National College of Ireland

Utilising Artificial Intelligence in Enhancing Zero-Day Attacks Detection

MSc Research Project
Cybersecurity

Mudiaga Agbroko
Student ID: x23207485

School of Computing
National College of Ireland

Supervisor: Kamil Mahajan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Mudiaga Agbroko
Student ID:	X23207485
Programme:	Cybersecurity **Year:** 2024
Module:	Research Project
Supervisor:	Kamil Mahajan
Submission Due Date:	
Project Title:	Utilising Artificial Intelligence in Enhancing Zero-Day Attacks Detection
Word Count:	**Page Count:** 17

5334

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Mudiaga Agbroko

Date: 11/12/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project	□

is lost or mislaid. It is not sufficient to keep a copy on computer.	

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Utilising Artificial Intelligence in Enhancing Zero-Day Attacks Detection

Mudiaga Agbroko
x23207485

Abstract

With the alarming increase in zero-day attacks and the limitations facing current traditional intrusion detection systems, enhancing zero-day attack detection is paramount. This research proposes the use of artificial intelligence algorithms in improving the detection of zero-day attacks. Three supervised machine learning algorithms were employed to evaluate the detection capability of machine learning models compared to traditional intrusion systems. The study was conducted by assessing the performance of Snort, an open-source intrusion detection/prevention system, Decision Tree Classifier, K-Neighbor Classifier, and Random Forest Classifier on the Canadian Institute for Cybersecurity Intrusion Detection Evaluation Dataset (CIC-IDS2017). To improve the performance of the machine learning algorithms, the features were standardised, the dataset's dimension reduced, and sampling techniques used in attaining a balanced dataset class. The Decision Tree Classifier, K-Neighbor Classifier, and Random Forest Classifier had an accuracy of 0.904, 0.929, and 0.919 respectively. The Decision Tree Classifier had the fastest runtime of 0.006 seconds and the highest processing rate, processing 150,000 entries per second.

1 Introduction

Threat actors exploit previously unknown vulnerabilities in a system's firmware, hardware, or software to carry out zero-day attacks (Dempsey *et al.*, 2018). The unknown nature of these vulnerabilities gives IT professionals little to no time to patch and address the weaknesses, hence the name zero-day. Google reported a 50 percent increase in zero-day vulnerabilities being publicly exploited in 2023 than in 2022 (Maddie Stone and Sadowski, 2024). Top technology giants like Microsoft, Apple, Cisco, VMware, Barracuda, and others have all disclosed zero-day flaws in their products (Wright, 2024). The exploit of the zero-day vulnerability on the MOVEit file transfer software utilised by thousands of enterprises globally led to the sensitive data breach of several organisations (National Cyber Security Centre, 2024).

The danger of zero-day intrusions is the ability of cybercriminals to carry out activities in the victim's system undetected (Tshuva, 2024). These activities can range from privilege escalation, data infiltration, deploying command and control, ransomware execution, etc. Like other intrusion attacks, zero-day attacks are detected using an intrusion detection system (IDS).

An IDS that can either be a hardware or software system detects malicious activities in a system and sends notifications so the intrusion can be investigated (Hashemi-Pour and Lutkevich, 2024). They can be network-based or host-based with the network based employed with a network and the host-based deployed on devices and computer systems. IDS employs signature-based or anomaly-based techniques in detecting intrusions.

1.1 Research Problem

Signature-based IDS holds a database of malicious signatures, raising an alarm when an intrusion matches a signature in its database (Sabahi and Movaghar, 2008). It is very effective in detecting known attacks patterns but unsuitable for zero-day attacks. Anomaly-based IDS, on the other hand, detects intrusion by studying the normal patterns and regular activities of a system and sends an alert when there's a deviation from these patterns and activities. The primary downside of anomaly-based IDS is its high false positive rate, classifying benign activities as malicious (Bai and Kobayashi, 2003). This research is conducted to address the limitations of traditional IDS, employing artificial intelligence (AI) algorithms to improve the detection of zero-day attacks.

1.2 Research Question

The primary question this research seeks to address is: How well can AI algorithms enhance the detection of zero-day attacks?

1.3 Research Objective

In addressing the research question, the main aim of this project was to evaluate the performance of AI models and traditional IDS in detecting intrusions. Due to the absence of real zero-day attack datasets, the CIC-IDS2017 dataset (Sharafaldin, Lashkari and Ghorbani, 2018) was selected for this research. The Snort open-source IDS was selected as the traditional IDS, while Random Forest, Decision Tree, and K-Neighbors classifiers were chosen as the AI algorithms.

1.4 Research Structure

The rest of this report is structured as follows: The literatures reviewed are outlined in **Related Work**; **Research Methodology** discusses the procedure followed in employing AI algorithms in the detection of zero-day attacks; **Design Specification** documents techniques and software employed in implementing the research. **Implementation** captures how the research is implemented. The research results are discussed and analysed in the **Evaluation** Section, and the **Conclusion and Future Work** summarise and conclude the entire research project.

2 Related Work

This section critically analyses related works to this project and is divided into machine learning-based and deep learning-based detection models. The literatures reviewed provided insights into the current state, limitations, and gaps in the research domain.

2.1 Machine Learning-Based Detection Models

To effectively detect zero-day attacks while reducing false positives, Pitre *et al.* (2022) proposed an IDS framework that combined a machine learning hybrid model, ensemble feature selection, and dataset fine-tuning techniques. The hybrid model comprises the logistic regression and SVM algorithms employed on the CSE-CIC-IDS2018 dataset. The Linear regression model had an accuracy of 93% and improved accuracy 95%. The downside of the proposed model is its complexity.

