Title: Protocol Anomaly Detection Using Decision Trees Machine Learning Algorithm in a Simulated Network

Kate Asante, 22037983

Abstract: In an era marked by the exponential expansion of the internet, an increased importance of strengthening network security against emerging threats has become a serious need. The rise of attacks, which could result in data breaches and the violation of user privacy, emphasizes the importance of strong defensive measures. Distribution Denial of Service (DDoS) assaults have emerged as a major issue among these threats. Cyberattacks on network protocols have become increasingly sophisticated, necessitating the enhancement of defense systems. In this context, the Anomaly Detection System (ADS) is critical in minimizing intrusions by detecting data pattern abnormalities that could indicate potential cyber threats. Our research proposes an implementation of a protocol anomaly detection using the supervised decision trees machine learning algorithm in a simulated network which mimic real-world instances. The widely used NSL-KDD dataset was used to train and test the decision tree model. We adopted the C4.5 method of the decision tree technique with criterion set as entropy, maximum-depth of 20 and minimum number of split of 10. The decision tree model after training and testing gave an accuracy of 99.98% and 80.0% after the model was integrated into a simulated network which outperforms most of the existing works done in the area of network intrusion detection. Our approach has shown a great potency in predicting anomalies on the network of a computer as one of the few simulated-based anomaly detection systems.

Keywords: protocol anomaly detection; decision trees; network intrusion detection system; cyberattacks; distribution of denial of service(DDoS); simulated network; machine learning; datasets; cybersecurity.

1. Introduction

In an era marked by the exponential expansion of the internet, the imperative to fortify network security against emerging threats has reached critical proportions. Almost all of the attacks can lead to data breaches, jeopardizing user privacy. One of the most popular attacks that organizations face is the Distribution Denial of Service (DDOS) attacks (Wighneswara et al., 2023). Cyberattacks, often targeting network protocols, have grown increasingly sophisticated, necessitating the development of robust defence mechanisms. Among these, the Network Intrusion Detection System (NIDS), stands out as a pivotal shield against malicious software intrusions. Operating through the analysis of data patterns, NIDS serves as an early sentinel, swiftly identifying deviations that could signal potential cyber threats (Al-Khassawneh 2023).

The difficulty of discovering patterns in data that do not match expected behaviour is mostly referred to as anomaly detection (Nassif et al., 2021). Anomaly detection system keeps track of a system's behaviour and alerts the user when there are substantial deviations from the usual (Muniyandi et al., 2011). Anomaly detections are implemented in a wide range of applications like fraud detection, loan application processing, and medical status monitoring as well as cyber security intrusion detection, fault detection for aviation safety research and many others (Xu et al., 2019).

Fundamentally, NIDS, which operates with the core objective of categorizing incoming data into benign or malicious segments, serving as an intelligent gatekeeper for network security, has been utilized for years (Samrin and Vasumathi 2017). However, conventional NIDS systems face a set of limitations. Their ability to maintain a consistently

high level of detection accuracy is often compromised, leaving organizations susceptible to novel and evolving threats that evade established patterns (Catillo et al., 2023).

Recently, Cybersecurity has ensured the use of computational intelligence techniques through the use of machine learning, deep learning, and data mining among others. Notwithstanding the notable progress in the use of computational intelligence techniques, as well as upcoming improvements in performance, robustness against cyber-security attacks, and insights into malicious samples and attacks, computational intelligence in cybersecurity still needs to advance significantly, in addition to overcoming many challenges, such as zero-day attacks. Furthermore, there is growing worry regarding the security and weaknesses of Machine Learning algorithms in the face of attacks as well as the challenge of early detection of anomalies on computer networks (Dasgupta et al., 2020).

Enhancing the intelligence and adaptiveness of machine learning techniques to addresses the said problems and improve upon robustness of anomaly detection, our research paper presents an implementation of a protocol anomaly detection using Decision Trees (DT) algorithm in a simulated network environment which closely mimics real-world scenarios. Although Decision Trees has been widely used in intrusion detection, the tool has not been used in a simulated environment to explore a deeper comprehension of it. Commonly, anomalies and malicious behaviours are noted in Transmission Control Protocol (TCP), Internet Control Message Protocol (ICMP) reports and User data Protocol (UDP) (Wighneswara et al., 2023). Our research will explore with the new version of data set of the KDD'99 data set popularly known as 'NSL-KDD' to train and test a Decision Tree machine learning algorithm. This system, honed within the simulated network, will learn to distinguish normal network behaviour from anomalies.

Our paper tends to contribute to the area of NIDS in three folds:

- With a primary objective of implementing a machine learning model based on Decision Trees in a simulated network to mimic a real-world network conditions for protocol anomaly detection. The simulated environment will enable the testing of the developed model in controlled scenarios, facilitating the evaluation of its accuracy and efficiency in detecting protocol anomalies.

- Our paper aims to comprehensively evaluate the performance of the Decision Trees-based protocol anomaly detection system on metrics such as accuracy, precision, recall, and F1-score.

- Furthermore, to identify the limitations and challenges of the Decision Trees model and conduct a comparative analysis to assess the effectiveness of the proposed approach in comparison to existing methods.

This paper is categorized into six parts. Part one is an introduction, part two is a literature review, part three details methods used, part four is the result, part five is evaluation and discussion and part six is the conclusion.

2. Related Work

In addressing the challenge of anomaly detection approaches in computer network, T. Kim and W. Pak (2022) argued that a good number of anomalies bypass detection in most machine learning-network intrusion detection systems (ML-NIDS). In order to curb this limitation, they introduced a machine learning based system which focuses on an efficient early classification and traits analysis through the use of representation features. While the paper presents a promising solution through early classification and learning trait analysis, it does not delve deeper into the characteristics and patterns of these elusive anomalies and the model does not satisfy for a comprehensive analysis. The authors made use of the ISCX2012, CIC-IDS2017, and IDS2018 datasets (datasets sourced from

Sharafaldin et al., 2018) however, the proposed research makes use of the NSL-KDD dataset which provides a broader range of attack variations and encompasses different network environments and attack vectors.

