Derna Academy Journal for Applied Sciences

E-ISSN: 3006-3159

DAJAS

Analysis and Classification of Cyberattack Patterns Using a Decision Tree Algorithm: A Synthetic-Data-Based Study

Radwan M Marwan

Department of Information Technology ,Higher Institute of Science and Technology, Tajoura, Libya

تحليل وتصنيف أنماط الهجمات السيبرانية باستخدام خوارزمية شجرة القرار :دراسة قائمة على البيانات الاصطناعية

رضوان محمد مروان

قسم تقنية المعلومات ،المعهد العالي للعلوم والتقنية، تاجوراء، ليبيا

*Corresponding author: E-mail address: it.marwanmr@gmail.com

Volume : 8 **Issue: 1** **Page Number: 159 - 177**

Keywords: Artificial Intelligence, Decision Tree, Cybersecurity, Phishing, SQL Injection, DoS/DDoS, Password-Related Attacks

Copyright: © 2026 by the authors. Licensee: The Derna Academy for Applied Science (DAJAS). This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) License (https://creativecommons.org/licenses/by/4.0/)

Received : 01\10\2026
Accepted: 20\10\2025
Published: 21\10\2026

https://doi.org/10.71147/qctf7y75

ABSTRACT

Cyber-attacks present a serious threat to the security of web applications and online services. Artificial Intelligence (AI) offers effective approaches for analyzing attack patterns and classifying different types of cyber-attack. This paper explores the effectiveness of the Decision Tree algorithm in classification tasks involving phishing, SQL injection (SQLi), denial-of-service and distributed denial-of-service (DoS/DDoS), and password-related attack techniques. The experiments were performed in the Python programming language using artificially generated data sets. Accuracy, confusion matrices, class-wise F1-scores, and feature importance were evaluated to assess the model's performance, identify significant features, and analyze classification errors. The results showed accuracy rates of 85.0%, 95.5%, 99.0%, and 52.0% in phishing, SQLi, DoS/DDoS, and password-related attack classification tasks, respectively. These results highlight the model's inconsistency in the recognition task of the four selected attack categories and emphasize the complexity of distinguishing between password-related attack classes. The paper provides a comprehensive comparison of how the Decision Tree algorithm can utilize the input features to classify attack patterns in controlled experimental conditions. However, the use of artificially generated data sets with specific attack patterns may impact the accuracy and applicability of the results. The findings of this study contribute to the growing body of research on the application of AI algorithms in cybersecurity domains to detect and classify cyber-attacks. However, further research is needed to validate these results on larger-scale studies and real-world data sets. Comparing the performance of different algorithms can help identify the strengths and weaknesses of each approach to enhance the accuracy and efficiency of cyber-attack detection and classification. Additionally, the results of this research emphasize the importance of having high-quality data and a well-defined problem statement when developing effective solutions for detecting and classifying cyber-attacks.

الخلاصة

تُشكّل الهجمات السيبرانية تهديداً خطيراً لأمن تطبيقات الويب والخدمات عبر الإنترنت. ويوفر الذكاء الاصطناعي (AI) أساليب فعالة لتحليل أنماط الهجمات وتصنيف أنواع مختلفة منها. تبحث هذه الورقة في مدى فاعلية خوارزمية "شجرة القرار (Decision Tree) "في مهام تصنيف الهجمات، بما في ذلك التصيد الاحتيالي(Phishing) ، وحقن لغة الاستعلام الهيكلية(SQLi) ، وهجمات حجب الخدمة وحجب الخدمة الموزع(DoS/DDoS) ، والهجمات المتعلقة بكلمات المرور. أُجريت التجارب باستخدام لغة البرمجة "بايثون (Python) "بالاعتماد على مجموعات بيانات مُولَّدة اصطناعياً. ولتقييم أداء النموذج، وتحديد السمات المؤثرة، وتحليل أخطاء التصنيف، تم قياس دقة النموذج، واستخدام مصفوفات الارتباك Confusion) (Matrices، ومقاييس "F1-score" لكل فئة على حدة، بالإضافة إلى تقييم أهمية السمات .(Feature Importance) أظهرت النتائج معدلات دقة بلغت 85.0% للتصيد الاحتيالي، و95.5% لهجماتSQLi ، و99.0% لهجماتDoS/DDoS ، و52.0% للهجمات المتعلقة بكلمات المرور. تسلط هذه النتائج الضوء على تباين أداء النموذج في التعرف على فئات الهجمات الأربع المختارة، وتؤكد على مدى تعقيد التمييز بين فئات الهجمات المرتبطة بكلمات المرور. تقدم الورقة مقارنة شاملة حول كيفية استفادة خوارزمية "شجرة القرار" من سمات المدخلات لتصنيف أنماط الهجمات في ظل ظروف تجريبية مضبوطة. ومع ذلك، فإن استخدام مجموعات بيانات مُولَّدة اصطناعياً تحتوي على أنماط هجوم محددة قد يؤثر على دقة النتائج وإمكانية تعميمها. تساهم نتائج هذه الدراسة في تعزيز الأبحاث المتنامية حول توظيف خوارزميات الذكاء الاصطناعي في مجال الأمن السيبراني للكشف عن الهجمات وتصنيفها؛ غير أن هناك حاجة إلى إجراء المزيد من البحوث للتحقق من هذه النتائج من خلال دراسات أوسع نطاقاً وباستخدام مجموعات بيانات واقعية. إن مقارنة أداء الخوارزميات المختلفة يمكن أن تساعد في تحديد نقاط القوة والضعف في كل نهج، مما يسهم في تحسين دقة وكفاءة عمليات الكشف عن الهجمات السيبرانية وتصنيفها. علاوة على ذلك، تؤكد نتائج هذا البحث على أهمية توفر بيانات عالية الجودة وتحديد دقيق لطبيعة المشكلة عند تطوير حلول فعالة للكشف عن الهجمات السيبرانية وتصنيفها.

الكلمات المفتاحية: الذكاء الاصطناعي، شجرة القرار، الأمن السيبراني، التصيد الاحتيالي، حقنSQL ، DoS/DDoS، الهجمات المتعلقة بكلمة المرور

1.Introduction

The use of the Internet has become an essential part of our daily lives, integrated into numerous activities from e-commerce sites and online trading platforms to government banking systems and other institutions. However, all these entities are vulnerable to various types of cyberattacks. Users of social media and e-commerce platforms, for instance, are susceptible to phishing, social engineering, ransomware, and Denial-of-Service (DoS) attacks, among other topics that will be discussed in greater detail later in this paper. Cyberattacks, in their various forms, represent a global challenge affecting individuals, corporations, and government institutions alike, though the severity of the resulting damage varies. People utilize social media platforms for a range of activities, including interpersonal communication, online shopping, and purchasing products directly from specific pages.As attackers evolve their methods employing sophisticated tools and leveraging artificial intelligence (AI) to launch attacks users remain in a constant state of anxiety. AI capabilities, such as vulnerability detection, website spoofing, mass email distribution, and server hijacking, heighten these concerns. Meanwhile, cybersecurity professionals find themselves in a race against time to identify, combat, and mitigate these emerging threats by analyzing attack vectors and understanding the specific techniques employed in each instance (Foley, 2026). Cyberattacks of various kinds have become a source of concern for many internet users ranging from individuals on social media and e-commerce platforms to institutions such as corporations, banks, and government agencies due to the psychological, financial, and societal harm they inflict. These attacks vary in their methods, operational mechanisms, and the specific damage they cause. A significant factor increasing the danger of an attack particularly regarding its programmatic behavior is a lack of understanding of the underlying mechanism or method employed. Failure to proactively detect attacks or analyze the strategies and tactics used by attackers leads to various issues, most notably the inability to effectively mitigate the risks these attacks pose to websites. Artificial intelligence encompasses knowledge integrated into computer systems through logic; it operates based on specific algorithms designed to solve problems requiring advanced, non-routine intelligence (Radakovic & Marjanovic, 2023).