Alfoudi *et al.* (2022) attempted to address the high false positive rates of machine learning based intrusion detection systems (IDS) that result from the imbalance nature of the training dataset. They proposed a hybrid algorithm based on Enhanced Density-Based Spatial Clustering of Applications with Noise (EDBSCAN) to monitor network activities and accurately detect intrusions while reducing computational complexity. The model improves the quality of the training datasets making them more balanced to enable effective attack detection even in small instances. The NSL-KDD and UNSW-NB15 datasets were employed in training and testing the model, outperforming the K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Random Forest algorithms. The complexity of manually selecting its features makes the model prone to error and time-consuming.

In an attempt to effectively distinguish between regular connections and intrusions, an Isolation Forest based network IDS was developed by S, G and Priya (2022) and tested on the NSL-KDD dataset, attaining an accuracy of 87.70. The proposed technique, which was efficient in processing huge volumes of data, was limited by the difficulty of selecting an appropriate threshold without the knowledge of the outlier type.

To improve the efficiency of intrusion detection while minimising the loss of relevant data, Vishwakarma and Kesswani (2023) proposed a two-phase Intrusion Detection System based on machine learning algorithms. The initial stage involves grouping data into four clusters using their datatypes and categorizing them using various Native Bayes classifiers. The final phase involves further classifying the normal data using the elliptic envelope unsupervised machine learning algorithm. The proposed technique was validated using the NSL-KDD, UNSW_NB15, and CIC-IDS2017 datasets, attaining an accuracy of 97%, 86.9%, and 98.59%, respectively, with low false positive rates. The downside of the proposed model is its poor performance when employed in multiclass classification.

An effective Machine Learning anomaly-based IDS was introduced by Shanthi and Maruthi (2023) to eliminate the challenge associated with detecting abnormal activities in cyber

environment due to the huge volumes of data. The Isolation Forest and Support Vector Machine models were employed and validated on the NSL-KDD dataset using the accuracy, F1-Score, and recall performance evaluation metrics. The Isolation Forest performed better in terms of accuracy and F-Score while the SVM model had better recall. Regardless of the performance of the models, their accuracy was influenced by how the variables were selected and their dimensionality.

The difficulty in effectively detecting zero-day attacks led Sarhan *et al.* (2023) to introduce a zero-shot learning framework employed in analysing the performance of Machine Learning based Intrusion Detection Systems. The Random Forest and Multi-Layer Perceptron models were employed on the UNSW-NB15 and NF-UNSW-NB15-v2 datasets. In preventing bias while training the models, network traffic flow variables were dropped. Label encoding was also used to convert categorical features to numeric values. The models performed very well in most intrusion classes but inaccurately detected other intrusion groups as zero-day attacks.

Hossain and Islam (2023) proposed an ensemble Machine Learning based IDS to address the challenges in building an effective and stable IDS. The proposed model was employed using various ensemble methods like Adaboost, Bagging, Gradient Boosting & XGBoost, Random Forest, and Simple Stacking on several datasets like NSL-KDD, KDDCUP, CICIDS, UNSW-NB, etc. The Random Forest based model outperformed the other techniques in terms of accuracy and detection rates. The limitations of the proposed technique are the huge computational power and time required in training and validating the models.

Gebremariam, Panda and Indu (2023) developed an advanced IDS built on a hybrid machine learning algorithm to enhance the accuracy of intrusion detection while minimising the false positive rates. The proposed hybrid model employed Genetic Algorithm-Artificial Neural Network, which was trained and tested using the CICIDS2017, NSL-KDD, and CICIDS2017 datasets. When evaluated on the CICIDS2017 dataset, the proposed model performed excellently attaining an accuracy, CLK-Means, F1-Score, precision, and recall of 99.82%, 100%, 99.85%, 99.91%, and 99.82% respectively. The model faced various limitations, such as computationally expensive, high complexity, etc.

Elsaid and Binbusayyis (2024) built an IDS upon an Optimized Isolation Forest algorithm was proposed to improve intrusion detection accuracy, F1-Score, and precision while minimising training time. The performance of the model was evaluated on the CICIDS-2018, NSL-KDD, and UNSW-NB15. The model attained accuracy, precision, and recall of 95.6%, 98.5%, and 95.8%, respectively, on the NSL-KDD dataset. Regardless of the model's performance, the downside is the high computational time.

To improve the accuracy and efficiency of anomaly-based IDS, Elsaid and Binbusayyis (2024) developed an ensemble Machine Learning model that uses Isolation Forest and Autoencoders. The framework combines the Isolation Forest's anomaly detection ability and the autoencoders efficient learning capabilities. When combined, the algorithms performed effectively and

reliability in detecting intrusions with an exceptional rate of stability. However, rigorous testing and validation of the model is required for optimal performance.

G *et al.* (2024)To improve the accuracy and efficiency of anomaly-based IDS, an ensemble Machine Learning model that uses Isolation Forest and Autoencoders was developed. The framework combines the Isolation Forest's anomaly detection ability and the autoencoders efficient learning capabilities. When combined, the algorithms performed effectively and reliability in detecting intrusions with an exceptional rate of stability. However, rigorous testing and validation of the model is required for optimal performance.

Touré *et al.* (2024) developed a zero-day attack detection system that proactively detects previously unknown threats on a network system, a hybrid learning framework that combined unsupervised and supervised Machine Learning algorithms was proposed. The framework evaluated on the IBM and the NSL-KDD datasets attained an accuracy of 98.4% and 96.6% respectively while tremendously reducing the rate of false detection. For the proposed model to perform effectively, rigorous training and validation are required.

Sharma, Rani and Driss (2024) proposed a machine learning algorithm that integrates Genetic Algorithm, Decision Tree, and Support Vector Machine for the development of a robust Intrusion Detection System. The proposed system was evaluated using the UNSW NB15 and Bot-IoT datasets, attaining an accuracy of 92.06% and 96.12% respectively. The model relied mainly on the Genetic Algorithm making its performance very slow.