Subsequently, Wighneswara et al. (2023) presented a model by utilizing supervised machine learning techniques for intrusion detection systems, with the objective of advancing protocol anomaly detection. Their paper was aimed at detecting deviations in network traffic harnessing the decision tree algorithm. Through the exploration of an oversampling technique and the use of the entropy criterion and max depth of 20 parameters, their model achieved an impressive accuracy of 99.95%. Similarly, with the same objective, Vinod et al. (2023) also introduced an innovative h-BOASOS hybrid system, utilizing and integrating supervised and unsupervised techniques for intrusion detection. While Wighneswara et al (2023) and Vinod et al. (2023) both made significant contributions, the performance of the decision trees and the hybrid model could be influenced by overfitting and computational complexities as opposed to our proposed research. In contrast to this, Bowen et al. (2023) suggested BLoCNet, another hybrid deep learning model intertwining unsupervised convolutional neural network (CNN) and bidirectional long short-term memory (BLSTM) layers, effectively identifying disturbing patterns in computer networks. BLoCNet's ability to capture spatial and temporal features presents a robust solution, however, the models were trained on the CIC-IDS2017, UNSW-NB15 and IOT-23 datasets. These datasets are limited in scope as compared to the NSL-KDD dataset. CIC-IDS2017 focuses on Botnet traffic, while UNSW-NB15 and IOT-23 focus on IoT network traffic.

In order to elevate detection of anomalies with advanced and comprehensive techniques Al-Khassawneh (2023) meticulously explored the effectiveness of machine learning classifiers on NSL-KDD dataset, illustrating the nuances of intrusion detection. While their paper emphasizes the potency of the random forest algorithm in efficiently identifying deviations and malicious activities, there is an absence of an in-depth exploration of the Decision Trees machine learning algorithm for intrusion detection. Although the study considers K-nearest neighbor, support vector machine, and random forest algorithms, it overlooks the potential contributions of Decision Trees in this context. Correspondingly, Talukder et al., 2022 introduced a promising hybrid model that leverages a diverse range of techniques to achieve remarkable accuracy and robustness. Their research showcased the performance of the model through data preprocessing and ensemble techniques. Although these studies display the dynamic landscape of intrusion detection, embracing advancements in machine and deep learning by reflecting early classification, the approaches do not hold for all kinds of traits analysis. Our proposed paper presents a deeper exploration of specific traits analysis which adds novelty to enhance broader applicability to bolster network security.

Saba et al. (2022) ventured into IoT intrusion detection with the aim of tackling complex network intrusion scenarios with precision. Using convolutional neural networks, the authors demonstrated a substantial accuracy network in detecting suspicious behaviour using NID and BOT-IoT datasets. In contrast, Dsouza et al. (2022) implemented a real-time network intrusion detection by evaluating multiple machine learning algorithms with decision trees emerging as the top parameter against KDD dataset. The studies experimented in different network contexts - IoT and real-time intrusion detection and maintain the same focus of navigating complicated network intrusion instances. However, the studies lack a deeper investigation into the network latency for real-time detection and examination of the scalability of the CNN-based approach.

Ozcam et al (2021) and Li et al. (2020) both introduced research with the aim of enhancing detection by unveiling intricacies in intrusion detection systems. Ozcam et al (2021) delved into identifying DDoS attacks particularly TCP-Flood, employing unsupervised learning methods like isolation forest and k-means clustering. Their approach effectively leveraged the power of decision trees in anomaly detection which gave them an excellent accuracy result using an unlabelled dataset which is noteworthy. By focusing on

TCP-SYN flood attack type through TCP flags and TTL times, the research narrowed down the scope for more precise detection. Similarly, Li et al. (2020) explored the performance of an unsupervised learning approach in anomaly detection by focusing on Border Gateway Protocol (BGP) anomalies resulting from specific malicious attacks: Slammer worm, WannaCrypt ransomware, and Moscow Blackout. The proposal of a recurrent neural network (RNN) for detection signifies a forward-thinking approach. Also, the use of updated messages from reputable datasets like Reseaux IP Europeans and Route views adds credibility to the study. Both studies contribute significantly to the field of network security by applying advanced techniques to detect specific anomalies. However, they underscore the need for a more granular analysis of the decision-making process within the employed algorithms and the potential for hybrid models.

To put our paper's significance into context, the given critical examination of related works reveals intriguing avenues (Sharma-Wallace et al., 2018). The effectiveness of Decision Trees, showcased by Wighneswara et al (2023) offers a starting point to heighten the accuracy of anomaly detection (Guezzaz et al., 2021). The comparisons made so far underscore the need for a cohesive framework that integrates algorithmic precision. Our research extends the narrative by embedding Decision Trees within a simulated network environment to effectively navigate the ever-shifting landscape of anomalies. Also, by integrating a simulated network space, it enhances a more accurate detection of anomalies, making intrusion detection models more adaptable to real-world scenarios while mitigating overfitting and promoting computational efficiency. It is of this paper's aim to inherit the strengths of the discussed related works but also extends its capabilities by integrating dynamic feature enhancement, positioning it as a more potent and versatile solution for detecting anomalies in a network traffic (Heidari et al., 2022).

