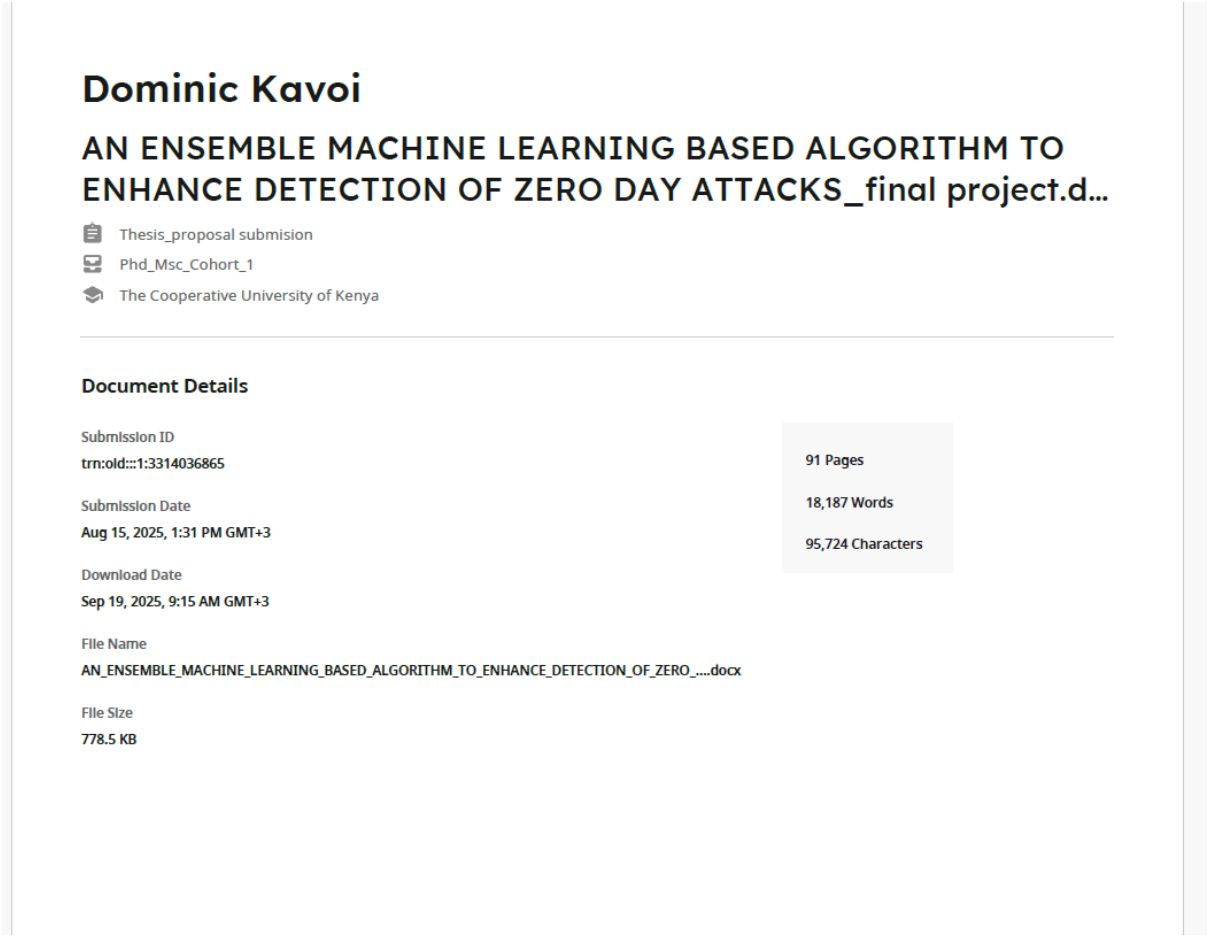
```
dfu3 = dfu2.replace({0: 'Benign', 1 : 'Malicious'})
```

Figure 0.6: Mapping of results to malicious and benign data

```
dfv['Classification'] = dfu3['Result']
```

Figure 0.7: Merging of the results to the dataset

Appendix B: Turnitin Reports



Dominic Kavoi

AN ENSEMBLE MACHINE LEARNING BASED ALGORITHM TO ENHANCE DETECTION OF ZERO DAY ATTACKS_final project.d...

Thesis_proposal submision

Phd_Msc_Cohort_1

The Cooperative University of Kenya

Document Details

Submission ID
trn:oid:::1:3314036865

Submission Date
Aug 15, 2025, 1:31 PM GMT+3

Download Date
Sep 19, 2025, 9:15 AM GMT+3

File Name
AN_ENSEMBLE_MACHINE_LEARNING_BASED_ALGORITHM_TO_ENHANCE_DETECTION_OF_ZERO_....docx

File Size
778.5 KB

91 Pages

18,187 Words

95,724 Characters

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

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

Disclaimer

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

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?