Decision Trees (DT): Decision trees are widely used to detect spam and cyber-intrusions due to their ability to identify rules and patterns within network traffic data and system activity. A decision tree generates a series of rules derived from available labeled data, organized into a tree-like structure. Various machine learning techniques, such as decision trees, have been employed to detect cyber-attacks. Because decision trees rely on training data drawn from past incidents and events, their capacity for improvement is enhanced (Blockeel et al., 2023) In general, decision trees are computationally inexpensive to build, fast at classifying unknown records, easy to interpret (especially when small in size), robust against noise (particularly when methods to prevent overfitting are applied), and capable of easily handling data redundancy. There are many types of attacks on websites through which websites, personal accounts or government agencies are hacked, and they differ from each other in the method of hacking and the mechanism of their work. We will mention in this paper a group of them. Phishing is a cyberattack in which an attacker tricks victims by sending

them deceptive links or creating fake websites that mimic real companies or government agencies.

These websites can ask users for personal information such as passwords, bank account numbers, or other private data. Generative artificial intelligence can also be utilized to create phishing messages that have fewer spelling and grammatical errors, thus tricking users into providing their credentials (Patil et al., 2026). A brute force attack uses trial and error to guess a password until the target is reached. Some attacks have variations for example, a dictionary attack uses a list of words as passwords, and the hybrid attack modifies the guessed passwords to generate new ones. In this case, the password spraying method uses a small number of passwords to guess the target's password. The credential stuffing attack uses personal information from previous data breaches to guess the target's password. The rainbow table is a method used to obtain a hash of the target's password while the keylogging attack captures a victim's keystroke and is an entirely different attack technique (Patil et al., 2026). A Denial-of-Service (DoS) attack is designed to incapacitate a website or server by overwhelming it with traffic or crashing it in another way. This task is complicated when distributed Denial-of-Service (DdoS) attacks are launched, in which the traffic comes from many various systems. The most common types of DdoS attacks are SYN flood, ICMP flood, HTTP GET flood, and others that target different areas of the web server. Attacks of this type can be detected and mitigated with the help of artificial intelligence, but the accuracy of AI depends on the available data and attack types (Chakraborty et al., 2023). SQL injection (SQLi) refers to an application that uses user input in an SQL query in an insecure manner, which may lead to the compromise of a database or other unintended effects. The main types are tautology-based, UNION-based, comment-based, and error-based. The presence of SQL characters such as quotation marks or semicolons does not necessarily mean an attack is occurring. The use of parameterized queries, input validation, and proper permissions in the database can prevent SQL injection (Pamarthi, 2020). Cybersecurity and artificial intelligence are among the most critical fields for internet users, particularly those vulnerable to attacks on their websites or personal accounts. Researchers have extensively studied various attack types, such as phishing, where attackers employ dangerous tactics including sending deceptive URLs to victims to harvest sensitive data. Artificial intelligence algorithms have been utilized to identify hidden patterns within website links, with studies showing that the Random Forest algorithm yields effective results in detecting phishing attacks (Sinchana & Thejaswini, 2025). SQL injection is considered one of the most dangerous attacks due to the direct access it grants to databases; by altering queries, attackers can gain unauthorized access to data. While website security providers have employed Intrusion Detection Systems (IDS) tools similar to firewalls, these have proven largely ineffective against zero-day vulnerabilities. However, other methods developed by researchers have successfully detected SQL injection attacks. These techniques involve creating regular expression objects based on SQL queries and specific SQL clauses, utilizing these patterns repeatedly (Pamarthi, 2020). Python's regular expression capabilities are combined with lexical analysis to parse the input, thereby distinguishing between malicious injection attempts and legitimate textual data. Studies have utilized and compared various artificial intelligence algorithms employing websites hosting malicious code and specific programming languages to detect attacks, yielding promising results (Kalgude, 2022; Njie & Gabriouet, 2024; Ismaeel et al., 2025). Machine learning techniques are also employed to detect DDoS attacks; these methods analyze data traffic to identify malicious programs that generate excessive data transmissionsuch as "SYN flood" attacks. Research indicates the use of Random Forest, Decision Trees, and Artificial Neural Networks (ANN) for this purpose (Ahmadi, 2024). While the availability of multiple detection methods for this type of attack is advantageous, analyzing the attack itself facilitates earlier detection and enhances prevention capabilities. Some studies have highlighted the challenges associated with detecting ransomware, particularly regarding the analysis of evolving attack vectors that conceal encryption processes and employ countermeasures to protect the decryption mechanism. Research indicates that randomized algorithms have demonstrated high efficiency in detecting ransomware attacks, a capability attributed to the power of artificial intelligence algorithms. In contrast, other studies have utilized artificial intelligence differently to analyze this type of attack, employing techniques such as Gaussian Discriminant Analysis (GDA), reinforcement learning (based on environmental experimentation and feedback), or deep learning via Artificial Neural Networks (ANN). These methods enable the detection of patterns and anomalies hidden within networks, images, documents, and system calls (Al-Odat, 2025; Ferdous et al., 2024). However, it is noted that while these studies employed such algorithms for detection, they did not utilize them to analyze the attack methodology itself. Certain programs, such as UNVEIL, provide a virtual environment for analyzing and detecting malware; UNVEIL monitors the system for suspicious activity, although it sometimes struggles to detect encrypted malicious programs (Al-Odat, 2025). Researchers have categorized XSS attack detection methods into three types: static analysis, which reviews source code; dynamic analysis, which analyzes the script during execution; and machine learning-based approaches, which build classifiers to analyze the behavior of new scripts including the ability to analyze malicious JavaScript code (Mereani & Howe, 2018; Njie & Gabriouet, 2024).

Other studies have utilized artificial intelligence specifically neural networks to distinguish between benign and malicious code by employing adversarial perturbation techniques; this involves introducing noise into benign code samples without compromising their functionality. This allows the code to bypass security measures, thereby aiding in the detection of attacks that exploit security evasion mechanisms (Kaushal, 2024). Additionally, a study utilizing Convolutional Neural Networks (CNNs) specifically the MRBN-CNN model for XSS attack detection demonstrated promising results, particularly in the model's ability to extract features from complex data patterns (Njie & Gabriouet, 2024). Artificial intelligence addresses many challenges associated with zero-day attacks; researchers have noted that deep neural networks can comprehensively interpret data behavior within a system unlike traditional systems that analyze files in isolation by learning patterns such as system call cycles and packet sequences, thereby enabling the analysis of previously undetected vulnerabilities (Kang, 2025). While various methods exist for detecting XSS attacks utilizing different algorithms existing approaches often fail to address deep analysis or the specific mechanics of the attack itself. Furthermore, studies indicate that Convolutional Neural Network (CNN) and Deep Neural Network (DNN) models are among the most effective for detecting brute-force attacks, particularly SSH password-guessing attacks (Hamza & Al-Janabi, 2024; Njie & Gabriouet, 2024).