3. Methods

The proposed method designed for our research, as shown in Figure 1, is categorized into two tiers. In Tier 1, the acquired NSL-KDD dataset is analysed and undergoes preprocessing to achieve a clean data (that is to check for duplicate and null values). Afterwards, the data is transformed and scaled into numerical and binary representations through the use of label encoding and one-hot encoding methods. Label encoding is used to convert ordinal data into numerical form and one-hot encoding into binary representation (Yu et al., 2020), after which the cleaned data is used to build the decision tree model and final results for the model are achieved. Tier 2 encompasses creation of the simulated environment, capturing of network traffic, and integration of the built decision tree model. After the environment is set, we explore with the use of Distributed Internet Traffic Generator (D-ITG) tool to capture traffic, after which the built decision tree model is integrated into the environment against the captured traffic to get the final results.

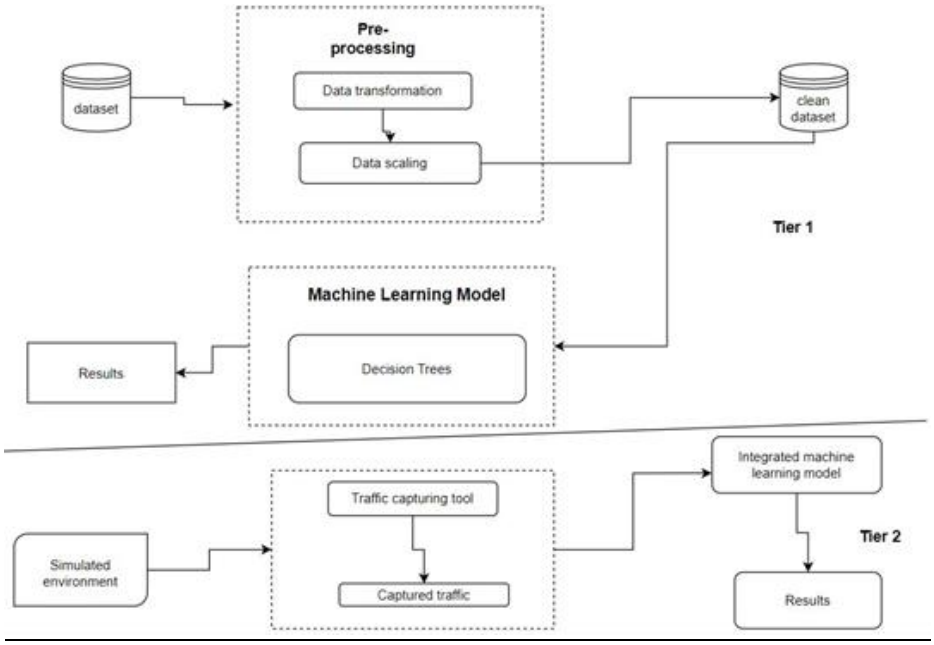

Figure 1. System architecture of proposed method.

3.1. Dataset Description

The datasets used to train and test the Decision Tree algorithm for our research is the network intrusion detection dataset, NSL-KDD dataset sourced from Kaggle, as came from Tavallaee et al., 2009 which details the updated version of KDD'99 dataset.

The NSL-KDD dataset consists of train and test sets which exclude redundant records from the train set to ensure that, the classifiers are not biased towards more frequent records. The quantity of the records in the train data has 125973 datapoints with 42 features and the test data has 22544 datapoints with 42 features with both consisting of normal and attack data (Tavallaee et al., 2009). Out of the 37 attacks that are contained in the test dataset, 21 are included in the training dataset. The training dataset contains the known attack types, while the test dataset contains additional attacks that are not present in the training datasets. The attacks are categorized into four types namely, denial of service attack (DOS), remote to local attack (R2L), user to root attack (U2R) and probing attack (Ingre and Yadav, 2015).

Table 1 below shows the major attack types in the training and testing dataset (Revathi and Malathi, 2013).

- DOS (Denial of Service): These attacks aim to make a service or network resource unavailable to users by bombarding the target system with a flood of requests, causing it to slow down or crash.

- Probing Attacks: These attacks are attempts to collect information about a target system or network for potential vulnerabilities. They do not necessarily aim to cause direct harm but rather seek weaknesses for possible future attacks.

- R2L (Remote to local): These attacks involve unauthorized access to a system using legitimate user credentials. The attacker gains unauthorized access by guessing, cracking, or stealing valid credentials.

- U2R (User to Root): These attacks involve an unauthorized user gaining superuser privileges on a system. The attacker starts as a regular user and then exploits vulnerabilities to escalate privileges.

Table 1. Classification of attacks for training and testing datasets.

Attacks	Attack Types
DOS	Neptune, Pod, Back, Land, Smurf, Teardrop, Processtable, Udpstorm, Apache2, Worm, Mailbomb
PROBING	Satan, IPsweep, Nmap, Portsweep, Mascan, Saint
R2L	Guess_password, Ftp_write, Phf, Warezmaster, Xsnoop, Xlock, Imap, Snmpgetattack, Httptunnel, Snmpguess, Named, Sendmail, Multihop
U2R	Xterm, Buffer_overflow, Loadmodule, Rootkit, Perl, Ps, Sqlattack

Figures 2 and 3 below provide a visual representation of the study of the NSL KDD dataset and display the quantity of the various attack types for both training and testing datasets.

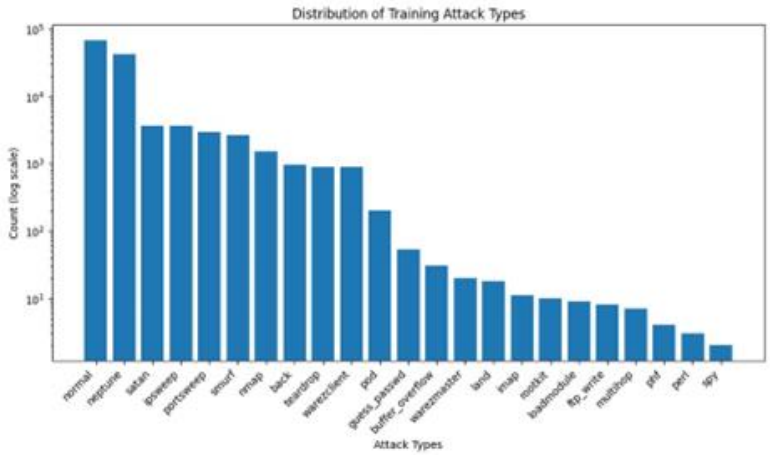

Figure 2. Visual Representation of Training Dataset: Differentiating Normal and Attack Data.