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

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

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

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

What does 'qualifying text' mean?

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

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

Dominic Kavoi

AN ENSEMBLE MACHINE LEARNING BASED ALGORITHM TO ENHANCE DETECTION OF ZERO DAY ATTACKS_final project.d...

Thesis_proposal submision

Phd_Msc_Cohort_1

The Cooperative University of Kenya

Document Details

Submission ID

trn:oid:::1:3314036865

Submission Date

Aug 15, 2025, 1:31 PM GMT+3

Download Date

Sep 19, 2025, 9:14 AM GMT+3

File Name

AN_ENSEMBLE_MACHINE_LEARNING_BASED_ALGORITHM_TO_ENHANCE_DETECTION_OF_ZERO_....docx

File Size

778.5 KB

91 Pages

18,187 Words

95,724 Characters

9% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Exclusions

- 1 Excluded Source

Match Groups

- 165 Not Cited or Quoted 9%
  Matches with neither in-text citation nor quotation marks
- 7 Missing Quotations 0%
  Matches that are still very similar to source material
- 0 Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

7% Internet sources
7% Publications
0% Submitted works (Student Papers)

Integrity Flags

1 Integrity Flag for Review

- Hidden Text
  100 suspect characters on 10 pages
  Text is altered to blend into the white background of the document.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Appendix C: NACOSTI License

REPUBLIC OF KENYA

NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Ref No: 660490

Date of Issue: 30/April/2025

RESEARCH LICENSE

This is to Certify that Mr.. DOMINIC JOHN KAVOI of The Cooperative University of Kenya, has been licensed to conduct research as per the provision of the Science, Technology and Innovation Act, 2013 (Rev.2014) in Nairobi on the topic: AN ENSEMBLE MACHINE LEARNING BASED ALGORITHM TO ENHANCE DETECTION OF ZERO DAY ATTACKS for the period ending : 30/April/2026.

License No: NACOSTI/P/25/4172805

660490

Applicant Identification Number

Director General
NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Verification QR Code

NOTE: This is a computer generated License. To verify the authenticity of this document, Scan the QR Code using QR scanner application.

See overleaf for conditions

Appendix D: Publication

Scientific Research Publishing

Journal of Information Security, 2025, 16(3), 406-436
https://www.scirp.org/journal/jis
ISSN Online: 2153-1242
ISSN Print: 2153-1234

An Ensemble Machine Learning Based Algorithm to Enhance Detection of Zero-Day Attacks: A Comparative Review

Dominic John Kavoi[1], Charles Jumaa Katila[1], Richard Otieno Omollo[2]

[1]School of Mathematics & Computer Science, Cooperative University, Nairobi, Kenya
[2]School of Informatics and Innovative Systems, Jaramogi Oginga Odinga University of Science and Technology, Bondo, Kenya
Email: djcavoi@gmail.com

How to cite this paper: Kavoi, D.J., Katila, C.J. and Omollo, R.O. (2025) An Ensemble Machine Learning Based Algorithm to Enhance Detection of Zero-Day Attacks: A Comparative Review. *Journal of Information Security*, **16**, 406-436.
https://doi.org/10.4236/jis.2025.163021

Received: February 10, 2025
Accepted: July 20, 2025
Published: July 23, 2025

Copyright © 2025 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/
Open Access

Abstract

In the current technological landscape, a lot of risks are present due to the availability of existing and novel kinds of attacks. For these attacks to be countered, systems that identify all the variants without any false positives and false negatives are in high demand. The existence of traditional attack detection methods, such as the signature-based algorithms, has proven that they cannot spot new attacks. This is because they work based on a database that has signatures of attacks. The other methods of detecting attacks that have been explored in this study are the hybrid and machine learning methods for detecting zero-day attacks. In this research, we are coming up with an ensemble set of machine learning algorithms that identify novel and existing attacks in real time from an existing dataset. All of these concepts are mainly based on the Confidentiality, Integrity and Availability (CIA) triad. In order to come up with this, the main method of deployment to be used is the machine learning pipeline. The study has a firm foundation based on theorems such as Bayes and the fundamental principles of computational learning theory. This is composed of stages such as the identification, cleaning, analysis and feature engineering of the data. From there, the ensemble algorithm will be implemented, its accuracy measured and then tuned to improve its efficiency.

Keywords

Zero-Day Attacks, Machine Learning, Ensemble Algorithms, Cybersecurity, Anomaly Detection, Intrusion Detection Systems (IDS), CAN Bus Dataset, Data Analysis