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

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

1. Introduction

1.1. Background

Today, the world is experiencing the transformation of services into digitized models. 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 [1]. A great portion of the population today depends on the internet to undertake activities that are carried out on a daily basis. This makes them susceptible to various cyberattacks and threats [2]. These threats and attacks are propagated and implemented through malicious programs referred to as malware. Malware is an application that is explicitly developed to perform mischievous activities that cause unauthorized access to a computer system, disrupting or damaging the underlying computer system and data [3].

From a global perspective, a lot of research has been done with regard to the various methodologies of artificial intelligence and how they can be used in cybersecurity. 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, ensemble 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 [4].

The detection of malicious activities and the offering of a safe ecosystem against cyberattacks has become a priority for researchers and security firms. A common technique for 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. 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 straightforward remediation approach, known as zero-day malware [5].

"Zero-day attack" is considered as a type of cyber-attack that manipulates software weakness that isn't known to those who would typically handle such issues, such as the software manufacturers or antivirus providers. "Zero-day" means that these vendors have no time to address the vulnerability before it's been exploited,

as they're only made aware of the issue when the attack happens [1]. Thus, they have "zero days" to develop and release a solution or patch to rectify the problem. These weaknesses are the main contributors to the attacks that lead to the compromise of systems in organizations. Another dimension to it is due to the fact that the vendor of some hardware or software that was implemented has limited time to resolve the attack that may have come up. This situation is usually a race against time because vulnerabilities may be released to the public. When this is done, a lot of consequences can be encountered. It has also been established that in order to survive zero-day attacks, detection and prevention strategies are very critical in order to safeguard systems against this vice. In order to carry out the simulations, a segregated environment from the normal technology ecosystem is usually set up. This environment is referred to as a sandbox. It is used as a proof-of-concept strategy for operations used in various modern-day attacks [6].

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 [7]. 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. This motivation has led to a noticeable amount of research, 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. Introduction of intelligence component to existing cybersecurity threat detection strategies, can add a complicated layer of security that can minimize the rate of occurrences of threats if planned correctly [8].

Internet environment safeguarded using signature or rule-based detection systems is 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. Other algorithms based on regression fit numerous linear regression metrics on the data to generate patterns required by ML model to learn from [9]. 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. Combining the strengths of regression-based algorithms with random-based algorithms can yield a hybrid algorithm that explores the concealed relationships between the dataset features

to form a hierarchical model that performs clustering more efficiently, and thus detects zero-day malware more accurately, which is the undertaking that this research seeks to explore [6].

There have been several challenges when it comes to resolving issues tied to joining the machine learning and cybersecurity fields. For instance, it may be difficult to select the best machine learning algorithm that can match the problem to be solved and the dataset at hand. From the technical side, there have 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 cybersecurity 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 vulnerabilities to causative attacks. These attacks interfere with the process by which the algorithms are trained [10].

1.2. Problem Statement

The pre-existing low accuracy of various kinds of algorithms is due to the use of one algorithm as a decision maker as to whether an item is malicious or not. When several algorithms are used in an ensemble, more accuracy is obtained. In the case of signature-based systems, they may be unable to detect new attacks due to the limitation of the new and latest signatures not being introduced to them.

A resultant effect of the low accuracy in the existing singular algorithms would be a high intensity of false positives and false negatives. This would make malicious data be considered benign. At the same time, benign data can be considered as malicious. These wrong results might lead to wrong informed decisions. The false positives and false negatives might also lead to actions that may hinder communication. For instance, packets can be blocked by the firewall(s) that may be dependent on this system through channels such as Application Programming Interfaces (API). Therefore, this study proposes to improve the accuracy of algorithms that detect zero-day and pre-existing attacks through the use of ensemble machine learning algorithms.

1.3. Objectives

1.3.1. Main Objective

The main objective of this research is to develop an ensemble machine learning based algorithm for enhancing the 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 an ensemble algorithm for detecting zero-day attacks.

iii) To implement the ensemble algorithm for detecting zero-day attacks.

iv) To validate the developed algorithm.

2. Critical Analysis of Literature and Research Gaps

2.1. Categories of Attacks in the Cybersecurity Domain

In cybersecurity, 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 1** below [11]. 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 [10].

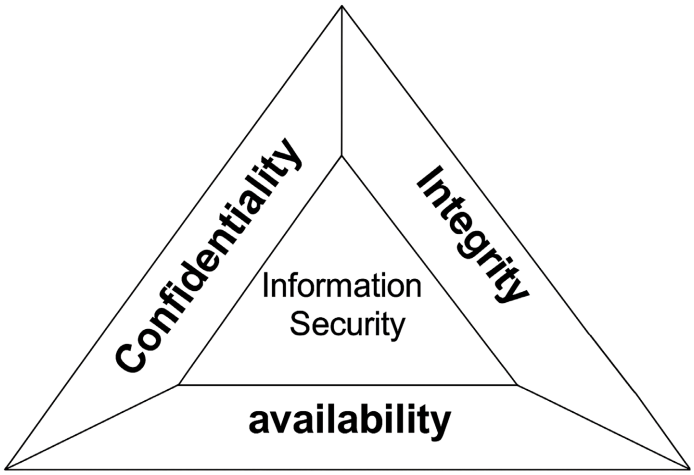

Figure 1. The CIA triad [11].