To meet the demands for a state-of-the-art IDS capable of effectively detecting zero-day attacks, Gowthami and Priscila, (2024) proposed a hybrid model that integrates supervised and unsupervised machine learning algorithms. When compared to an existing detection system, the proposed model attained a higher detection accuracy with a lower false positive rate. Although the proposed model performed well, it wasn't scalable in a large network environment.

To effectively detect network intrusion, Giraddi *et al.* (2024) proposed a machine learning framework comprising of Decision Tree, Support Vector, Naïve Bayes, and Artificial Neural Network algorithms was proposed. The Support Vector algorithm had the highest accuracy of 99%, while the Artificial Neural Network had the best efficiency and speed. The framework wasn't trained and validated on a robust dataset making it prone to a high false detection rate when employed in a real-world environment.

Wategaonkar *et al.* (2024) combined behavioural analytics and Support Vector Machine classifier were combined to improve the effectiveness of zero-day attacks detection. The proposed framework attained an accuracy, recall, and F1-score of 0.93, 0.94, and 0.93 respectively. Regardless of the framework's performance, its scalability limitations must be addressed before adoption in a real-world scenario.

Dai *et al.* (2024) combined to enhance the detection of zero-day intrusions, autoencoders were with Random Forest and XGBoost. The technique performed with accuracy by leveraging autoencoders to obtain relevant features during the training phase, making the model suitable in real-world scenarios. The proposed technique is limited by its huge resource requirement and scalability issues.

Patel *et al.* (2024) proposed a novel approach was to address the scalability concerns of Bayesian networks in detecting zero-day intrusions. The technique divided a huge Bayesian network into subsets to reduce the computational time. When evaluated, the proposed technique showed a lower mean error and execution time compared to an individual large network. The proposed model is limited by its complexity.

2.2 Deep Learning-Based Detection Models

Teymourlouei, Stone and Jackson (2023) employed a Deep Learning algorithm-based IDS which is a subset of Machine Learning to enhance the detection of zero-day attacks .The model was employed on the MTA-KDD dataset, which was divided into a ratio of 70:30 for training and testing respectively attaining an accuracy of over 97% and a 0.1 loss. The model was limited by its complexity and high false positive rates.

A Deep Learning based IDS was proposed by Soltani *et al.* (2023) to eliminate the challenges of effectively detecting zero-day attacks. The model was evaluated using the CIC-IDS2017 and CSE-CIC-IDS2018 datasets, attaining an accuracy over 99% in the majority of the attacks. The additional time and cost required in labelling the training dataset limits the proposed technique

The Dynamic Long Short-Term Memory-based anomaly detection model was employed by Arun, Nair and Sreedevi (2024) to improve the detection of zero-day attacks. The proposed model was evaluated on the CICIDS2017 and the NSL-KDD datasets. They performed well by detecting anomalies quickly, effectively, and efficiently. The model's extensive training and validation requirement is a notable downside.

2.3 Summary

The most common limitations faced by the related works reviewed include the complexity of the model, huge computational resource requirements, and extensive training and testing of models. The best performing models are those proposed by Soltani *et al.* (2023), Gebremariam, Panda and Indu (2023) and Giraddi *et al.* (2024) with an accuracy of 99%, 99.82%, and 99% respectively.

3 Research Methodology

This section discusses the methodology employed in the research project, it's divided into 3 subsections: Dataset, traditional IDS and machine-learning based IDS.

3.1 Dataset

The Canadian Institute for Cybersecurity (CIC) intrusion detection dataset (CIC-IDS2017) (Sharafaldin, Lashkari and Ghorbani, 2018) was employed in evaluating the traditional IDS and ML-based IDS. The dataset that comprises both benign and cyber-attack traffic is available in both PCAP and CSV formats. In generating the dataset, live data was captured from 9 AM Monday, July 3, 2017, to 5 PM, Friday, July 7, 2017. The malicious traffic in the dataset includes botnet, brute force, DoS, DDoS, Heartbleed, infiltration, port scan and web attacks. The protocols captured comprise email, FTP, HTTP, HTTPS, and SSH protocols. The dataset was downloaded manually from the CIC webpage, excluding Monday because it contained just benign traffic. The CIC-IDS2017 dataset PCAP files were used on the traditional IDS while the CSV files were used to train and test the machine learning models. Table 1 shows an overview of the traffic captured in the CIC-IDS2017 dataset.

Table 1: An overview of the traffic captured in the CIC-IDS2017 dataset

Day	Traffic Class
Monday	Benign
Tuesday	Benign FTP-Patator SSH-Patator
Wednesday	Benign DoS slowris DoS Slowhttptest DoS Hulk DoS GoldenEye Heartbleed
Thursday	Benign Web Attack -Brute Force Web Attack – XSS Web Attack – Sql Injection Infiltration Port Scan
Friday	Benign Botnet ARES Port Scan DDoS LOIT

3.2 Traditional IDS

Snort, an open-source signature-based IDS/IPS which is able to analyse real-time traffic and log packets was employed as the traditional IDS in this research. Intrusions are detected using rules that tell Snort what to do when a condition is met. Snort rules are structured into 6 components: action, proto, source, dir, dest, and body (The Snort Team, 2021). The action instructs Snort on what to do when a signature is matched, proto specifies the protocol, source the source IP address and source port, dir specifies if it's unidirectional or bi-directional, dest specifies destination IP address and port, while the body contains information about detection

and other details. Snort rules are either evaluated as the signature's full evaluation or a fast pattern query.

3.2.1 Snort Processing

Snort processing is done following the following steps as shown in Figure 2:

- Packet: The CIC-IDS2017 PCAP files are merged into a single PCAP file using Wireshark's mergecap to enable Snort to process it effectively.
- Decode: Snort examines the individual PCAP packets to determine their characteristics, including the IP address and port. It also examines other components of each packet for anomalies.
- Pre-process: The examined packets are pre-processed to ascertain the message content by reorganising and repackaging the fragments/segments.
- Detect: Detection is done by using rules to search for patterns in the
- Log: Relevant information gotten from the previous phases is logged as alerts.
- Verdict: The final phase is making decisions based on the logs generated.