Research Gap

Previously published articles used artificial intelligence and machine learning algorithms to identify and classify network events associated with cyber-attacks. Each study focused on specific attack categories, set of features, data sets, and assessment criteria. For example, Zunaid et al. (2026) identified phishing attempts using URL-based features, George and Zeratsion (2025) examined brute-force cyber-attacks targeting SSH services, while Sreedharan and Sampath (2026) explored the use of machine learning in detecting SQL injections. These articles provide valuable insights into the research area but do not allow concluding which classification method is better at distinguishing between several attack categories.

The research gap addressed by this study includes the need to evaluate attack-related features and classification errors in the context of different attack categories using a consistent methodology. This article examined the utilization of Decision Tree algorithm in the identification of four types of attacks: phishing attempts, SQL injection, DoS/DDoS activities, and password-related events. The current study analyzed variable importance, classification errors reflected in confusion matrices, and F1-scores for different classes to evaluate how well the selected algorithm could perform in a multi-classification task and identify the input features contributing to the recognition process.

The main contribution of this article includes the comparison of model's characteristics for several attack categories. Unlike previous articles, which focused on the specifics of one type of attack, the current research integrated multiple attack categories in the analysis. However, the findings should be considered as reflecting the behavior of the algorithm on the provided data sets, which are mostly simulated examples. More research is needed to confirm the model's applicability in practice, which includes the utilization of alternative data sets and the assessment of the approach's generalizability.

Comparative analysis of previous studies

Table 1. Comparative analysis of selected studies related to AI-based cyberattack detection

Scope and remaining research needs	Attacks studied	Data type	Algorithm approach	Study
Focuses on phishing classification. The presented description does not establish whether the system generalizes other attack categories or unseen attack pattern	Phishing websites	URL-based lexical and structural features; Dataset size and origin need further verification	Logistic Regression, Decision Tree, Random Forest	Zunaid et al. (2026) Phishing Website Detection using URL-Based Machine Learning for Real-Time Browser Security
Focuses on SSH authentication behavior rather than multiple web-application attack mechanisms. Generalization to other attack categories remains outside its scope**.**	SSH brute-force attacks	SSH login logs, including real and synthetic data	Random Forest, DBSCAN, compared with Fail2Ban and PAM Tally2	George and Zeratsion (2025) Detection of SSH Brute-Force Attacks Using Machine Learning
Primarily distinguishes normal queries from malicious ones; The study identifies the need to investigate and classify more SQL injection types. Its tests also show sensitivity to the composition of training data	SQL injection	SQLite3 dataset from Kaggle, containing normal and SQL injection queries	Decision Tree, Random Forest, SVM, Logistic Regression, LSTM, and clustering	Sreedharan (2026) Prediction of SQL Injection using Machine Learning
Proposes a broader detection-and-prevention framework, but the reviewed description does not provide sufficient detail to establish how well individual attack mechanisms are explained or how the model performs across all attack types	D/DOSphishing, malware injection, and unauthorized access	Network traffic and benchmark datasets such as NSL-KDD and CICIDS are discussed	Random Forest, SVM, ANN, and a proposed hybrid approach incorporating Isolation Forest	Kumari et al. (2026) AI Based Cyber Threats and Vulnerability Detection, Prevention and Prediction System

2. Methodology

The paper employs a descriptive analytical methodology to identify the most frequent attacks, including phishing, SQL injection, DoS/DDoS, and password-related cyberattacks. Experiments were performed in Python using the Decision Tree classifier. The datasets for experiments were created synthetically, with 80% of instances used for training and 20% for testing. Accuracy, precision, recall, F1-score, and confusion matrices were calculated to evaluate the results. Moreover, the feature importance and tree structure were assessed to analyze how the model classifies instances. As data was synthetically generated, the results only show how the model behaves with given input data, not proving its efficiency in practice.

Design and Results

Phishing Attack Analysis Using Decision Tree

A Decision Tree classifier was trained on the features of phishing URLs, including the presence of login forms, subdomains, IP addresses, and suspicious TLDs, as well as URL length, domain age, and special characters such as @. The model was trained on a synthetic dataset comprising 1000 instances that were split into training and validation sets in an 8:2 ratio. The classifier was constructed using the entropy criterion, a maximum depth of five, and class weighting. The performance of the classifier was evaluated using accuracy, precision, recall, and macro F1-score, with a confusion matrix and class-wise F1 scores. The feature importance and the tree structure were also analyzed for interpretability. Since the dataset was synthetic, the results were obtained for a proof-of-concept scenario and should be validated using a real-world phishing database.

DoS/DDoS Attack Analysis Using Decision Tree

A dataset containing 1,000 instances of traffic was generated with a selected packet rate, average packet size, and urgency score. Three previously defined rules were then applied to encode specific attack types with a set of features, including SYN Flood, ICMP Flood, HTTP GET Flood, and normal packets. The data sample was then split into training (80%) and testing (20%) subsets. A classification model was trained on the resulting data using the entropy criterion with a Decision Tree algorithm. The evaluation of the model was conducted using accuracy, precision, recall, and F1-score metrics, with macro averaging for precision and recall, and a confusion matrix for visualization. Additionally, the model's performance was analyzed using class-wise F1-scores and feature importance. As the dataset was engineer-generated, the results should be viewed as a proof-of-concept experiment rather than an empirical analysis of network traffic.

SQL Injection Attack Analysis

The experiment was carried out in Python with the help of the Scikit-learn library and the Decision Tree classifier. Its purpose was to differentiate between benign SQL queries and four types of SQL Injection: Tautology, UNION-Based, Comment-Based, and Error-Based attacks. For that, a data set consisting of 1000 instances was created. Each instance consisted of five features: the length of the query containing characters, and binary features of UNION, OR, comment symbols (--), and error symbols. The data set was split into training (80%) and testing (20%) subsets. The performance of the classifier was evaluated with the help of Accuracy, Precision, Recall, and F1-score metrics, which were calculated with the help of a confusion matrix. Moreover, the importance of each feature for the classification was estimated. The results of the experiment can be used to estimate the ability of a Decision Tree classifier to differentiate between SQL Injection attacks of various types on the data set used, but they cannot be extrapolated to real data.

Brute-Force Attack Analysis

The experiment was carried out in Python with Sklearn and the Decision Tree classifier to classify the five types of password-attack methods, including Online Dictionary, Hybrid Attack, Offline Rainbow Table, Credential Stuffing, and Password Spraying. In particular, the four features were used, including time complexity, resource cost, probability of detection, and evasion success rate.

The data set of 1500 instances were created for the experiment where 80% was used for training, and 20% was used for testing. The decision tree was constructed with entropy, maximum depth of 5, and minimum samples of 5 per leaf. Accuracy, macro averaged precision, recall, and F1-score were calculated for the model evaluation with a confusion matrix and feature importance analysis. Additionally, the illustrative risk scores were provided for each attack type based on predefined inputs and analyzed separately from the machine learning model.