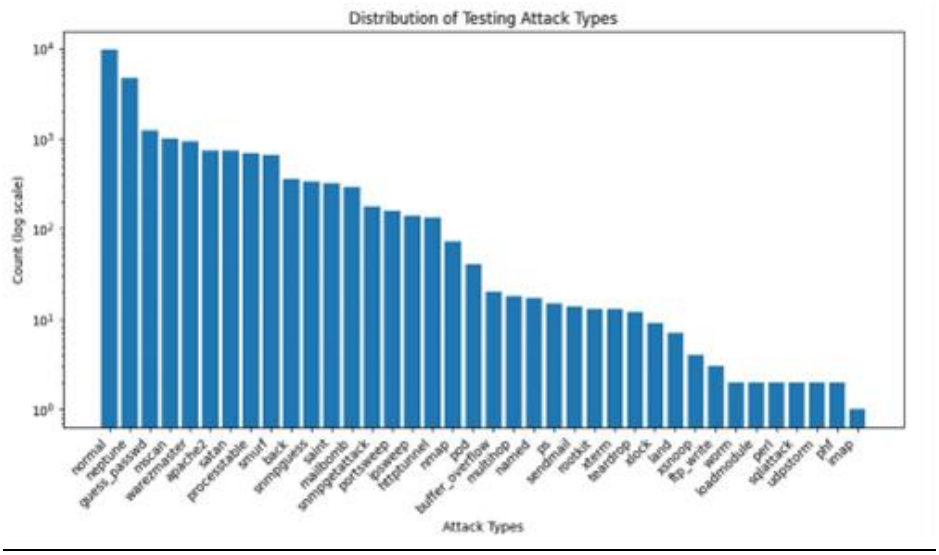

Figure 3. Visual Representation of Testing Dataset: Differentiating Normal and Attack Data.

3.2. Data Pre-processing

There NSL-KDD dataset comprises of three non-numeric datapoints: protocol type, flag and service. We transform the datasets into numerical and binary vectors with label encoding to convert ordinal data into numerical form and one-hot encoding into binary representation (Su et al., 2020). Data scaling is performed to standardize or normalize the features in both datasets to ensure they have similar scales, in order to converge faster and better results for the decision tree algorithm.

3.2.1 Feature Selection

We utilize a Random Forest Classifier to perform Recursive Feature Elimination with Cross-Validation (RFECV), to identify the most important features in the datasets like TCP, UDP and ICMP along with other relevant features for the decision tree predictive modelling. The parameter '**rfecv.fit(x_train_encoder, y_train)**' was called to perform the feature selection process. It fits the RFECV model to that training data, where '**x_train_encoder**' represents the input features and y_train represents the target labels (normal or attack data) (Mustaqim et al., 2021).

3.3. Decision Tree Algorithm

Decision tree algorithm is one of the supervised machine learning algorithms widely used in data mining techniques. The classifier possesses the capabilities of processing huge amount of data and is mostly used to categorise information based on training sets and class labels, to classify newly available data, and to make assumptions about categorical class names (Jijo and Abdulazeez, 2021). The algorithm approaches a problem by connecting nodes of the criteria to build a tree structure (Mustaqim et al., 2021). Decision trees use several methods in solving problems. Methods like C4.5, CART and ID3. In this work, the paper focuses on the C4.5 and CART to make comparisons and select the method that produces the best prediction for the simulated network. Figure 4 below shows the structure of DT.

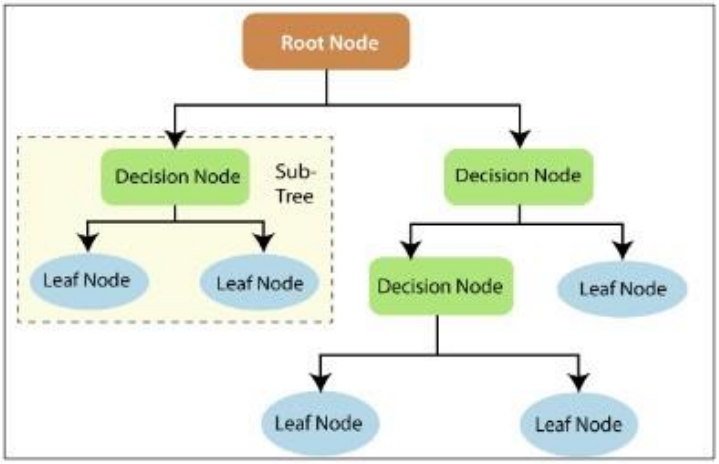

Figure 4. Structure of DT (Jijo and Abdulazeez, 2021)

In the CART algorithm, the 'Gini' criterion is the method used to elevate the quality of a split in the decision tree. It measures the probability of a randomly selected element being misclassified. The gini of a node is calculated as:

$$Gini(p) = 1 - \sum_{i=1}^{n} (p_i)^2 \quad (1)$$

In the C4.5 algorithm, the 'Entropy' is a criterion that calculates the information gain achieved by a particular split. Information gain measures how much the uncertainty in the target variable (class labels) is reduced after a split. The entropy of a node is calculated as:

$$Entropy(p) = -\sum_{i=1}^{n} p_i log_2(p_i) \quad (2)$$

3.4. Performance metrics

Our research evaluates the model using accuracy, precision, recall, f1score and confusion matrix as performance metrics. These metrics are mostly known to provide a comprehensive understanding of the performance of a classification model. The recall and precision calculations are based on the results of data testing with the ground truth value represented by True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (TN) (Canbek et al., 2021).