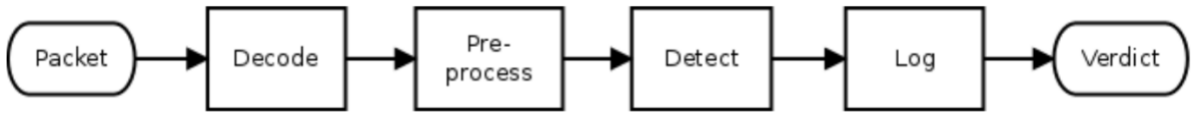

Figure 1: Snort processing steps (The Snort Team, 2021**).**

3.3 Machine Learning-based IDS

Since the research was about enhancing zero-day attack detection, which involves classifying intrusions into normal or abnormal. Three supervised machine learning classifiers, including Decision Tree, Random Forest, and K-Neighbors classifiers were employed. Supervised machine learning utilises labelled datasets in making classification and prediction. Figure 2 shows an overview of the methodology employed in implementing the machine learning algorithms.

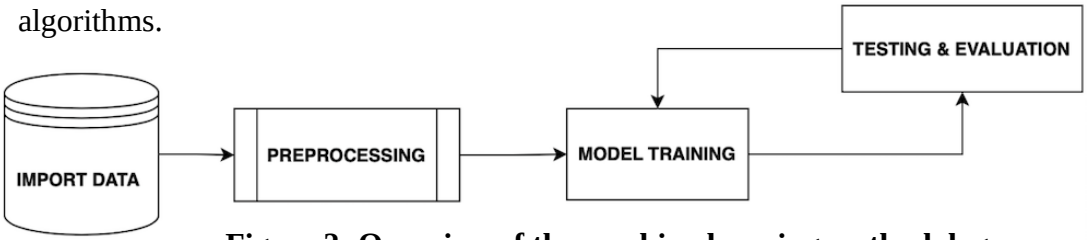

Figure 2: Overview of the machine learning methodology.

3.3.1 Data Collection

The initial phase was data collection where the machine learning CSV file of the CIC-IDS2017 was loaded as a dataframe into the machine learning environment. The Tuesday, Wednesday, Thursday, and Friday CSV files were used because they contained both benign and malicious traffic. The imported dataset contained 79 columns and 2,300,825 rows, with over 75% of the

traffic benign. Exploratory data analysis provided an insight into the traffic classification of the CIC-IDS2017 dataset as shown in Figure 3.

BENIGN	1743179
DoS Hulk	231073
PortScan	158930
DDoS	128027
DoS GoldenEye	10293
FTP-Patator	7938
SSH-Patator	5897
DoS slowloris	5796
DoS Slowhttptest	5499
Bot	1966
Web Attack Brute Force	1507
Web Attack XSS	652
Infiltration	36
Web Attack Sql Injection	21
Heartbleed	11

Figure 3: An overview of the imported data traffic classification.

3.3.2 Data Cleaning & Preprocessing

Upon loading the dataset, the next step was to clean and preprocess the data into a suitable format to be used on the machine learning models, thereby optimising the overall performance. Data cleaning was done by removing whitespaces from column names, removing duplicates, handling missing and infinite values, and grouping similar traffics together. The missing and null values were handled by using the mean imputation technique. Features with just a unique value were also dropped due to their irrelevancy towards the research.

Upon cleaning the data, the huge size of the dataset and the number of features was still an issue. The features excluding the target which is the traffic label were standardised using StandardScaler techniques for normalisation and to improve the performance of the models (Sharma, 2024). To address the size of the datasets, the principal component analysis dimensional reduction technique was used to simplify the dataset and minimise the computational requirement. To prevent bias in the models' performance due to the imbalanced nature of the target variable, undersampling and oversampling techniques were employed to ensure the traffic classes are equally distributed.

3.3.3 Models' Training & Testing

After cleaning and pre-processing the dataset, the next step was to train and test the supervised machine learning models. The dataset set was divided into 80% and 20% for training and testing the models respectively. K-Neighbors classifier with 7 numbers of neighbors, Decision Tree classifier with a max depth of 10, and Random Forest classifier with 10 numbers of estimators, max-depth of 10, and a random state of 42 were employed.

3.3.4 Evaluation

In assessing the performance of machine learning models, several evaluation metrics are used. Some of these metrics include confusion matrix, precision, sensitivity, accuracy, F1 score, etc. The confusion matrix is used to determine the precision of a classification algorithm, precision specifies the ratio of the correctly predicted items, sensitivity calculates the algorithm's ability to effectively predict true positives, accuracy is the proportion of correct predictions made by the model, and the F1 Score is employed in assessing the effectiveness of an imbalanced dataset. In this research, the accuracy is used in evaluating the performance of the models in accurately detecting the attacks. Other metrics employed include the runtime, which measures the time duration it takes to make predictions, and the throughput, which is the number of samples processed per second.