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 some good amount of time. 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 [12].

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 [13].

2.2. Zero-Day Attacks

Zero-day attacks refer to flaws that were not yet identified in various kinds of software and networks. It is through these zero-day attacks that there may be successful attacks that may lead to various forms of compromise. There has been a lot of dynamism when it comes to changes regarding technological concepts that are used to solve various problems on a day to day basis. Out of this dynamism, there may sprout various fresh attacks that may have not been identified in the recent past. Also, all kinds of attacks may not be fully identified. This is because different attackers may take different approaches in order to solve the objectives that they may have intended to achieve. The thought and execution process of managing the zero-days that have already been identified has also been noted to be a very huge challenge to security professionals. This is due to the fact that most incident response processes are not a one-day event when it comes to their full implementation [14]].

A common trait that has been identified in zero-day attacks is the fact that they leverage on the fact that vendors are not aware of an existing issue within the system that they may be providing to various clients. The complexity comes in when it comes to the confirmed identification and patching of the zero-days by the vendors. The mode of operations of these attacks can take a lot of different shapes. For instance, the zero-day can target the hardware or even the accounts that may be attached to the devices [14].

One of the major traits of zero-day attacks is the fact that they can be able to take up a similar method of operation as compared to other attacks. From there, some modifications can be made in order to take up a novel operation strategy. That is when it may then be able to have a successful attempt on the system that might have been targeted. According to [15], there was a very huge challenge when it came to the identification of polymorphic worms. The attack was considered to be a denial-of-service attacks. This is because resources such as memory in the devices that were subjected to this attack became depleted. These attacks had a tendency of spreading through networks. Through this, the attack was also noted to have the ability to compromise several hosts that may be connected to the same network simultaneously. The dynamism of this attack that made them to be considered to be a zero-day was the fact that they had the ability to operate in new and unique modes. Due to the new formats that were coming up with time, there were different signatures that were being recorded upon every instance of identification. Since the new signature was not recorded in a database or repository of existing attacks, they had the potential of being considered as zero-day attacks.

2.2.1. 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 [16].

The challenge with the research by [16] is that they do not provide an adequately detailed description of the Symantec's 2006-2021 dataset. There needs to be clear validation of the suitability of the dataset to the research that was being conducted. Also, there were no implementation descriptions that were provided with regards to the proposed utilization of the C# detection method for the detection of worms. The performance metrics and the results obtained after the evaluation have not been clearly defined in the research.

2.2.2. 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 [17].

It has been highly observed that the creation of these zero-day attacks have 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 [17].

2.3. Approaches for Detecting Attacks in Cybersecurity

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 [18].

2.3.1. 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. 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 [19].

2.3.2. 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 [20].

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 [21].

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 an ensemble 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 [22].

There was a research done 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 [23].

The described research brought along with it weaknesses that ranged from the unclear explanation of the tuning strategies used for the one-class Support Vector Machine (SVM) algorithm. This is a model that highly relies on tuning strategies such as cross validation and grid search. On the other hand, the algorithm is not well suited for huge datasets. These key performance indicators can be identified during the test phase before it becomes deployed into various real-life scenarios. The dataset used was also not able to fully cover on the modern network traffic patterns. This is because the dataset was available in 2008. This dataset does not cover the network traffic properties that are available today. This is due to the many technological changes that have happened to date [23].

According to a project carried out by [24], it was also established that there is a possibility of carrying out an ensemble 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.

The research does not clearly outline the processing speed to produce the results after it is received in the system. This metrics can be done in time units such as milliseconds and seconds. It does also not outline the latency that would be encountered in case high volumes of traffic such as 10Gbps would be reliant on the system for the identification of security issues. The computation requirements on the hardware side of the equipment needed to operate this program were not

outlined. This would also be anchored on factors such as the volume of traffic that would need to be processed by the system [24].

2.3.3. Hybrid Methods for Attack Detection

In another research, a group 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 [25]. The main dataset that was used in the experimentation phase was the DARPA KDD Cup Dataset 1999. In summary, the main categories of methods that can be used in the detection of attacks are signature, machine learning and hybrid methods as summarized in **Figure 2** below [25]. 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.