Precision determines the correctness of the positive predictions by assessing false positives. This is calculated as:

$$Precision = \frac{T_P}{T_P + F_P} \quad (3)$$

Recall defines the exactness of the positive cases in the actual data by assessing cases where the actual class is positive, but the model predicts negative. Less false negatives are influenced by a higher recall. It is calculated as:

$$Recall = \frac{T_P}{T_P + F_N} \quad (4)$$

Accuracy calculates the percentage of correct predictions among all the predictions. It is calculated as:

$$Accuracy = \frac{T_P + T_N}{T_P + F_P + T_N + F_N} \quad (5)$$

F1-Score illustrates the mean value of the precision and recall. It gives a balance by minimizing false positives and false negatives. It is calculated as:

$$F1 - score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \quad (6)$$

Confusion matrix visualises the classification model on a set of data for which the true values are known.

3.5 Network Traffic Simulation

To further illustrate the efficiency of the developed model, simulation of real-world traffic was passed as input to the model and the metrics were evaluated. The simulated traffic was generated using the Distributed Internet Traffic Generator (D-ITG). This tool is an open-source framework capable of producing traffic at packet level accurately reproducing traffic which is as stochastic in nature as real-world traffic. The tool supports both IPv4 and IPv6 traffic generation and can generate traffic at the network, transport, and application layers.

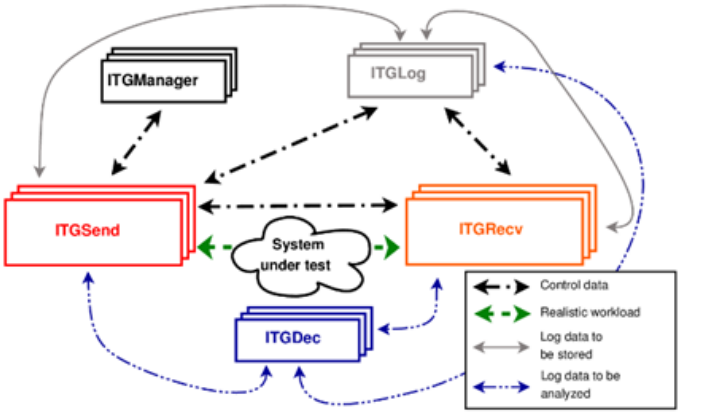

Figure 5. D-ITG Architecture

4. Results

In this section, we present the results obtained from our implemented protocol anomaly detection system using the decision tree algorithm within the simulated network environment. The outcome is presented and analysed to evaluate the potency and performance of our approach.

4.1. Decision Tree Evaluation

To find an ideal model for the simulated environment, the decision tree model was trained by varying the hyperparameters and evaluation criteria. This was trained by varying the max-depth and the number of splits. The evaluation criteria used were "entropy" and "gini".

The results obtained are highlighted in Tables 2 and 3. Table 2 shows the performance metrics obtained for varying the max-depth and number of splits when the criterion was set to gini. Likewise, table 3 shows the performance metrics obtained for varying the parameters when the criterion was set to entropy.

Table 2. Performance metrics for Gini.

Max-depth	Number of Splits	Accuracy	Precision	F1-score	Recall
50	100	98.60%	97.09%	98.52%	100%
50	50	99.53%	99.00%	99.50%	100%
50	**10**	**99.90%**	**99.79%**	**99.89%**	**100%**
50	200	97.95%	95.79%	97.85%	100%
50	500	97.07%	94.08%	96.94%	99.99%
10	100	96.33%	92.69%	96.20%	99.99%
10	50	96.64%	93.27%	96.51%	99.99%
10	**10**	**96.70%**	**93.38%**	**96.57%**	**99.99%**
10	200	96.03%	92.14%	95.91%	99.99%
10	500	95.30%	90.84%	95.19%	99.98%
20	100	98.60%	97.08%	98.52%	100%
20	50	99.43%	98.79%	99.39%	100%
20	**10**	**99.73%**	**99.43%**	**99.71%**	**100%**
20	200	97.95%	95.79%	97.85%	100%
20	500	97.06%	94.08%	96.94%	99.99%

Table 3. Performance metrics for entropy.

Max-depth	Number of Splits	Accuracy	Precision	F1-score	Recall
50	100	98.88%	97.66%	98.82%	100%
50	50	99.48%	98.90%	99.44%	100%
50	**10**	**99.98%**	**99.83%**	**99.91%**	**100%**
50	200	98.32%	96.53%	92.74%	100%
50	500	96.35%	92.74%	96.23%	100%
10	100	97.20%	94.34%	97.08%	100%
10	50	97.51%	94.93%	97.40%	100%
10	**10**	**97.65%**	**95.20%**	**97.08%**	**100%**
10	200	96.91%	93.77%	96.78%	100%
10	500	95.57%	91.32%	95.46%	100%
20	100	98.88%	97.66%	98.82%	100%
20	50	99.48%	98.90%	99.91%	100%
20	**10**	**99.98%**	**99.83%**	**99.91%**	**100%**
20	200	98.32%	96.53%	98.23%	100%
20	500	96.35%	92.74%	96.23%	100%