Considering that the data for the experiment was generated with specific feature inputs, the results should be viewed as the classification accuracy for the specified data set. Moreover, it cannot be stated that the model is tested enough to provide reliable detection of attacks in the practical environment

3. Ethical Approval

This study did not involve human participants, animals, or personally identifiable information. All experiments were carried out on artificially generated datasets (1,000 instances each for phishing, SQL injection, and DoS/DDoS, and 1,500 instances for password-related attacks), created in Python with predefined feature rules. No real user data, network traffic, authentication logs, or live systems were collected, accessed, or tested, and no actual attacks were launched. Because the work is a computational study based entirely on synthetic data, formal review by an institutional ethics committee was not required. The research was conducted in line with accepted principles of research integrity and responsible cybersecurity research.

5. Results

Results and Analysis: Learning about SQL Injection Attacks

SQL Injection (SQLi) is a web application attack type that utilizes malicious input to modify SQL queries in such a way that they interfere with the intended database operation. Within this study, four attack patterns were analyzed: Tautology-Based, UNION-Based, Comment-Based, and Error-Based ones. Particularly, tautology attacks involve the use of logical conditions (e.g., OR 1=1), while UNION-based ones utilize UNION operator to combine two or more queries. Additionally, comment-based attacks use specific markers to hide parts of the query from interpretation, and error-based ones take advantage of database errors to retrieve information. In such a manner, the identified attack patterns are rather different, which is why the current Decision Tree model was implemented to classify them.

Decision Tree Structure and Attack Pattern Feature Importance

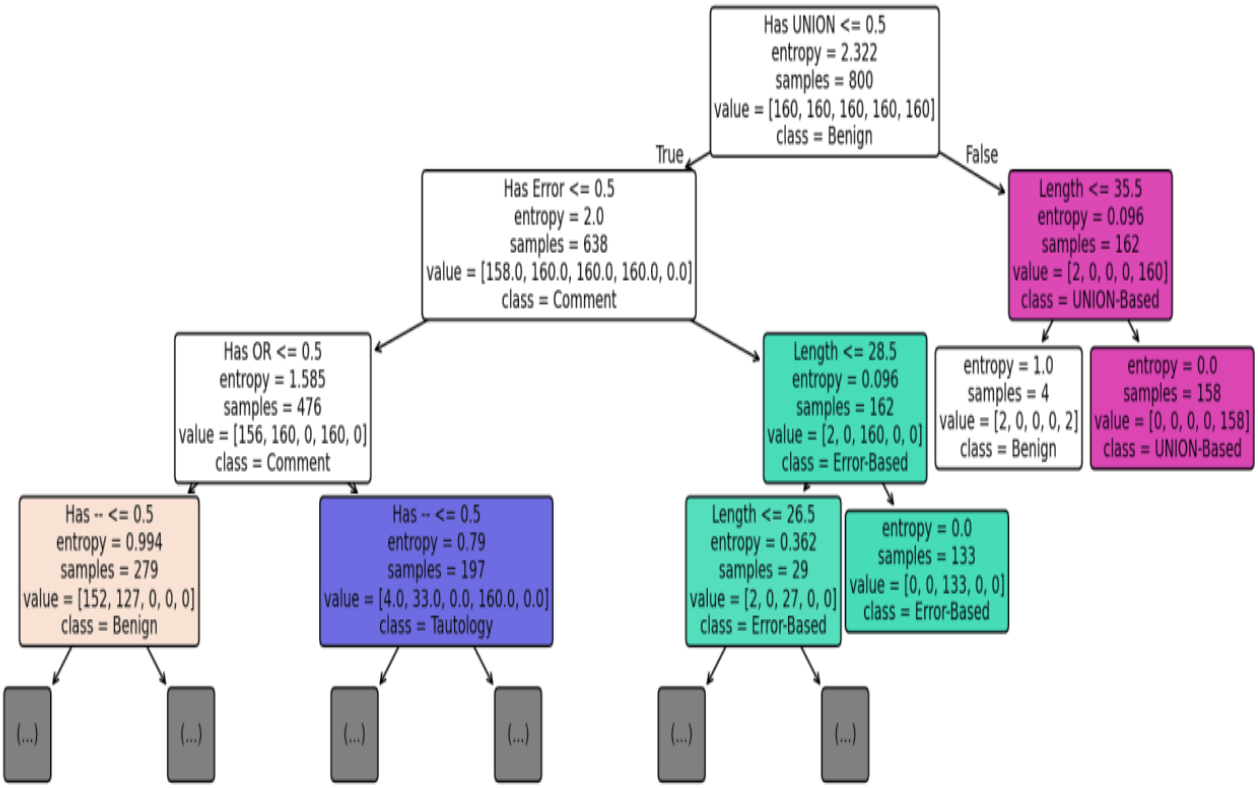


Figure 1: Decision Tree Classification Logic for SQL Injection

Figure 1 shows Decision Tree was constructed using five features, including query length, presence of UNION, OR, comment markers(--), and error signs. As seen from the plot above, the Decision Tree structure was rather simple and straightforward, where all four attack types and benign requests could be easily separated. However, it is important to note that individual features cannot be used for discriminations of benign and malicious requests since some of them are similar for different classes.

Feature Importance Analysis for DoS/DdoS

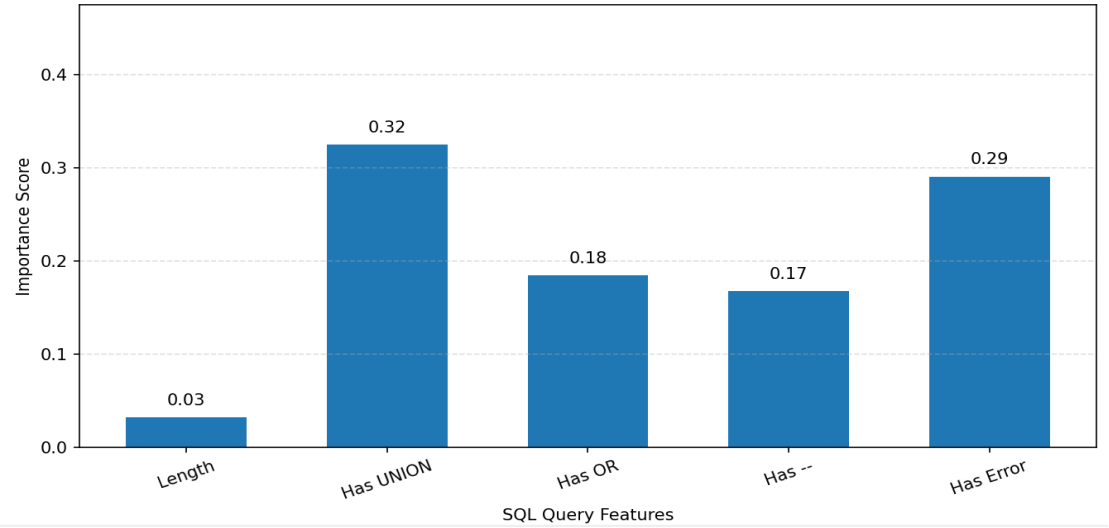


Figure 2: Feature Importance Analysis for SQL Injection

The results suggest that Has UNION has the highest importance with a score of 0.32, followed by Has Error with a score of 0.29, Has OR with a score of 0.18, and Has -- with a score of 0.17, while Length has the lowest score of 0.03. Therefore, it can be concluded that the SQL keywords and error indication contribute more to classification than length. However, the results were obtained using the synthetic dataset and might not generalize to other datasets.