4 Design Specification

The entire research project was conducted using the MacBook Pro 2017, 2.3 GHz Dual-Core Intel Core i5 processor, Intel Iris Plus Graphics 640, 8GB memory and 256 GB storage. Other techniques, software, etc used are discussed briefly below:

- Mac Terminal: Environment for deploying Snort and running other software.
- CIC-IDS2017: The dataset employed.
- Homebrew: The package manager used to install Wireshark, Snort, Jupyter Notebook, and their dependencies.
- Mergecap (Wireshark): For merging the dataset PCAP files into a single PCAP file.
- Snort: For processing the dataset PCAP file and generating alert logs.
- Snort Community Rules: Employed in configuring Snort for detection.
- Microsoft Excel: For initial exploration of CSV files,
- Python: The programming language employed.
- Jupyter Notebook: Environment for running ML models and for analysing Snort CSV logs.
- Dataframe: Pandas data structure for storing the CSV datasets and Snort alert logs on Jupyter Notebook.
- NumPy: Python library for performing numerical operations.
- Pandas: Python-based data analysis library for working with datasets.
- Matplotlib.pyplot: Interactive library for visualisation.
- Seaborn: Python-based data visualisation library.
- StandardScaler: For standardising features.
- PCA: For reducing the dimension of the dataset.
- SMOTE: For oversampling minority features.
- Train_Test_Split: For splitting dataset.
- .Fit: For training models.
- .Predict: For testing models.
- Accuracy_Score: For evaluating the models using the accuracy.
- Time: For measuring the runtime.

- DecisionTreeClassifier: For importing the Decision Tree model.
- KNeighborsClassifier: For importing the K-Neighbor Model.
- RandomForestClassifier: For importing the RandomForestModel.

5 Implementation

The research project was implemented by using Snort to process the CIC-IDS2017 PCAP files and the ML models trained and tested on the CIC-IDS2017 CSV files. The PCAP files were merged using Mergecap with the -w flag to enable Snort to process it on the Mac Terminal. Snort was configured using the community rules and logs generated in CSV format. The generated alert log was exported to Jupyter Notebook for analysis and visualisation. The Machine learning algorithms were trained and tested using the CIC-IDS2017 ML CSV files on the Jupyter Notebook environment. The scikit-learn Python module was used to import the machine learning models into the Jupyter Notebook environment.

6 Evaluation

This results and findings for evaluating both Snorts and the machine learning models on the CIC-IDS2017 datasets are discussed in the section.

6.1 Evaluation of Snort on the CIC-IDS2017 PCAP Files

To check the performance of Snort in effectively detecting intrusion attacks, it was evaluated on the CIC-IDS PCAP files. Snort processed the merged PCAP files using the community rules to generate logs in CSV format and processing statistics on the console. Snort processed a total of 44,660,731 packets with 266,327 alerts generated, which is 0.6% of the entire PCAP file. The runtime was 14 minutes and 37 seconds, with a throughput of 50,887 packets/sec and 255 Mbits/sec. The most attacked class detected is ‘Attempted Administrator Privilege Gain’ of 175,425.

6.2 Machine Learning Models on the CIC-IDS2017

The Decision Tree, K-Neighbor, and Random Forest supervised learning classifiers were employed on the CIC-IDS2017 dataset in ML CSV format to determine their effectiveness in accurately detecting intrusions. The models had an accuracy of 0.904, 0.929, and 0.919 respectively. With a runtime of 0.006 seconds, 1.256 seconds, and 0.015 seconds respectively. The Decision Tree Classifier had a higher processing rate followed by the K-Nearest Neighbor Classifier and the Random Forest Classifier.

6.3 Discussion

The result from the research shows that the machine learning models outperformed the traditional IDS in detected intrusions using the CIC-IDS2017 dataset. Snort was only able to capture 0.6% of malicious compared to 24.2% of malicious traffic in the original dataset as

shown in Figure 4. Snort wasn't able to accurately identify the malicious traffic, mislabeling them as shown in Figure 5. Also, the traditional IDS had the slowest runtime and was only able to process more entries than the KNN classifier, as shown in Figure 6. The KKN classifier had the best accuracy compared to the other ML models as shown in Figure 7, but had the slowest runtime and smallest processing rate. With an accuracy of 0.904, alongside the highest processing rate and fastest runtime, the Decision Tree Classifier is considered the best performing model.