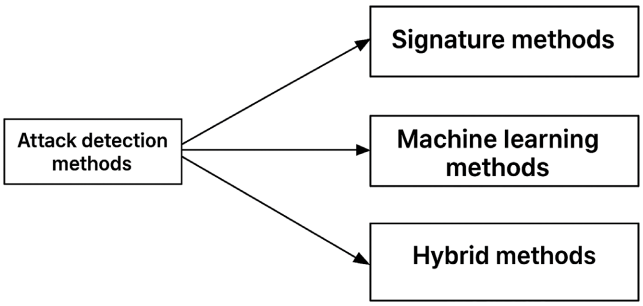

Figure 2. Methods for detecting attacks [25].

2.4. Challenges in Detecting Attacks in Cybersecurity

There is a huge amount 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 [26]. 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 [27].

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 [28].

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 are 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 [28].

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. Thirdly, the new attacks that can be launched to systems may not be in their databases and may be considered as benign traffic that is in transit to and from the systems being secured [29].

2.5. Trends in Machine Learning and Cybersecurity

In the merging of cybersecurity and machine learning algoritms, a lot of projects have been carried out in order to solve specified problems. [23] were trying to use ensemble 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. This is a necessary item to account for because all points that are vulnerable to the infrastructure need to be accounted for their security posture.

Another great project that was implemented involved the use of the Support Vector Machines (SVMs) and Neural Networks (NN) in order to form an ensemble 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). The operational metrics that would outline which data is a facilitator for the attack and those which are not clearly justified in the research.

2.6. Strengths and Weaknesses of the Existing Zero Day Detection Algorithms

Research was carried out by [30]. 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%. This therefore meant that the tool was not stable enough to counter real life scenarios that it was meant to prevent.

According to [3], 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.

The research by [31] 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%. This low rate was an indicator of a system that was not stable enough to be deployed in a real life scenario. The low percentages of the results that were obtained made it inefficient to counter the emerging zero-day attacks and was a major risk if the system was intended to solve some real-life scenarios.

An experiment that was tied to the utilization of several algorithms on the NSL-KDD dataset was done by [32]. 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%.

2.7. Research Gaps

Table 1 below summarizes the gaps that have been identified in the research materials that have been used within the scope of this study:

Table 1. Research gaps.

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. *Journal of King Saud University—Computer and Information Sciences*, 30(1), 2-17	Was not able to identify polymorphic worms
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)	Ensemble 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

3. Research Methodology

The main strategy that will be used in this project is the machine learning pipeline. This workflow will be based on the Python programming language. It is composed

of various stages that work together to solve the problems at hand. It is assumed that the data that will be fed into the system will be raw and unprocessed. After it is loaded, it is preprocessed and transformed in order to be in a standard structure for subjection to the next processes in the machine learning cycle. The data will then be subjected to the process of feature engineering to choose the best features that will be used in training of the machine learning algorithms to be used in the ensemble [33]. Some feature engineering strategies that may be used in this process include StandardScaler, MinMaxScaler, RobustScaler and MaxAbsScaler. If feature selection might also be considered, SelectKBest and VarianceThreshold might the algorithms that may be considered in this process [34].

When the model is ready for subjection to the machine learning process, an ensemble of several algorithms will be used. Hypothetically, algorithms such as random forest, gradient boosting, decision trees, KNearest Neighbour Classifier algorithms might be the ones that will be used in this project [35]. From there, some model evaluation strategies will be used to evaluate the accuracy of the ensemble algorithms with regards to the provision of high accuracy. If the accuracy might be found to be low, further improvements will be done on the algorithm through the use of hyperparameter tuning strategies. Lastly, the machine learning algorithm will be monitored in order to ensure that it has a consistent record of correct results [36].

3.1. Dataset Description

The dataset that will be used in this process is targeted towards investigating cyber attacks that can be relayed to automotive networks in vehicles. The dataset that will be 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 [36].

The dataset is composed of attributes such as data value represented in bytes of range zero to seven, timestamp, Controller Area Network (CAN) identifier, Data Length Code (DLC). For the actual structure of the data to be obtained, it would be important to reverse the preprocessing procedures that were used. The use of Electronic Control Units (ECUs) is common in the technology of cars that are being used today. Though the functionalities are advanced, this has led to the availability of flaws that destruct the bare minimum security measures that would be expected for their operations. The Controller Area Network (CAN) technology lacks robust encryption and authentication mechanisms. Through this flaw, components such as the Universal Serial Bus (USB), Compact Disc (CD) players and On-Board Diagnostics easily become vulnerable interfaces that can be used by attackers to compromise the vehicle according to their desires [37].

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 ensemble machine learning algorithms that will be 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 will be 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.

3.2. Data Collection Instruments

The data that is used in this project is secondary data obtained from the internet. 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 ensemble algorithm to be developed. As for the features that will be used, there exists a lot ways of analyzing and preparing them for being subjected to the machine learning algorithms that will be used.

3.3. 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.

3.4. 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 will be data visualization. This will be mainly done through graphs and plots. In return, this will enable us to get more understanding of how the data is like.