Classification Performance and Error Analysis



Figure 3: Confusion Matrix for SQL Injection Classification (Predicted Class)

Figure 3 shows Analysis of Confusion Matrix in SQL Injection The given confusion matrix indicates that the Decision Tree algorithm has successfully classified four types of malicious SQL queries. In particular, there are 40 Error-Based, 40 Tautology, 39 Benign, and 39 UNION-Based instances. However, there are also 33 Comment class instances, of which only 33 were classified correctly, while 7 instances were mistaken for Tautology attacks. Thus, overall, the Decision Tree algorithm has demonstrated a performance of around 95.5% on the test set. The two classes that were the most challenging for the algorithm were Comment and Tautology. Nevertheless, these results were obtained on artificially generated data, which means that it is necessary to evaluate the algorithm's performance on real-life data sets of SQL injections.

Class-Wise F1-Score for SQL Injection

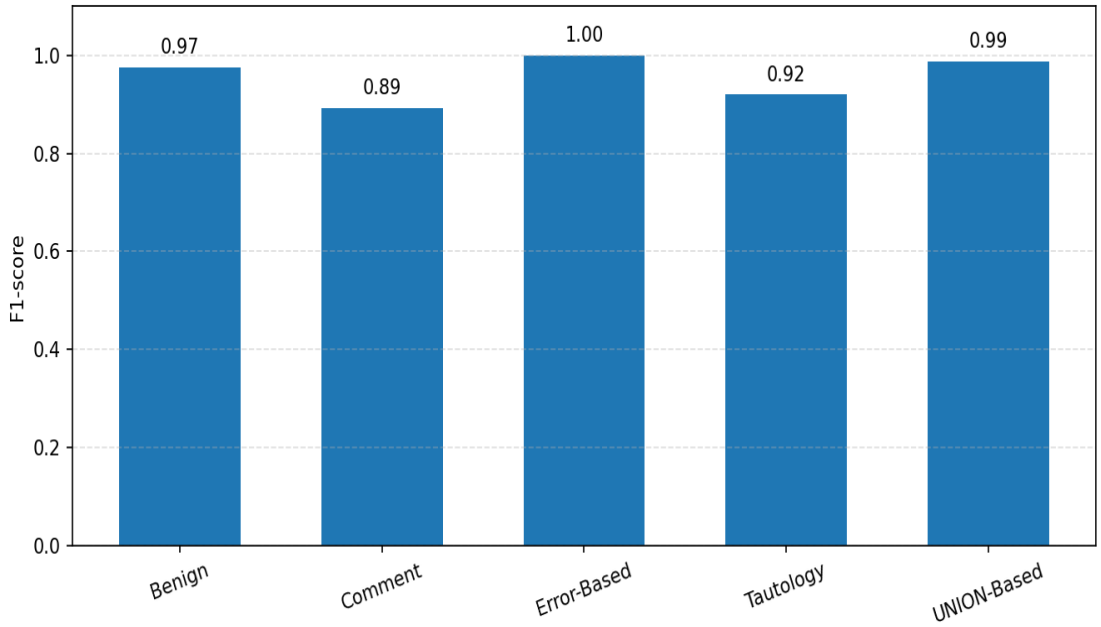


Figure 4: Class-Wise F1-Score for SQL Injection Detection (SQL Injection Class)

In Figure 4, it is possible to see the F1-scores for each class. The highest score was achieved by Error-Based (1.00), followed by UNION-Based (0.99), Benign (0.97), Tautology (0.92) and Comment (0.89). It can be seen that the model was able to classify Error-Based and UNION-Based queries best, while the Comment category was the worst. The F1-scores visualization helps understand the results better, which the confusion matrix cannot fully display. However, it is still necessary to validate the results on a larger scale with real-world data sets.

Overall Interpretation and Limitations

In conclusion, the current analysis demonstrated that different patterns of SQL injection attacks could be easily separated using a simple Decision Tree classifier. Indeed, taking advantage of multiple features (query length, UNION, OR, --, error signs) allowed effective discrimination between UNION-based, Comment-based, Tautology-based Error-based attacks and benign requests. However, it should be noted that the current study utilized a synthetic dataset with predefined attack patterns, while real-world SQL attacks might have a different structure. In such a manner, further studies should be conducted to evaluate the effectiveness of the current approach on a larger scale or utilizing benchmark datasets.

Results and Analysis: Understanding Phishing Case Study

Phishing is a security attack when hackers use social engineering to trick users into giving away their credentials. This study analyzed the URL attributes related to phishing: login form, many subdomains, IP address, suspicious top-level domain (TLD), long URL, domain age, and symbols. The results below demonstrate how these attributes can represent the URL's characteristics.