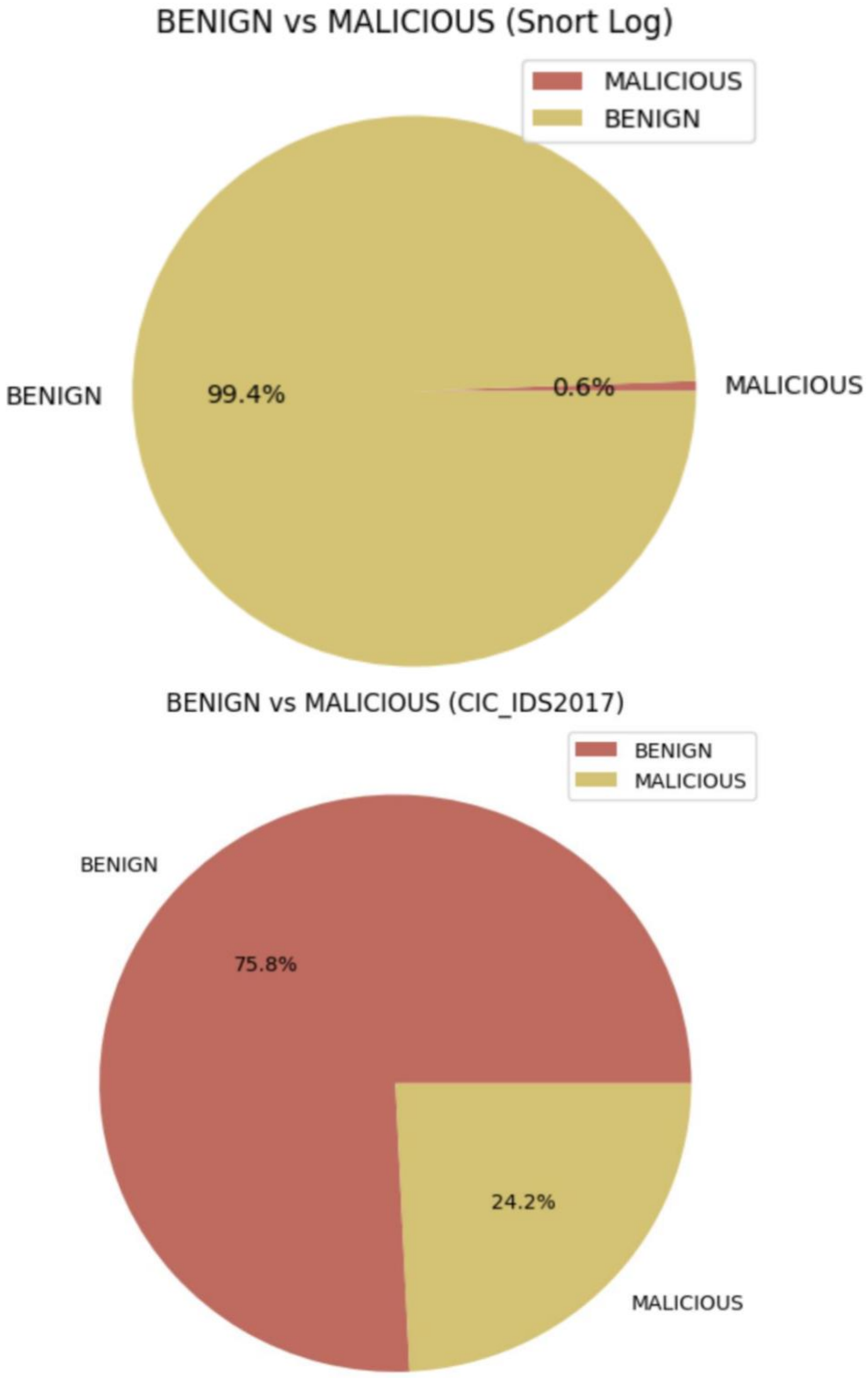

Figure 4: Malicious traffic captured by Snort compared to the original dataset.

```
Traffic Classification
Attempted Administrator Privilege Gain                          175425
Potentially Bad Traffic                                          84029
Misc activity                                                     3240
Detection of a Network Scan                                       1359
Attempted Information Leak                                        1043
Information Leak                                                   441
Executable code was detected                                       383
Attempted Denial of Service                                        314
Access to a potentially vulnerable web application                  67
Web Application Attack                                               9
Successful Administrator Privilege Gain                              5
A system call was detected                                           4
An attempted login using a suspicious username was detected          3
Misc Attack                                                          2
A suspicious string was detected                                     2
Generic Protocol Command Decode                                      1
Name: count, dtype: int64
```

Figure 5: Snort mislabelling the malicious traffic captured.

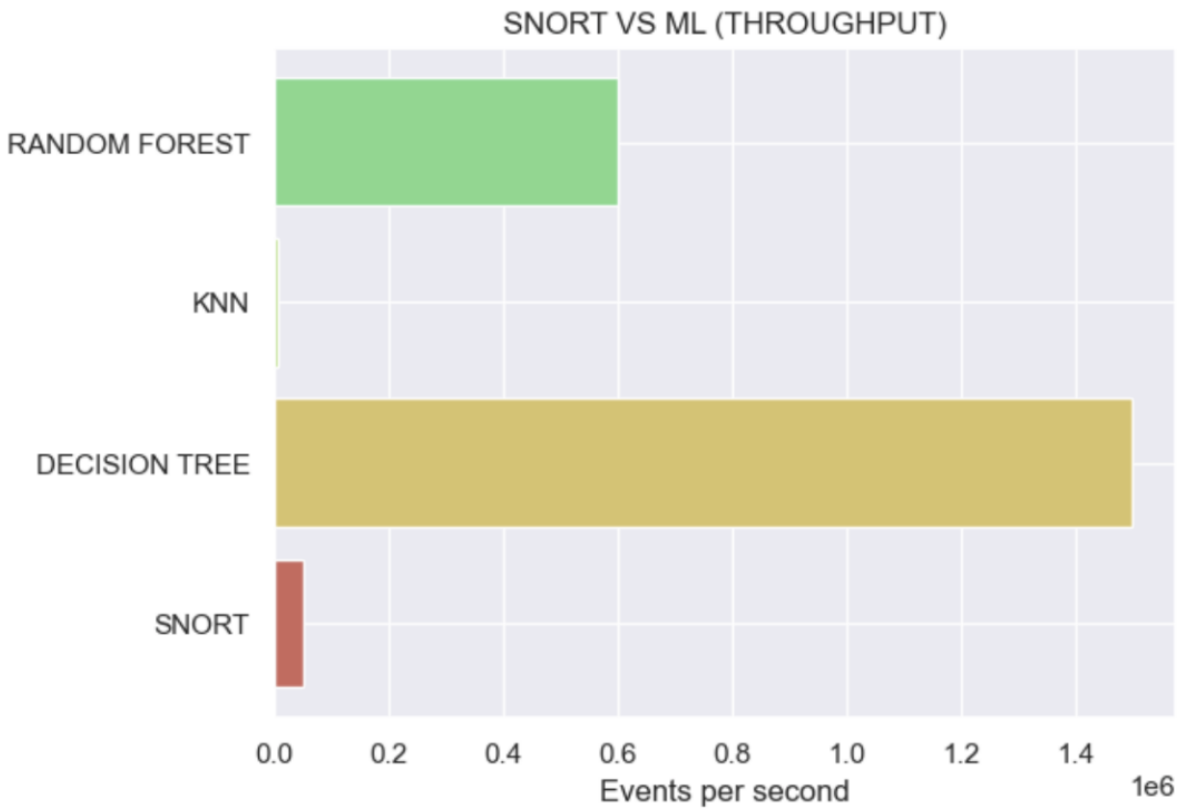

Figure 6: Processing rate of Snort and ML models.