3.5. The Machine Learning Pipeline

The machine learning pipeline is the main strategy that is used in this project. 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:

a) Handling of missing values.

b) Handling of duplicate data.

c) Handling of 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 will be prepared to go through an ensemble machine learning algorithm. The process of feature engineering involves actions such as transformations, feature creation, feature selection and feature extraction.

6) **Subjection to the ensemble algorithm**—The prepared datasets will then be subjected to an ensemble 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 will obtained and then mapped to the original dataset. Further analysis and decision making can be carried out after this process. This mapping will be done by fixing the results obtained from the ensemble algorithm to the datasets that will be used in the project.

8) **Testing of accuracy of the algorithm**—The main testing strategy that will be used in this process will be the inbuilt accuracy metrics algorithms in the Python programming language.

9) **Hyperparameter tuning of the algorithm**—This involves boosting the accuracy of the ensemble algorithm to obtain better performance of the results.

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

The above descriptions can be summarized as shown in Figure 3 below.

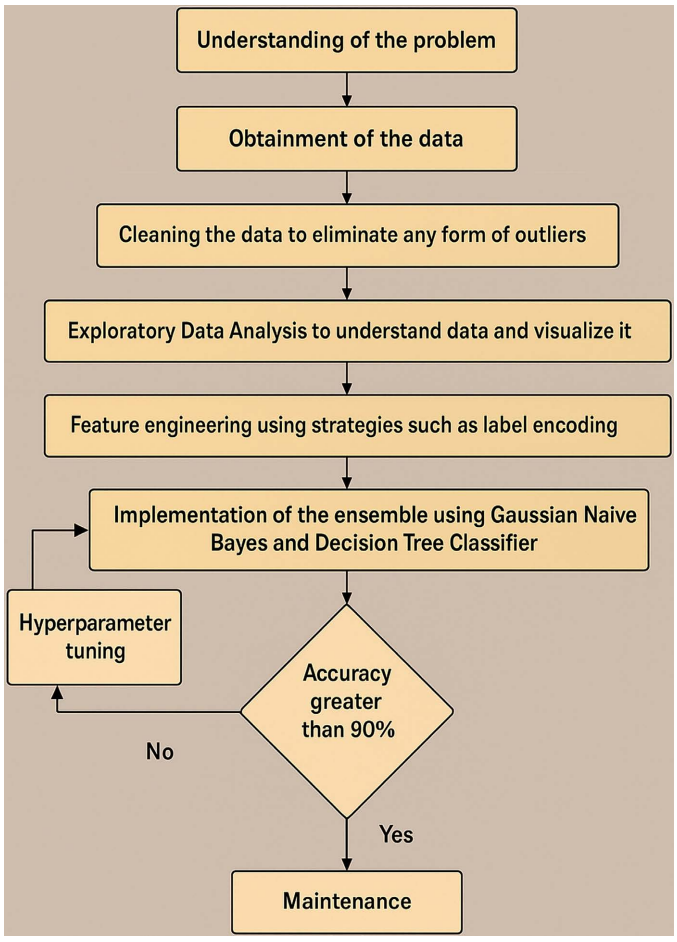

Figure 3. The machine learning pipeline flow chart.

The above process involves the utilization of the Decision Tree Classifier algorithm and the Gaussian Naïve Bayes algorithm for the implementation of the ensemble algorithms. The data being used is the one 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 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 are made in this phase is to have the data being used for training as 70% and that used for testing as 30%. The test size is what is 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.

3.6. Accuracy Measurement

The data is subjected to both the hard and soft ensemble classifiers. 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 are 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.

4. Experimental Results

4.1. Sampling of the Datasets Structure

The datasets were in generally two different formats. The first format is as sampled in Table 2 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:

Table 2. Sample data without the malicious records.

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

Source: Authors' results.

The second dataset as sampled in Table 3 below had one main difference from the initial dataset. This was the fact that it had an extra column which was that of subclass. The categories that were available in the subclass section of the dataset were normal data, replay, fuzzing, flooding, flooding and spoofing attacks.

4.2. Merging of the Datasets

The data that did not have subclasses were merged into a single dataset as sampled below (Table 4, Table 5). Cumulatively, it had a total of 360,032 rows after the merging process:

Table 3. Sample data with the malicious records.

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

Source: Authors' results.

Table 4. Sample data without subclass.

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

Source: Authors' results.

After the merging process, the following sample results were obtained. It had a total of 3,312,119 rows and 6 columns:

Table 5. Merging of the second datasets' structure.

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

Source: Authors' results.

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

4.3. Missing Values

The data does not have any missing values as summarized in Tables 6-8.

Table 6. Sample structure after merging the six datasets.

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

Source: Authors' results.

Table 7. Missing values in the dataset.

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

Source: Authors' results.

4.4. 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:

Table 8. Data types for the columns.

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

Source: Authors' results.

4.5. 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 in Figure 4 below.