Decision Tree Structure and Features Importance

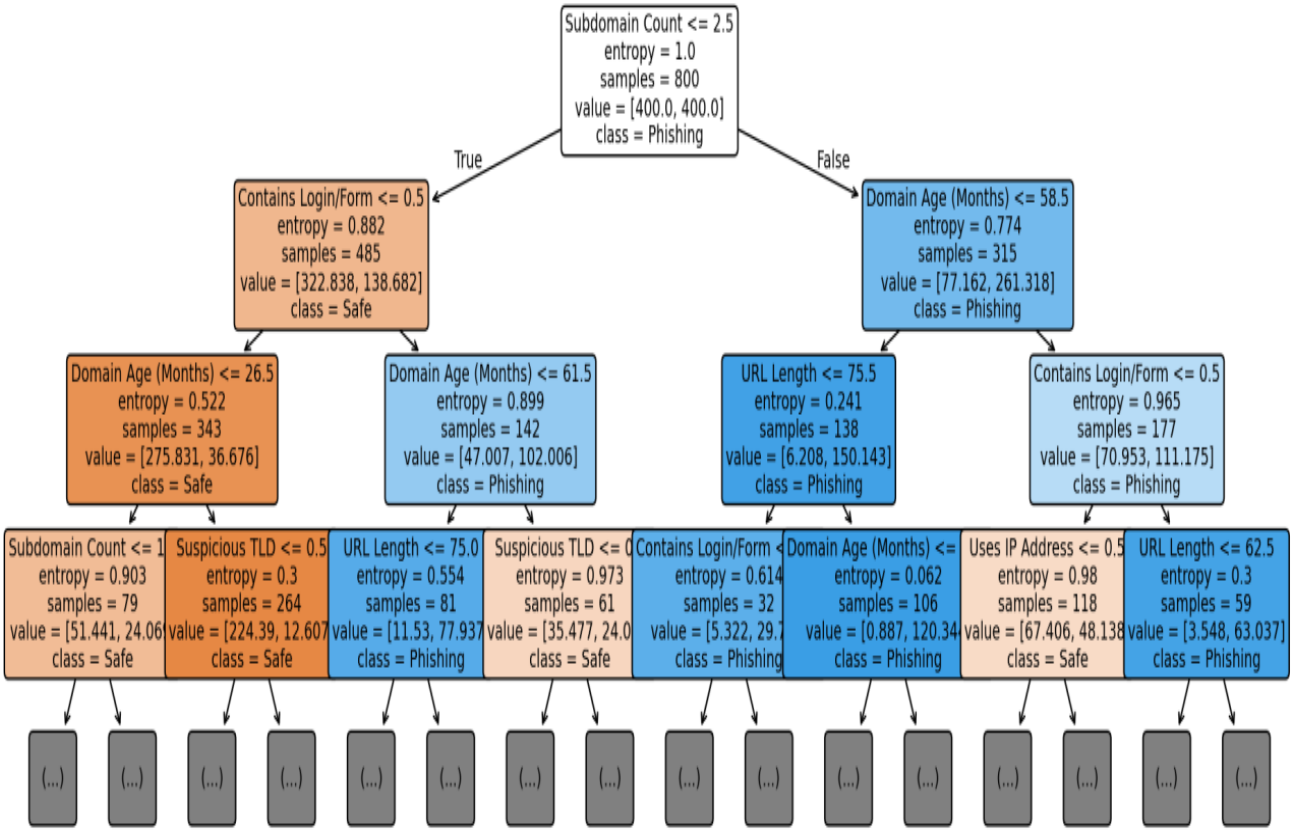


Figure 5: Decision Tree Classification Logic for Phishing

Figure 5 shows Decision Tree was applied to classify URLs into two categories: Safe and Phishing. The task required seven different attributes, and the tree below demonstrates how these attributes are combined to classify the URL as phishing or safe. Particular leaves of the tree were not sufficient for classification, and thus, it is possible to see how the classification algorithm combines different features in particular cases.

Feature Importance Analysis for Phishing Detection

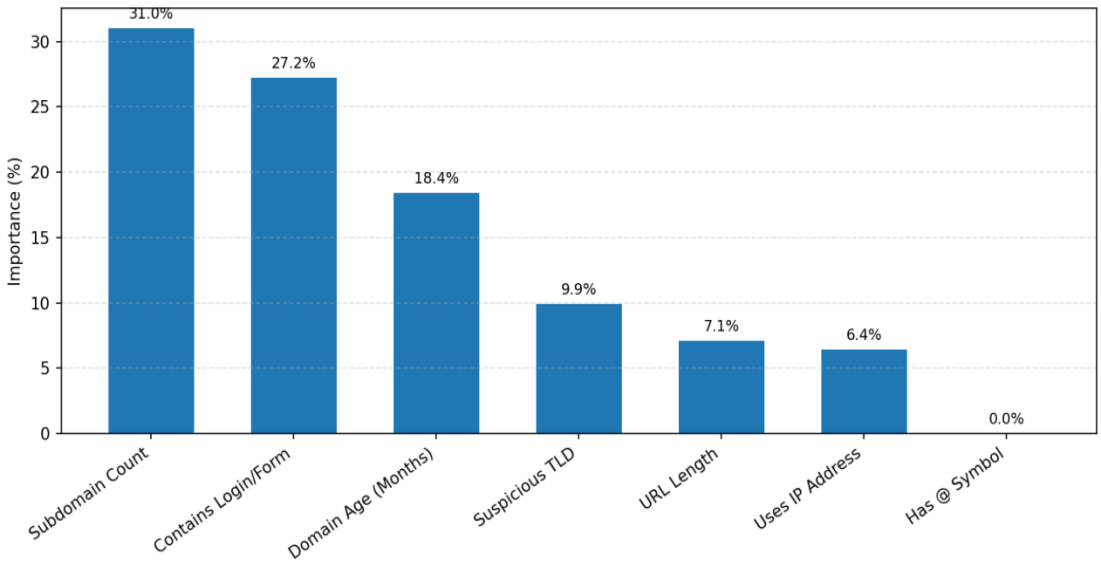


Figure 6: Feature Importance Analysis for Phishing Detection (URL Features)

Figure 6 shows that Subdomain Count possesses the highest feature importance, comprising 31% of all features, followed by the Contains Login/Form indicator at 27.2%. In turn, Domain Age accounts for 18.4% of all features, while Suspicious TLD comprises 9.9%. URL Length and Uses IP Address contributed to 7.1% and 6.4%, respectively. Finally, Has @ Symbol has no impact at all, scoring 0.0% in the present experiment.

Overall, it should be noted that Subdomain Count and Contains Login/Form account for the largest share of feature importance, comprising 58.2% in total. This finding implies that the complexity of a domain name and the presence of login data forms have the most significant impact on the model's predictions. Nevertheless, it is important to note that these results were obtained with the given data set, which does not necessarily guarantee that the use of these features will yield similarly high results for other sets, including in practice.

Classification Report and Confusion Matrix

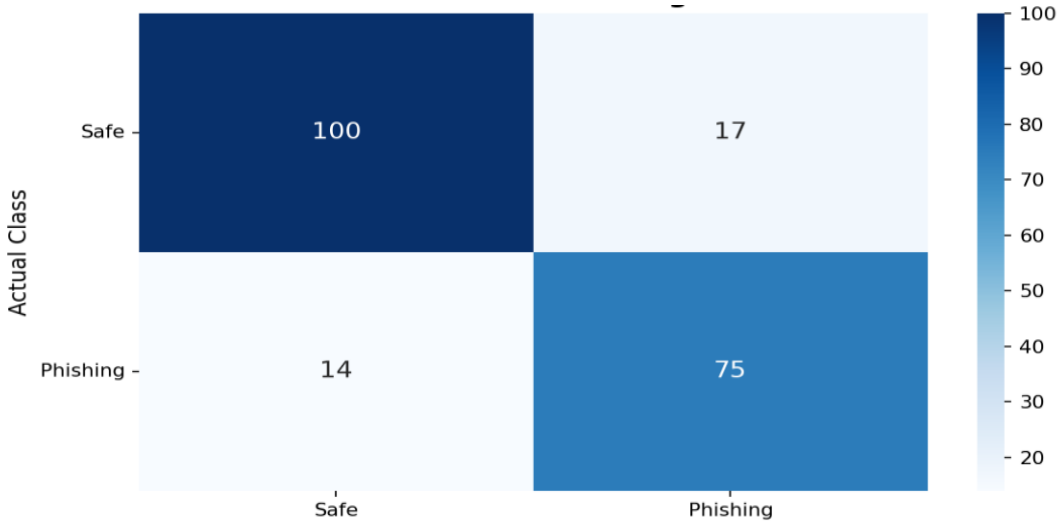


Figure 7: Confusion Matrix for Phishing Classification (Predicted Class)

Figure 7 shows The given confusion matrix indicates that 17 safe URLs were mistakenly identified as phishing, and 14 phishing sites were mistakenly identified as safe. One can see that errors of the second type are more serious from the security perspective, as they may involve a successful attack on the user's device. Therefore, it is critical to ensure that the number of false negatives is minimized in the classification model. In total, 175 of 206 instances (85.0%) were predicted correctly, which demonstrates that the model is somewhat effective at distinguishing between phishing and legitimate websites but requires additional tweaks to enhance its performance. Moreover, there is a need to test the algorithm on actual data to evaluate its efficacy.