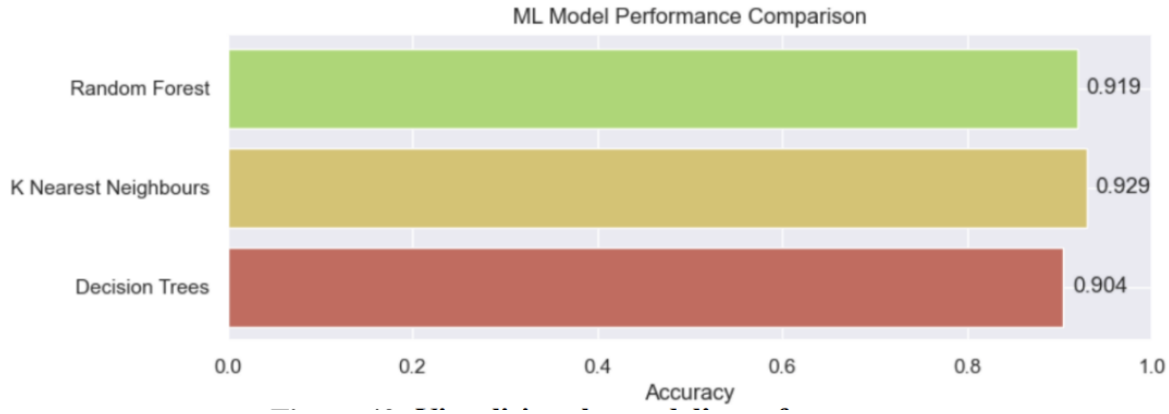

Figure 7: Evaluating the accuracy of the ML models.

7 Conclusion and Future Work

To determine how well AI algorithms can enhance the detection of zero-day attacks, the effectiveness of traditional IDS and AI models was evaluated on the CIC-IDS2017 dataset. Snort IDS/IPS was employed on the CIC-IDS2017 dataset PCAP files while the Decision Tree Classifier, K-Neighbor Classifier, and Random Forest Classifier were used on the ML CSV files. Snort was only able to capture 0.6% of malicious traffic compared to 24.2% in the original dataset; it also had the slowest runtime of over 14 minutes and the 2nd smallest processing rate. Decision Tree Classifier, K-Neighbor Classifier, and Random Forest Classifier attained an accuracy of 0.919, 0.929, and 0.904 respectively. The Decision Tree Classifier had the fastest runtime and highest processing rate compared to the other models and Snort.

Future work will be to fine-tune the ML models to improve their accuracy and employ more datasets to capture more comprehensive attacks. A hybrid IDS framework that combines both Snort and ML models will also be explored in enhancing zero-day attack detection. The framework will utilise Snort in signature-based detection and ML in anomaly-based detection.

References

Alfoudi, A.S. *et al.* (2022) 'Hyper clustering model for dynamic network intrusion detection', *IET Communications*, p. cmu2.12523. Available at: https://doi.org/10.1049/cmu2.12523.

Arun, A., Nair, A.S. and Sreedevi, A.G. (2024) 'Zero Day Attack Detection and Simulation through Deep Learning Techniques', in *2024 14th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. *2024 14th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, India: IEEE, pp. 852–857. Available at: https://doi.org/10.1109/Confluence60223.2024.10463429.

Bai, Y. and Kobayashi, H. (2003) 'Intrusion Detection Systems: technology and development', in *17th International Conference on Advanced Information Networking and Applications, 2003. AINA 2003. 17th International Conference on Advanced Information Networking and Applications. AINA 2003*, Xi'an, China: IEEE Comput. Soc, pp. 710–715. Available at: https://doi.org/10.1109/AINA.2003.1192972.

Dai, Z. *et al.* (2024) ‘An intrusion detection model to detect zero-day attacks in unseen data using machine learning’, *PLOS ONE*. Edited by A.O. Ibrahim, 19(9), p. e0308469. Available at: https://doi.org/10.1371/journal.pone.0308469.

Dempsey, K. *et al.* (2018) *Automation support for security control assessments: software asset management*. NIST IR 8011-3. Gaithersburg, MD: National Institute of Standards and Technology, p. NIST IR 8011-3. Available at: https://doi.org/10.6028/NIST.IR.8011-3.

Elsaid, S.A. and Binbusayyis, A. (2024) ‘An optimized isolation forest based intrusion detection system for heterogeneous and streaming data in the industrial Internet of Things (IIoT) networks’, *Discover Applied Sciences*, 6(9), p. 483. Available at: https://doi.org/10.1007/s42452-024-06165-w.

G, M. *et al.* (2024) ‘An Ensemble Framework for Network Anomaly Detection Using Isolation Forest and Autoencoders’, in *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*. *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, Chennai, India: IEEE, pp. 1–6. Available at: https://doi.org/10.1109/ADICS58448.2024.10533499.

Gebremariam, G.G., Panda, J. and Indu, S. (2023) ‘Design of advanced intrusion detection systems based on hybrid machine learning techniques in hierarchically wireless sensor networks’, *Connection Science*, 35(1), p. 2246703. Available at: https://doi.org/10.1080/09540091.2023.2246703.

Giraddi, V. *et al.* (2024) ‘Machine Learning Approach to Intrusion Detection: Performance Evaluation’, *Procedia Computer Science*, 235, pp. 1851–1859. Available at: https://doi.org/10.1016/j.procs.2024.04.176.

Gowthami, G. and Priscila, S.S. (2024) ‘Zero-Day Threat Detection A Machine Learning Paradigm for Intrusion Prevention’, in *2024 7th International Conference on Circuit Power and Computing Technologies (ICCPCT)*. *2024 7th International Conference on Circuit Power and Computing Technologies (ICCPCT)*, Kollam, India: IEEE, pp. 852–857. Available at: https://doi.org/10.1109/ICCPCT61902.2024.10672858.

Hossain, Md.A. and Islam, Md.S. (2023) ‘Ensuring network security with a robust intrusion detection system using ensemble-based machine learning’, *Array*, 19, p. 100306. Available at: https://doi.org/10.1016/j.array.2023.100306.