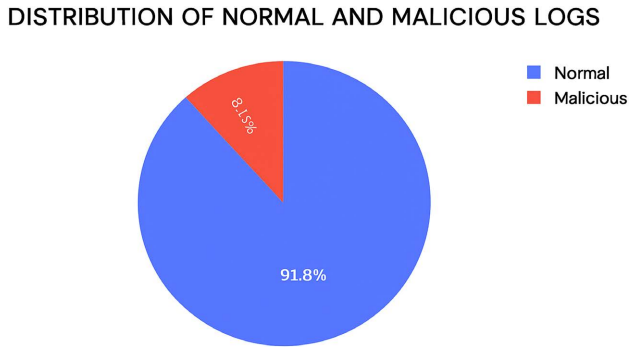

Figure 4. Distribution of attacks in the dataset with subclasses (Source: Authors' results).

The above results can also be expressed as a percentage as summarized in Table 9.

Table 9. Distribution of values as a percentage.

Class	Percentage
Normal	90.96023
Attack	9.039772

Source: Authors' results.

4.6. Analysis of Logs According to Their Subcategories

Table 10 below summarizes how the logs were distributed during the collection process with regards to their specific groups:

Table 10. Distribution of attacks according to their categories.

Attack	Rate
Normal	3,372,743
Flooding	154,180
Fuzzing	89,879
Replay	47,593
Spoofing	7756

Source: Authors' results.

The Figure 5 below shows how the attacks were distributed when combined with the normal logs:

Due to the high count of normal packets, it is necessary to drop values that came from them. Figure 6 below was obtained after the values were dropped:

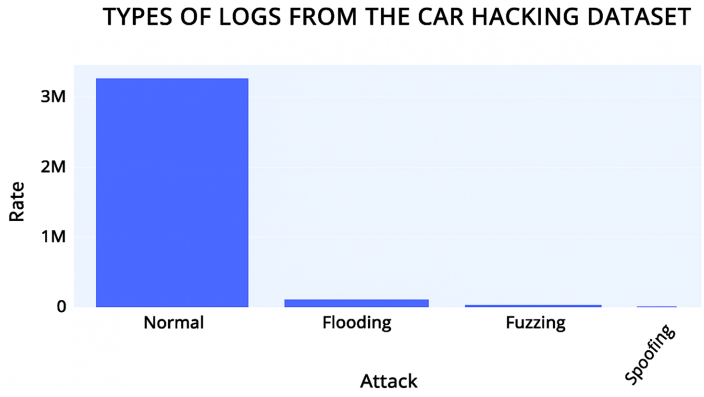

Figure 5. Distribution of all subcategories in the logs (Source: Authors' results).

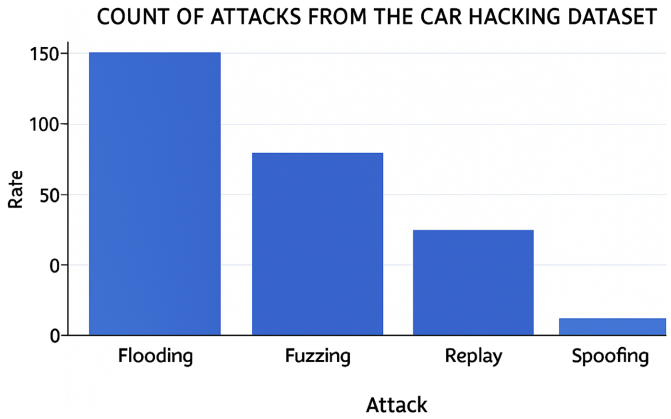

Figure 6. Count of attacks without the normal packets (Source: Authors' results).

4.7. Machine Learning Results

Various accuracy metrics were used to confirm the truthfulness 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% as shown in **Figure 7** below.

```
# print the f-1 scores
print('F1-score of the Decision Tree Classifier
      '{}'.format(np.round(f1_vch,6)*100)
print('F1-score of the Gaussian NB Classifer:
      '{}'.format(np.round(f1_vcs,6)*100)

F1-score of the Decision Tree Classifier: 96.98
F1-score of the Gaussian NB Classifier: 96.9885
```

Figure 7. Accuracy results for the hard and soft classifier (Source: Authors' results).

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.

Out of the data that was subjected to the ensemble machine learning process, the following results were obtained for the four outputs described above (Table 11).

Table 11. Results for the confusion matrix.

Metric	Values
True positive	979,231
True negative	61,604
False positive	27,777
False negative	33,034

Source: Authors' results.

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 1: Precision calculation formula

The above procedure led to the following inputs and results:

979,231/(979,231+27,777) = 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 2: Recall calculation formula

From the above equation, the following values were obtained:

979,231/(979,231 + 33,034) = 0.967366

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 Table 12 below.

Table 12. Accuracy results.

Accuracy Metric	Result
Precision	97.24%
F1-Score	97.00%
Recall	96.74%
Average Score	96.99%

Source: Authors' results.

The final output obtained from the ensemble algorithm is usually in the format

of ones and zeros as summarized in Table 13 below.