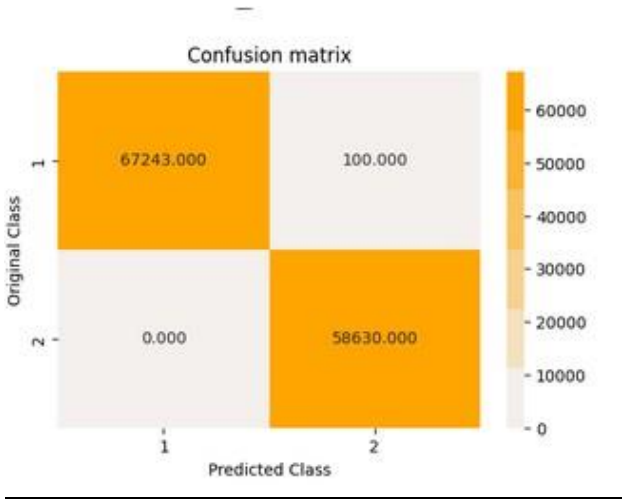

Figure 6. Confusion matrix for best model

Table 4. Performance Metrics of Integrated Model in Simulated Network

Metric	Results
Accuracy	80.0%
Precision	92.0%
Recall	80.0%
F1-Score	70.0%

5. Evaluation and Discussion

To find the best-predicting model to be used for the simulated environment, we conducted experiments with the CART and C4.5 decision tree methods. This section of our paper presents an extensive analysis and interpretation of the achieved results from our experiments while making comparisons with some existing works.

Tables 2 and 3 show the performance of the metrics for the DT model using entropy and gini methods respectively, while using varied combinations of maximum depth and number of splits. These parameters were varied because they affect the decision tree significantly. Maximum depth restricts the number of layers the decision tree can have while the number of splits determines the number of branches or nodes in the tree (Tzirakis and Tjortjis, 2017).

From the tables, consistently high levels of accuracy, precision, recall, and F1-scores in both sets of experiments can be observed. The highlighted rows in the tables show the maximum depth value and number of split combinations that gave the highest performance. The accuracy ranged from 95.30% to 99.90% when gini was used as the splitting criterion, and precision values ranged from 90.84% to 99.79% with an f1-score ranging from 95.19% to 99.89%. The entropy-based experiments yielded similar but slightly higher performance with accuracy ranging from 95.57% to 99.98%, while precision values were from 91.32% to 99.83%. F1-scores are also high, giving a good balance between precision and recall with values ranging from 92.74% to 99.91%. Additionally, the consistent performance of 100% of the model's recall indicates its ability to determine all actual positive instances. These results show that our model can produce accurate predictions and accurately capture positive instances.

After analyzing the performances from the selected criteria, entropy was adopted as the best method to integrate into the simulated network. Hence, the model for the

simulated network had a maximum depth of 20, 10 splits, and entropy as the criterion method. Figure 6 shows the confusion matrix of the model adopted for testing in the simulated network. The choice of the model was made to address overfitting and imbalanced values as higher values for the maximum depth and number of splits posed a risk of overfitting.

The deployment of the model into a simulated network gives an understanding of real-world applicability that addresses the challenge of early detection of anomalies on computer networks, as well as its efficiency in detecting protocol anomalies. After a successful deployment, we examined the model's performance using the metrics accuracy, precision, recall, and F1-scores as shown in Table 4. The performance of the integrated model in the simulated network environment showed satisfactory results. The model's accuracy of 80.0% suggests that it properly made minimal errors on just some of the data. Furthermore, the high precision score of 92.0% indicates that the model's positive predictions were correct, lowering the number of false positives and alarms. The model's recall score of 80.0% demonstrates its ability to detect a high percentage of genuine network anomalies. This trait is especially important in network security applications, where quick detection of even minor deviations is critical to preventing threats. The F1-score of 70.0% illustrates a well-balanced trade-off between precision and recall, suggesting the model's ability to maintain a strong equilibrium between precise detection and minimum false alarms. The integrated model's high performance underlines its effectiveness in detecting anomalies in a simulated network environment. The capacity to distinguish between typical network activity and dangerous threats highlights its potential value in bolstering network security measures. Because of the model's ability to learn and adapt to changing network behaviors, it serves as a proactive tool for detecting novel attack patterns or anomalies.

We made comparisons of our results after the model was integrated into our simulated network with previous research in the field of NIDS using machine learning techniques. Our accuracy values align with the performance published by Wighneswara et al. (2023) and Dsouza et al. (2022) who used comparable datasets and decision tree models. However, the precision and recall values of our model outperform those published by Al-Khassawneh (2023) which suggests the effectiveness of our chosen approach.

While the integrated model yields encouraging results, it is imperative to recognize that simulated environments might not perfectly represent the complexity of real-world network circumstances. The assumptions made through the simulation process could have an impact on how well the model performs. Further studies should evaluate the model's performance in conditions like live network environments or large-scale simulations as well as exploring with other deep learning techniques, as our approach utilized a machine learning technique.

6. Conclusions

Our investigations have shown that the development and implementation of a comprehensive network security solution using machine learning techniques hold the potential to significantly improve threat detection. Our study has leveraged the strengths of decision trees to detect protocol anomalies indicative of potential malicious activities. The simulated network, designed to mirror real-world scenarios, serves as a controlled testing ground for the development, training and fine-tuning of machine learning models. Although decision tree model is successful at predicting positive cases with good accuracy results of 80.0% after the model was integrated into the simulated network, the computational performance analysis demonstrated that the decision tree model has got a high execution time. Finally, the minimum number of samples split parameter successfully addressed the challenge of overfitting. The proposed approach can be explored further with other datasets and simulated environment types.