Maddie Stone and Sadowski, J. (2024) ‘A review of zero-day in-the-wild exploits in 2023’, 27 March. Available at: https://blog.google/technology/safety-security/a-review-of-zero-day-in-the-wild-exploits-in-2023/ (Accessed: 11 December 2024).

National Cyber Security Centre (2024) ‘MOVEit vulnerability and data extortion incident’, *NCSC*, 27 June. Available at: https://www.ncsc.gov.uk/information/moveit-vulnerability (Accessed: 11 December 2024).

Patel, R.N. *et al.* (2024) ‘Exploring Scalable Bayesian Networks For Identification of Zero-day Attack Paths’, in *2024 Silicon Valley Cybersecurity Conference (SVCC)*. *2024 Silicon Valley Cybersecurity Conference (SVCC)*, Seoul, Korea, Republic of: IEEE, pp. 1–8. Available at: https://doi.org/10.1109/SVCC61185.2024.10637360.

Pitre, P. *et al.* (2022) ‘An Intrusion Detection System for Zero-Day Attacks to Reduce False Positive Rates’, in *2022 International Conference for Advancement in Technology (ICONAT)*. *2022 International Conference for Advancement in Technology (ICONAT)*, Goa, India: IEEE, pp. 1–6. Available at: https://doi.org/10.1109/ICONAT53423.2022.9726105.

S, S., G, S. and Priya, B. (2022) ‘Network Intrusion Detector Based On Isolation … Forest Algorithm’, in *2022 1st International Conference on Computational Science and Technology (ICCST)*. *2022 1st International Conference on Computational Science and Technology (ICCST)*, CHENNAI, India: IEEE, pp. 932–935. Available at: https://doi.org/10.1109/ICCST55948.2022.10040395.

Sabahi, F. and Movaghar, A. (2008) ‘Intrusion Detection: A Survey’, in *2008 Third International Conference on Systems and Networks Communications*. *2008 Third International Conference on Systems and Networks Communications*, Sliema, Malta: IEEE, pp. 23–26. Available at: https://doi.org/10.1109/ICSNC.2008.44.

Sarhan, M. *et al.* (2023) ‘From zero-shot machine learning to zero-day attack detection’, *International Journal of Information Security*, 22(4), pp. 947–959. Available at: https://doi.org/10.1007/s10207-023-00676-0.

Shanthi, K. and Maruthi, R. (2023) ‘Machine Learning Approach for Anomaly-Based Intrusion Detection Systems Using Isolation Forest Model and Support Vector Machine’, in *2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA)*. *2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India: IEEE, pp. 136–139. Available at: https://doi.org/10.1109/ICIRCA57980.2023.10220620.

Sharafaldin, I., Lashkari, A.H. and Ghorbani, A.A. (2018) ‘Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization’, in *4th International Conference on Information Systems Security and Privacy (ICISSP)*. *ICISSP 2018*, Portugal.

Sharma, A., Rani, S. and Driss, M. (2024) ‘Hybrid evolutionary machine learning model for advanced intrusion detection architecture for cyber threat identification’, *PLOS ONE*. Edited by R. Singh, 19(9), p. e0308206. Available at: https://doi.org/10.1371/journal.pone.0308206.

Sharma, P. (2024) ‘The Ultimate Guide to 12 Dimensionality Reduction Techniques (with Python codes)’, *Analytics Vidhya*, 7 November. Available at: https://www.analyticsvidhya.com/blog/2018/08/dimensionality-reduction-techniques-python/ (Accessed: 12 December 2024).

Soltani, M. *et al.* (2023) ‘An adaptable deep learning-based intrusion detection system to zero-day attacks’, *Journal of Information Security and Applications*, 76, p. 103516. Available at: https://doi.org/10.1016/j.jisa.2023.103516.

Teymourlouei, H., Stone, D. and Jackson, L. (2023) ‘Identifying Zero-Day Attacks with Machine Learning and Data Reduction Methods’, in *2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE)*. *2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE)*, Las Vegas, NV, USA: IEEE, pp. 2285–2290. Available at: https://doi.org/10.1109/CSCE60160.2023.00372.

The Snort Team (2021) ‘Snort 3 User Manual’.

Touré, A. *et al.* (2024) ‘A framework for detecting zero-day exploits in network flows’, *Computer Networks*, 248, p. 110476. Available at: https://doi.org/10.1016/j.comnet.2024.110476.

Tshuva, N. (2024) *Zero Day Vulnerabilities, Attack Examples, Detection and Prevention, Sternum IoT*. Available at: https://sternumiot.com/iot-blog/zero-day-vulnerabilities-attack-examples-detection-and-prevention/ (Accessed: 11 December 2024).

Vishwakarma, M. and Kesswani, N. (2023) ‘A new two-phase intrusion detection system with Naïve Bayes machine learning for data classification and elliptic envelop method for anomaly detection’, *Decision Analytics Journal*, 7, p. 100233. Available at: https://doi.org/10.1016/j.dajour.2023.100233.

Wategaonkar, S.R. *et al.* (2024) ‘Targeting Insider Threats and Zero-Day Vulnerabilities with Advanced Machine Learning and Behavioral Analytics’, in *2024 4th International Conference on Innovative Practices in Technology and Management (ICIPTM)*. *2024 4th International Conference on Innovative Practices in Technology and Management (ICIPTM)*, Noida, India: IEEE, pp. 1–6. Available at: https://doi.org/10.1109/ICIPTM59628.2024.10563816.

Wright, R. (2024) *10 of the biggest zero-day attacks of 2023 | TechTarget, Search Security*. Available at: https://www.techtarget.com/searchsecurity/feature/10-of-the-biggest-zero-day-attacks-of-2023 (Accessed: 11 December 2024).