Table 13. Sample result before mapping.

Index	Result
864,437	1
113,047	1
789,075	0
2,254,348	1
914,097	0
2,430,204	0

Source: Authors' results.

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 (Table 14).

Table 14. Sample results after carrying out the mapping process.

Index	Result
1,259,672	Malicious
269,962	Benign
2,277,998	Benign
181,758	Benign
1,197,516	Malicious

Source: Authors' results.

After that process, Table 15 below samples the merged structure of what the data was and how the ensemble machine learning algorithms were able to classify them as either malicious or benign:

Table 15. Results of the ensemble algorithm.

Index	Timestamp	Arbitration_ID	DLC	Data	Results
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

Source: Authors' results.

5. Discussion and Future Recommendations

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 minor differences between the predictions and the correct results of the test data, that is summarized in Table 16 below.

Table 16. Difference between original data and predictions.

Category	Original data	Predictions made
Malicious data	1,007,507	1,007,008
Benign data	94,139	94,638
Total	1,101,646	1,101,646

Source: Authors' results.

The second major observation was a very minor difference 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 confluence forms a very strong confirmation of the robustness of the ensemble algorithm implementation.

5.1. Relationship to Original Objectives

The original research objectives that were outlined earlier revolved around 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 are considered as major research questions. After carrying out this study, it is generally concluded that there exists a lot of strength in ensemble algorithms as compared to signature algorithms when it comes to the detection of zero-day attacks. It is also confirmed that the machine learning pipeline would also become an efficient methodology for the design and implementation of the ensemble algorithms that may provide high accuracy to the classifications being made.

5.2. Future Recommendations

This study has formed a very strong background for more unanswered questions. These questions are tied to the Hyperparameter tuning of the currently implemented ensemble algorithm to achieve an accuracy higher than 97% with the same Controller Area Network dataset. Also, more research can be done on simulating a real-time data flow using the same Controller Area Network dataset, or some other dataset with a similar structure. The research would be on whether real-time analytics would be able to perform better as compared to the ensemble algorithms implemented. The results have been able to prove 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 accurate with its high percentage of accuracy.

5.3. Conclusions

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 are zero-day or existing attacks. Also, ensemble algorithms have been proven to perform efficiently without carrying out Hyperparameter tuning on them. Naturally, the ensemble 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 to generate lower accuracy than that which was obtained by the individual or ensemble algorithms.

Another major conclusion is the fact 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 exist 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.

5.4. Recommendations

Following the research done under this study, there is a good number of recommendations that can be derived from it. First, there is a need for the implementation of a real-time ensemble machine learning algorithm for the detection of zero-day attacks. When the same implementation is done but in real time, it can be noted that there may be a higher number of threats that may be obtained. This is because the nature of the ensemble algorithm that has been implemented is to be initialized manually by a person. Once the process is automated and set up strategically, more findings can be obtained, which can also be used as a learning point to improve the accuracy of the machine learning algorithms that have been implemented.

There is also a need to establish how neural networks would be able to perform

in the detection of zero-day attacks, especially with the Controller Area Network dataset. The ensemble of machine learning algorithms was able to generate a good level of accuracy when it comes to this project. Due to the efficiency of neural networks, there exists a chance of them providing slightly better results as compared to those obtained by the ensemble algorithms. If the accuracy may be higher, another important thing to consider is the efficiency in performance when it comes to the execution of the neural networks that assist in the detection of zero-day attacks. Metrics that would be used to measure efficiency would be the processing speed and also the resources that have been used, such as memory and processing power.

Another key recommendation is the establishment of a maintenance framework for this machine learning pipeline, especially when it has been deployed into the real-world environment. This deployment process may need items to be looked into, such as how to handle the algorithm when its performance decreases with time. Also, how the data would be handled in order to maintain a consistently high performance regardless of the huge or small volume of data that may be subjected to it when it comes to the detection of zero-day and currently existing attacks. When this is done, the algorithm might have the ability to maintain a good performance track record.

5.5. Suggestions for Future Research

From this study, the following suggestions may be given with regard to future research that may be derived from this project. Firstly, there is a need for the implementation of a real-time ensemble 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, and 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 time. 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 a need for designing a maintenance framework for this machine learning pipeline, especially when it has been subjected to the real-world environment. This will assist in the provision of consistently high results during its operations.

Acknowledgements

Acknowledgement for the success of this research is to the Department of Computer Science and Information Technology at the Cooperative University of Kenya.

The fraternity provided great support that led to the completion of this research.

Funding Information

The researcher was able to carry out the funding process in all stages of this research successfully. All the resources that were needed for this process were availed.

Author's Contributions

Dominic John Kavoi: Defined the research concept, performed literature review, drafted the initial manuscript. He also managed revisions based on feedback from the supervisors. **Dr. Charles Jumaa Katila:** Offered strategic guidance in developing the research design and methodology, provided valuable insights during data analysis, and played a key role in refining the manuscript through substantial revisions. **Dr. Richard Omolo:** Oversaw the project and offered feedback on the research framework. Contributed to refining the interpretation of the findings and reviewed the final manuscript for submission.