Class-Wise F1-Score for Phishing

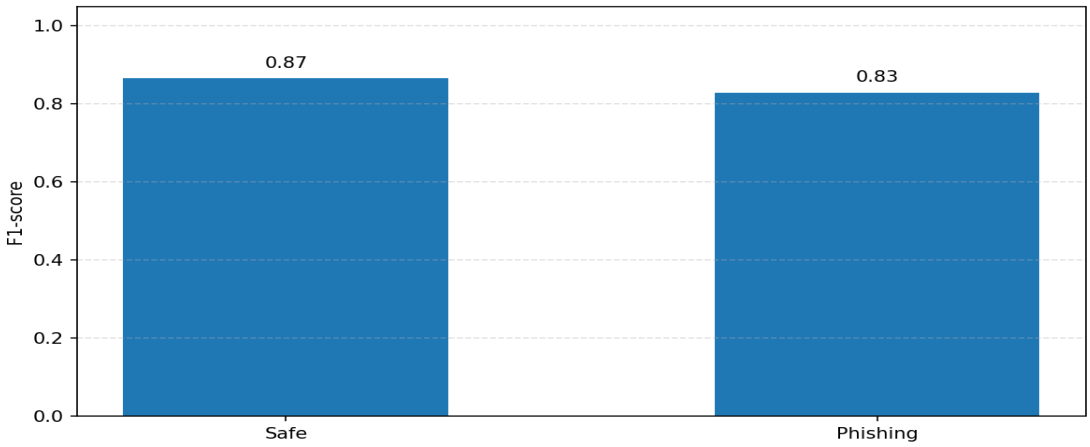


Figure 8: Class-Wise F1-Score for Phishing Classification (Phishing Class)

Figure 8 shows The model achieved an F1-score of 0.87 for Safe websites and 0.83 for Phishing websites. The lower score for phishing indicates the fact that some phishing sites were incorrectly identified as safe. Compared to the SQL injection model, which had higher F1-scores, the phishing model demonstrated a lower level of classification accuracy in this experiment. This finding may be explained by the difference in the features and patterns incorporated in each synthetic set. It is necessary to perform further analysis employing real sets to evaluate whether similar results would be obtained for actual websites.

General Discussion and Future Work

The results of this study demonstrate how Decision Tree classifies URLs in cases of phishing security attacks. The results below prove that the attributes provided enough information for the classification task, and the Decision Tree algorithm performed accordingly. Still, this study's limitations prevent making definitive conclusions about the classifier's performance. The experiment utilized one synthetic dataset with predefined labeling rules and thus did not reflect the actual performance of the classifier on the real dataset of phishing URLs and legitimate ones. Future work should consider conducting similar experiments with actual verified phishing URLs provided by companies as well as the URLs of real web applications.

Results and Analysis: DoS/DDoS Attack Coursework

In this study, Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks, which use networks to overload websites and services with requests or traffic, are investigated. Specifically, three attack patterns – SYN Flood, ICMP Flood, and HTTP GET Flood – were analyzed, where different aspects of network traffic were used to simulate an attack scenario. The investigation involves analyzing how these peculiarities of traffic patterns can be used to classify them.

Decision Tree Structure

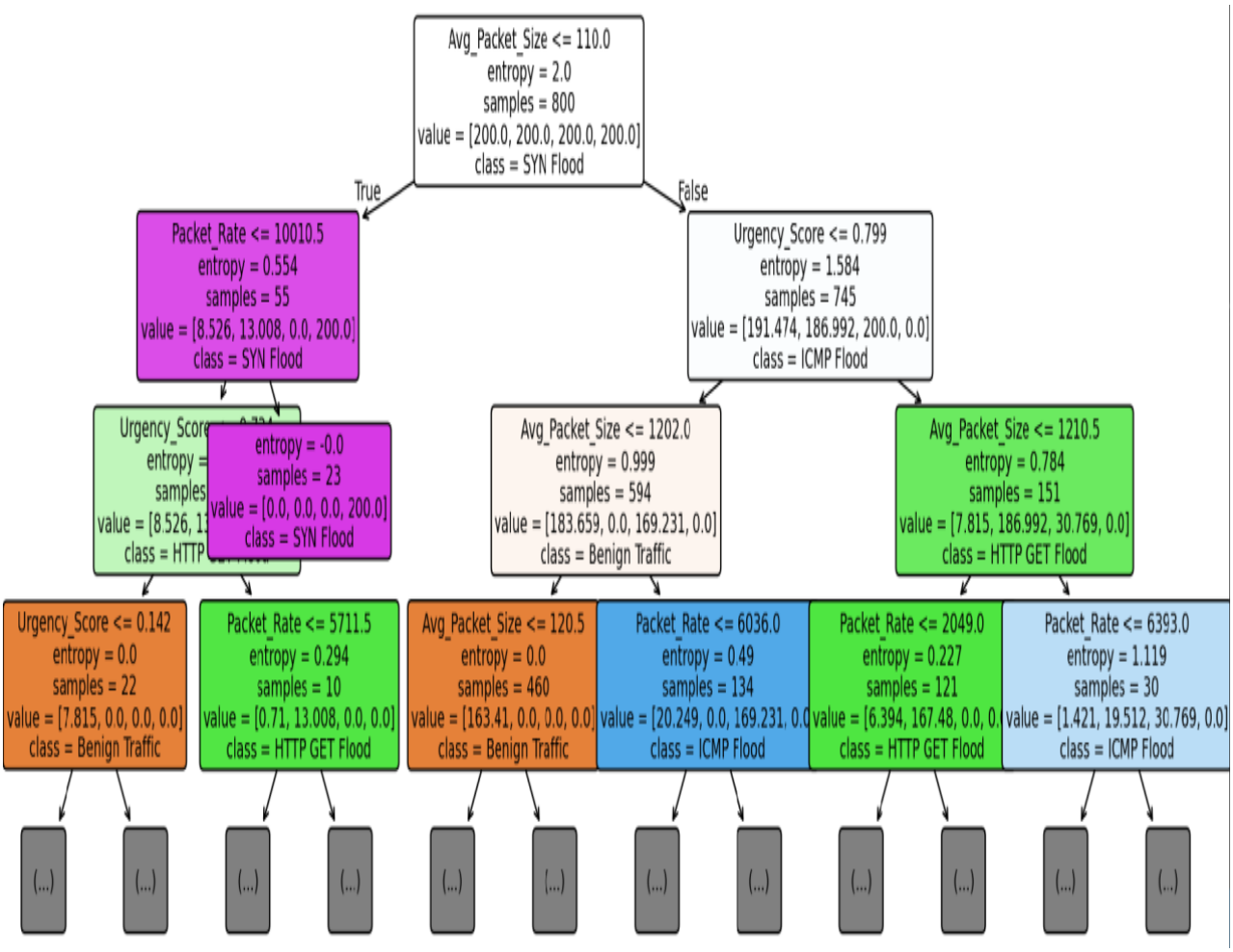


Figure 9: Decision Tree Structure for DoS/DDoS Traffic Classification

Figure 9 shows the decision tree was built for the classification task of distinguishing between Benign Traffic, SYN Flood, ICMP Flood, and HTTP GET Flood categories. It is possible to see that three features – packet rate, average packet size, and urgency – were used in the classification process. The tree structure implies that these features can be used to identify traffic pattern categories on the basis of specific criteria. Moreover, the tree accounts for rules defining the simulated patterns used in the study.