Data Availability Statement: The NSL-KDD dataset is publicly available at http://205.174.165.80/CICDataset/NSL-KDD/

All source codes used in the implementation of this research is available at https://github.com/KateAsante/UFCE4B-60-M-cyber-security-research-paper

Acknowledgments: This work represents the culmination of collective efforts, insights, and support, without which its completion would not have been possible. We wish to express our sincere appreciation to our esteemed supervisor, Dr. Abdullahi Arabo, for his unwavering guidance throughout this journey. Our sincere thanks also go to our peer, Baffour Sarkodie-Mensah for his thoughtful discussions, feedback, and assistance. We acknowledge the University of the West of England (UWE Bristol) as well as our module leaders for facilitating resources for conducting this research. Finally, we express our deepest appreciation to our families and friends for their immeasurable support and encouragement, which has been a constant source of motivation.

References

1. A. B. Nassif, M. A. Talib, Q. Nasir and F. M. Dakalbab (2021) Title: Machine Learning for Anomaly Detection: A Systematic Review Available: https://ieeexplore.ieee.org/abstract/document/9439459 [Assessed: 4TH August, 2023]
2. Adnan Mohsin Abdulazeez and Bahzad Taha Jijo (2021) Title: Classification Based on Decision Tree Algorithm for Machine Learning Volume 2 pp 21 Available: https://www.jastt.org/index.php/jasttpath/article/view/65/2 [Assessed: 24TH August, 2023]
3. Alifiannisa Alyahasna Wighneswara, Anita Sjahrunnisa, Yasinta Romadhona, Khoifah Inda Maula, Salsabila Mazya Permataning Tyas, Ary Mazharuddin Shiddiqi and Hudan Studiawan (2023) Title: Network Behavior Anomaly Detection using Decision Tree Available: https://ieeexplore.ieee.org/document/10134589/authors#authors [Assessed: 4TH August, 2023]
4. A. Dsouza, V. Lanjewar, A. Mahakal and S. Khachane (2022) Title: Real Time Network Intrusion Detection using Machine Learning Technique Available: https://ieeexplore.ieee.org/document/10014863 [Assessed: 20th April, 2023]
5. Alamin Talukder, Fida Hasan Khondokar, Manowarul Islam, Ashraf Uddin, Akhter, Arnisha, Abu Yousuf, Mohammad, Alharbi Fares and Mohammad Ali (2022) Title: A Dependable Hybrid Machine Learning Model for Network Intrusion Detection Available: https://ui.adsabs.harvard.edu/abs/2022arXiv221204546A/abstract [Assessed: 4TH August, 2023]
6. Amuthan Prabakar Muniyandi, R. Rajeswari, R. Rajaram (2011) Title: Network Anomaly Detection by Cascading K-Means Clustering and C4.5 Decision Tree algorithm Available: https://www.sciencedirect.com/science/article/pii/S1877705812008594 [Assessed: 4TH August, 2023]
7. Arash Heidari, Nima Jafari Navimipour and Mehmet Unal (2022) Title: Applications of ML/DL in the management of smart cities and societies based on new trends in information technologies: A systematic literature review Available: https://doi.org/10.1016/j.scs.2022.104089 [Assessed: 17TH August, 2023]
8. Azidine Guezzaz, Said Benkirane, Mourade Azrour and Shahzada Khurram (2021) Title: A Reliable Network Intrusion Detection Approach Using Decision Tree with Enhanced Data Quality Available: https://doi.org/10.1155/2021/1230593 [Assessed: 4TH August, 2023]
9. A. Z. Mustaqim, S. Adi, Y. Astuti and Y. Pristyanto (2021) Title: The Effect of Recursive Feature Elimination with Cross-Validation (RFECV) Feature Selection Algorithm toward Classifier Performance on Credit Card Fraud Detection Available: https://scholar.google.co.id/citations?view_op=view_citation&hl=id&user=qv_awmAAAAAJ&citation_for_view=qv_awmAAAAAJ:XiVPGOgt02cC [Assessed: 24TH August, 2023]
10. B. Ingre and A. Yadav (2015) Title: Performance analysis of NSL-KDD dataset using ANN Available: https://ieeexplore.ieee.org/document/7058223 [Assessed: 17TH August, 2023]
11. B. Özçam, H. H. Kilinc and A. H. Zaim (2021) Title: Detecting TCP Flood DDoS Attack by Anomaly Detection based on Machine Learning Algorithms Available: https://scholar.google.com/citations?view_op=view_citation&hl=en&user=oHPADc0AAAAJ&citation_for_view=oHPADc0AAAAJ:Se3iqnhoufwC [Assessed: 4TH August, 2023]
12. Dr. A. Malathi and S. Revathi (2013) Title: A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection Available: https://www.academia.edu/download/75596334/4.IJERT_DEC.pdf [Assessed: 23RD August, 2023]
13. Dipankar Dasgupta, Zahid Akhtar, and Sajib Sen (2020) Title: Machine learning in cybersecurity: a comprehensive survey Volume 19 Available: https://doi.org/10.1177/1548512920951275 [Assessed: 4TH August, 2023]