Ethics

The submitted article is an original work and has not been published by anyone else. All ethical issues that might come up were also considered during the course of the work.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Kumar, V. and Sinha, D. (2021) A Robust Intelligent Zero-Day Cyber-Attack Detection Technique. *Complex & Intelligent Systems*, **7**, 2211-2234. https://doi.org/10.1007/s40747-021-00396-9

[2] 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**, e5428. https://doi.org/10.1002/cpe.5428

[3] Hindy, H., Atkinson, R., Tachtatzis, C., Colin, J., Bayne, E. and Bellekens, X. (2020) Utilising Deep Learning Techniques for Effective Zero-Day Attack Detection. *Electronics*, **9**, Article 1684. https://doi.org/10.3390/electronics9101684

[4] Abri, F., Siami-Namini, S., Khanghah, M.A., Soltani, F.M. and Namin, A.S. (2019) Can Machine/Deep Learning Classifiers Detect Zero-Day Malware with High Accuracy? 2019 *IEEE International Conference on Big Data* (*Big Data*), Los Angeles, 9-12 December 2019, 3252-3259. https://doi.org/10.1109/bigdata47090.2019.9006514

[5] 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*), Kochi, 17-20 October 2019, 533-539. https://doi.org/10.1109/tencon.2019.8929620

[6] Al-Rushdan, H., Shurman, M., Alnabelsi, S.H. and Althebyan, Q. (2019) Zero-Day Attack Detection and Prevention in Software-Defined Networks. 2019 *International*

Arab Conference on Information Technology (*ACIT*), Al Ain, 3-4 December 2019, 278-282. https://doi.org/10.1109/acit47987.2019.8991124

[7] Zhou, Q. and Pezaros, D. (2020) A Prediction-Based Model for Consistent Adaptive Routing in Back-Bone Networks at Extreme Situations. *Electronics*, **9**, Article 2146. https://doi.org/10.3390/electronics9122146

[8] 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**, 662-670. https://doi.org/10.34028/iajit/17/4a/11

[9] Lahcen, R.A.M. and Mohapatra, R.N. (2022) Challenges in Cybersecurity and Machine Learning. *PanAmerican Mathematical Journal*, **32**, 14-33.

[10] Rong, C. and Çayirci, E. (2009) Security Attacks in Ad Hoc, Sensor and Mesh Networks. In: Çayirci, E. and Rong, C.M., Eds., *Security in Wireless Ad Hoc and Sensor Networks*, Wiley, 105-120.

[11] Ko, M., Osei-Bryson, K. and Dorantes, C. (2009) Investigating the Impact of Publicly Announced Information Security Breaches on Three Performance Indicators of the Breached Firms. *Information Resources Management Journal*, **22**, 1-21. https://doi.org/10.4018/irmj.2009040101

[12] Kocakulak, M. and Butun, I. (2017) An Overview of Wireless Sensor Networks Towards Internet of Things. 2017 *IEEE 7th Annual Computing and Communication Workshop and Conference* (*CCWC*), Las Vegas, 9-11 January 2017, 1-6. https://doi.org/10.1109/ccwc.2017.7868374

[13] Butun, I., Osterberg, P. and Song, H. (2020) Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures. *IEEE Communications Surveys & Tutorials*, **22**, 616-644. https://doi.org/10.1109/comst.2019.2953364

[14] Ablon, L. and Bogart, A. (2017) Zero Days, Thousands of Nights: The Life and Times of Zero-Day Vulnerabilities and Their Exploits. RAND Corporation. https://www.rand.org/pubs/research_reports/RR1751.html

[15] Kaur, D. and Kaur, P. (2016) Empirical Analysis of Web Attacks. *Procedia Computer Science*, **78**, 298-306. https://doi.org/10.1016/j.procs.2016.02.057

[16] Vaisla, K.S. and Saini, R. (2014) Analyzing of Zero Day Attack and Its Identification Techniques. https://www.researchgate.net/profile/Dr-Kunwar-Vaisla/publication/260489192_Analyzing_of_Zero_Day_Attack_and_its_Identification_Techniques/links/0046353170069a2a06000000/Analyzing-of-Zero-Day-Attack-and-its-Identification-Techniques.pdf

[17] Riofrío, X., Astudillo-Salinas, F., Tello-Oquendo, L. and Merchan-Lima, J. (2021) The Zero-Day Attack: Deployment and Evolution. *Zenodo*, **8**, 39-53. https://doi.org/10.5281/zenodo.5747676

[18] Cardenas, A.A., Baras, J.S. and Seamon, K. (2006) A Framework for the Evaluation of Intrusion Detection Systems. 2006 *IEEE Symposium on Security and Privacy* (*S&P*'06), Berkeley, 21-24 May 2006, 15-77. https://doi.org/10.1109/sp.2006.2

[19] Einy, S., Oz, C. and Navaei, Y.D. (2021) The Anomaly- and Signature-Based IDS for Network Security Using Hybrid Inference Systems. *Mathematical Problems in Engineering*, **2021**, Article ID: 6639714. https://doi.org/10.1155/2021/6639714

[20] Oladipupo, T. (2010) Machine Learning Overview. In: Zhang, Y.G., Ed., *New Advances in Machine Learning*, InTech, 1-12. https://doi.org/10.5772/9374

[21] Oladipupo, T. (2010) Types of Machine Learning Algorithms. In: Zhang, Y.G., Ed., *New Advances in Machine Learning*, InTech, 1-32. https://doi.org/10.5772/9385

[22] Vitorino, J., Andrade, R., Praça, I. and Maia, E. (2021) A Comparative Analysis of Machine Learning Techniques for IoT Intrusion Detection. arXiv: 2111.13149.

[23] Zhuang, W., Ye, Y., Chen, Y. and Li, T. (2012) Ensemble Clustering for Internet Security Applications. *IEEE Transactions on Systems, Man, and Cybernetics, Part C* (*Applications and Reviews*), **42**, 1784-1796. https://doi.org/10.1109/tsmcc.2012.2222025

[24] Limthong, K. (2013) Real-Time Computer Network Anomaly Detection Using Machine Learning Techniques. *Journal of Advances in Computer Networks*, **1**, 1-5. https://doi.org/10.7763/jacn.2013.v1.1

[25] Pandeeswari, N. and Kumar, G. (2015) Anomaly Detection System in Cloud Environment Using Fuzzy Clustering Based Ann. *Mobile Networks and Applications*, **21**, 494-505. https://doi.org/10.1007/s11036-015-0644-x

[26] MIT Lincoln Laboratory (2023) 1999 DARPA Intrusion Detection Evaluation Dataset. https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset

[27] Shaukat, K., Luo, S., Varadharajan, V., Hameed, I.A. and Xu, M. (2020) A Survey on Machine Learning Techniques for Cyber Security in the Last Decade. *IEEE Access*, **8**, 222310-222354. https://doi.org/10.1109/access.2020.3041951

[28] Modi, C., Patel, D., Borisaniya, B., Patel, H., Patel, A. and Rajarajan, M. (2013) A Survey of Intrusion Detection Techniques in Cloud. *Journal of Network and Computer Applications*, **36**, 42-57. https://doi.org/10.1016/j.jnca.2012.05.003

[29] Ibrahim Hairab, B., Aslan, H.K., Elsayed, M.S., Jurcut, A.D. and Azer, M.A. (2023) Anomaly Detection of Zero-Day Attacks Based on CNN and Regularization Techniques. *Electronics*, **12**, Article 573. https://doi.org/10.3390/electronics12030573

[30] Holm, H. (2014) Signature Based Intrusion Detection for Zero-Day Attacks: (Not) a Closed Chapter? 2014 47*th Hawaii International Conference on System Sciences*, Waikoloa, 6-9 January 2014, 4895-4904. https://doi.org/10.1109/hicss.2014.600

[31] Li, Z., Qin, Z., Shen, P. and Jiang, L. (2019) Zero-Shot Learning for Intrusion Detection via Attribute Representation. In: Gedeon, T., Wong, K. and Lee, M., Eds., *Neural Information Processing. ICONIP* 2019, Springer International Publishing, 352-364. https://doi.org/10.1007/978-3-030-36708-4_29

[32] Reazul, M., Rahman, A. and Samad, T. (2017) A Network Intrusion Detection Framework Based on Bayesian Network Using Wrapper Approach. *International Journal of Computer Applications*, **166**, 13-17. https://doi.org/10.5120/ijca2017913992

[33] d'Aloisio, G., Marco, A.D. and Stillo, G. (2022) Modeling Quality and Machine Learning Pipelines through Extended Feature Models. arXiv: 2207.07528.

[34] Olson, R.S. and Moore, J.H. (2018) Identifying and Harnessing the Building Blocks of Machine Learning Pipelines for Sensible Initialization of a Data Science Automation Tool. In: Riolo, R., Worzel, B., Goldman, B. and Tozier, B., Eds., *Genetic Programming Theory and Practice XIV*, Springer, 211-223. https://doi.org/10.1007/978-3-319-97088-2_14

[35] Kaur, A. and 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**, 2-17. https://doi.org/10.1016/j.jksuci.2016.04.002

[36] Dupont, G., Lekidis, A., Hartog, J.D. and Etalle, S. (2019) Automotive Controller Area Network (CAN) Bus Intrusion Dataset v2. https://data.4tu.nl/articles/dataset/Automotive_Controller_Area_Net-

work_CAN_Bus_Intrusion_Dataset/12696950/2

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