14. D. Vasumathi and R. Samrin and (2017) Title: Review on anomaly based network intrusion detection system Available: https://ieeexplore.ieee.org/document/8284655 [Assessed: 4TH August, 2023]
15. D. Vinod and M. Prasad (2023) Title: A novel hybrid automatic intrusion detection system using machine learning technique for anomalous detection based on traffic prediction Available: https://ieeexplore.ieee.org/document/10127442 [Assessed: 4TH August, 2023]
16. Gürol Canbek, Tugba Taskaya Temizel and Seref Sagiroglu (2021) Title: BenchMetrics: a systematic benchmarking method for binary classification performance metrics Available: https://link.springer.com/article/10.1007/s00521-021-06103-6 [Assessed: 25TH August, 2023]
17. Iman Sharafaldin, Arash Habibi Lashkari and Ali A. Ghorbani (2018) Title: Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization Available: https://www.scitepress.org/Papers/2018/66398/66398.pdf [Assessed: 17TH August, 2023]
18. Jiandong Chen, Ming Pu and Wenxuan Hou (2019) Title: The trend of the Gini coefficient of China (1978–2010) Volume 17 Available: https://doi.org/10.1080/14765284.2019.1663695 [Assessed: 24TH August, 2023]
19. Lean Yu, Kin Keung Lai , Rongtian Zhou and Rongda Chen (2020) Title: Missing Data Preprocessing in Credit Classification: One-Hot Encoding or Imputation? Available: https://doi.org/10.1080/1540496X.2020.1825935 [Assessed: 23RD August, 2023]
20. Lisa Sharma-Wallace , Sandra J. Velarde and Anita Wreford (2018) Title: Adaptive governance good practice: Show me the evidence! Available: https://doi.org/10.1016/j.jenvman.2018.05.067 [Assessed: 4TH August, 2023]
21. M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani (2009) Title: A detailed analysis of the KDD CUP 99 data set Available: https://ieeexplore.ieee.org/document/5356528 [Assessed: 17TH August, 2023]
22. Marta Catillo, Antonio Pecchia, Umberto Villano (2023) Title: CPS-GUARD: Intrusion detection for cyber-physical systems and IoT devices using outlier-aware deep autoencoders Available: https://doi.org/10.1016/j.cose.2023.103210 [Assessed: 17TH August, 2023]
23. Panagiotis Tzirakis and Christos Tjortjis (2017) Title: T3C: improving a decision tree classification algorithm's interval splits on continuous attributes Available: https://link.springer.com/article/10.1007/s11634-016-0246-x [Assessed: 25TH August, 2023]
24. T. Su, H. Sun, J. Zhu, S. Wang and Y. Li (2020) Title: BAT: Deep Learning Methods on Network Intrusion Detection Using NSL-KDD Dataset Available: https://ieeexplore.ieee.org/abstract/document/8988230 [Assessed: 24TH August, 2023]
25. T. Kim and W. Pak (2022) Title: Robust Network Intrusion Detection System Based on Machine-Learning With Early Classification Available: https://ieeexplore.ieee.org/abstract/document/9687553 [Assessed: 4TH August, 2023]
26. Tanzila Saba , Amjad Rehman , Tariq Sadad, Hoshang Kolivand and Saeed Ali Bahaj (2022) Title: Anomaly-based intrusion detection system for IoT networks through deep learning model Available: https://www.sciencedirect.com/science/article/pii/S0045790622001100 [Assessed: 20th April, 2023]
27. University of Maryland (2010) Title: Midterm solutions (corrected) Available: http://www.cs.umd.edu/~lily/Teaching/498FAssignments/Midterm-solutions-corrected.pdf [Assessed: 24TH August, 2023]
28. Xiaodan Xu, Huawen Liu, Minghai Yao (2019) Title: Recent Progress of Anomaly Detection Available: https://doi.org/10.1155/2019/2686378 [Assessed: 4TH August, 2023]
29. Y. A. Al-Khassawneh (2023) Title: An investigation of the Intrusion detection system for the NSL-KDD dataset using machine-learning algorithms Available: https://ieeexplore.ieee.org/document/10187360 [Assessed: 4TH August, 2023]
30. Z. Li, A. L. G. Rios and L. Trajković (2020) Title: Detecting Internet Worms, Ransomware, and Blackouts Using Recurrent Neural Networks Available: https://ieeexplore.ieee.org/abstract/document/9283472 [Assessed: 4TH August, 2023]

MEETING LOGS

Title of Project:	Protocol Anomaly Detection Using Decision Trees Machine Learning Algorithm in a Simulated Network
Student name:	Kate Biama Asante
Supervisor name:	Dr. Abdullahi Arabo

Date	Meeting Notes	Actions
7th March, 2023	• Take out real live implementation. • Improve upon signature-based methods (Find other trends of detection) • Review papers published in the last five years • Research about simulation of live systems.	• Research topic was modified to implement a simulated system instead of a live network. • Protocol anomaly detection trend was chosen over signature based method • Related works ranging from 2019 to 2023 was analysed. • Extensive research was done about simulated networks.
21st March, 2023	• Reviewed previous meeting notes and expected improvements • Make good comparisons of related works. • Key objectives should be at most five. • Find data resources to use in building machine learning model • Conduct more research to find more papers for literature review	• Literature review reflected extensive comparisons between papers. • Objectives were narrowed to five. • NSL-KDD dataset to be used for the training and testing of the machine learning model was chosen. • Literature review had almost 20 papers been reviewed.
30TH May, 2023	• Reviewed previous meeting notes and expected improvements. • Proceeding to begin research paper and implementation of proposed project • Research paper can make a final comparison between with new work and other 3 existing papers.	• Started writing research paper. • Machine learning model was built while testing with different parameters to enhance efficiency.

	• Experiments with different parameters to find a suitable predicting model for the simulated network.	
6TH July, 2023	• Reviewed previous meeting notes and expected improvements. • Simulation set up.	• Research into building a simulated network was done.
1ST August, 2023	• Reviewed previous meeting notes and expected improvements. • Discussed written literature review for research paper • Making extensive comparisons between papers for literature review while highlighting their weaknesses.	• Literature review was extensively discussed in research paper with highlights of weakness of the reviewed papers. • More research was done into the building of simulated network.
16TH August, 2023	• Reviewed previous meeting notes and expected improvements. • Create a virtual line to differentiate adopted methodologies in research paper of the architectural view. • Citing of authors should include surname only. • Introduction should mention research questions and objectives for the study.	• Architectural view of research methodology was modified to display the parts. • Citation and reference skills were improved upon. • Introduction for research paper was modified to reflect research questions and objectives. • Research paper was completed and submitted.