Feature Importance Analysis for DoS/DdoS

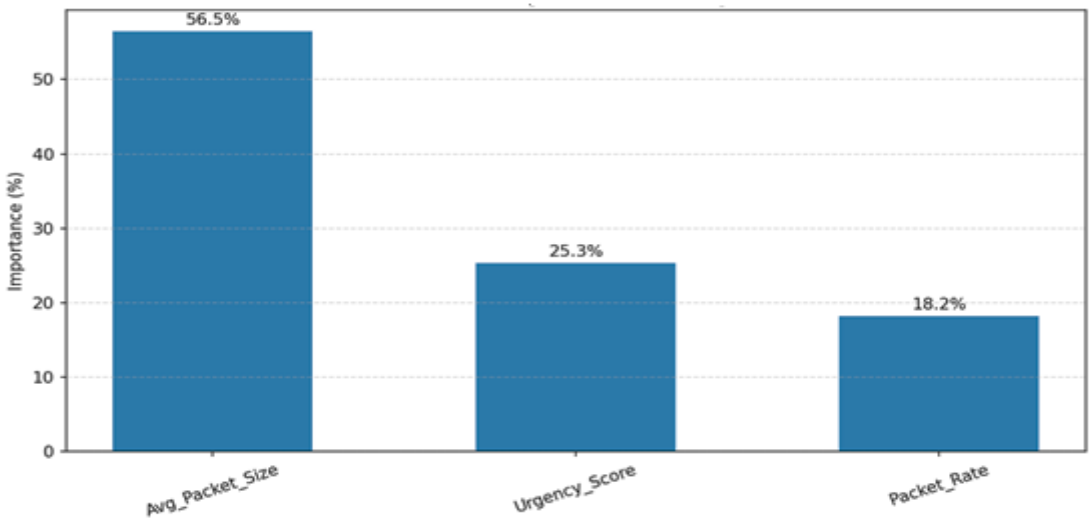


Figure 10: Feature Importance Analysis for DoS/DdoS Traffic(Traffic Features)

Figure 10 shows The outcome of the experiment suggests that the most common packet size the most contributed to the model's performance, with the value of 56.5%. The second most contributing field was Urgency_Score, with 25.3%, and the third was Packet_Rate, with 18.2%. Thus, the first two fields contributed to the model's performance by 81.8%, which indicates that they were the most critical.

In general, the model relied more on packet size and urgency score than on packet rate in this experiment. Moreover, it should be noted that these results were determined based on the data provided. Therefore, the conclusion that the same fields contribute to the model's performance in the same way cannot be fully transferred to practice without additional research. In addition, testing the model on actual traffic data would be beneficial for this task.

Confusion Matrix for DoS/DDoS

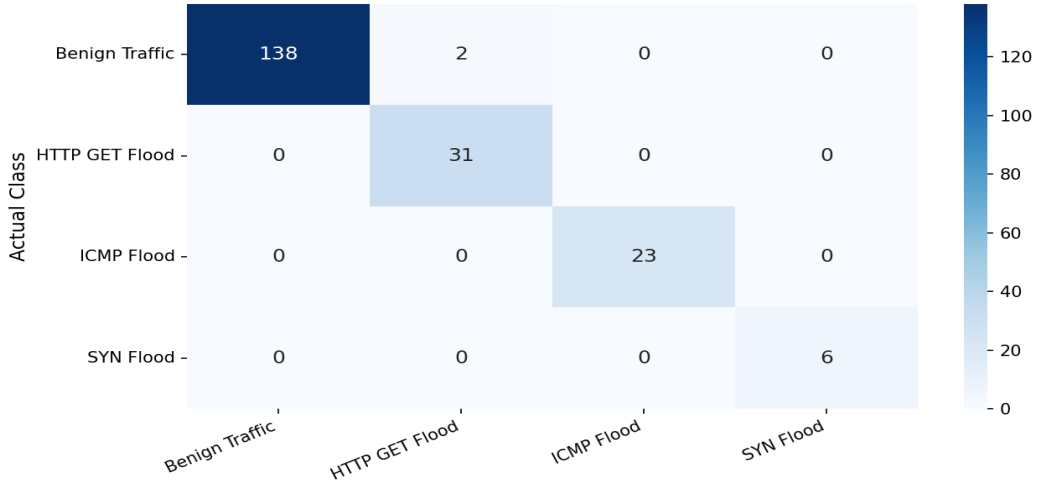


Figure 11: Confusion Matrix for DoS/DDoS Classification (Predicted Class)

Figure 11 shows The confusion matrix shows that the Decision Tree model correctly classified all HTTP GET Flood (31 cases), ICMP Flood (23 cases), and SYN Flood (6 cases). It also correctly classified 138 benign traffic cases, while two benign cases were incorrectly classified as HTTP GET Flood attacks.
Overall, the model achieved 99.0% accuracy, correctly classifying 198 out of 200 samples. Although the model performed very well on this dataset, these results are based on the experimental data used in the study and do not establish equivalent performance on real network traffic. Further validation using real-world datasets is needed to assess its reliability and generalization.

Class-wise F1-Score

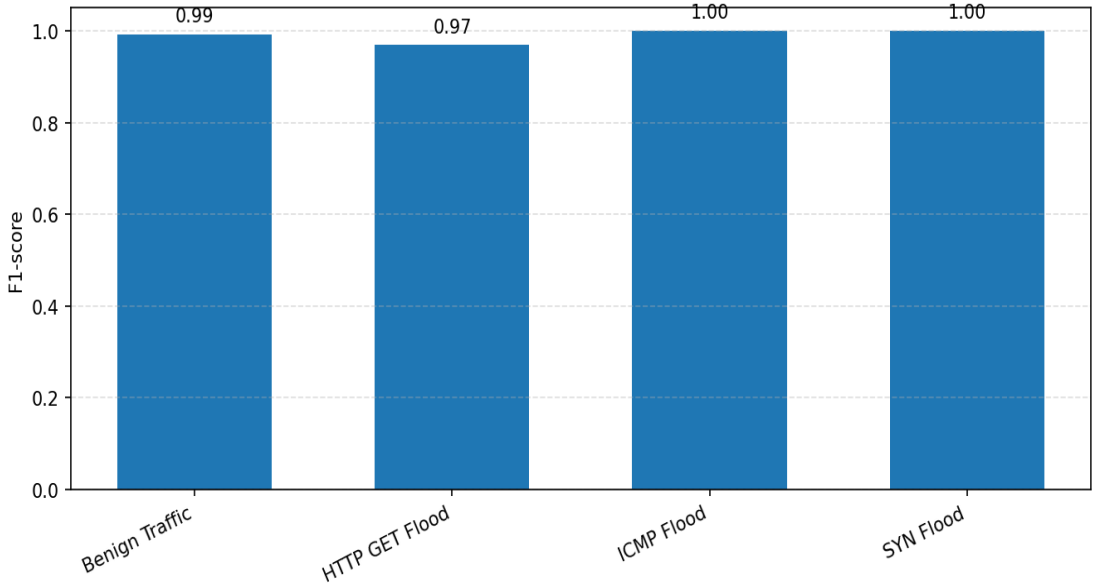


Figure 12: Class-Wise F1-Score for DoS/DdoS Classification (Traffic class)

Figure 12 shows The model demonstrated an F1-score of 1.00 for ICMP Flood and SYN Flood, 0.99 for Benign Traffic, and 0.97 for HTTP GET Flood. This implies that there was a high accuracy in classifying these traffic categories with only two benign traffic instances incorrectly classified as HTTP GET Flood attack traffic.
The model reached 99% accuracy on the simulated DoS/DDoS dataset; however, this accuracy rate was achieved on a simulated set of data with predefined rules for attacks. As such, while the model can identify simulated attacks with a very high degree of accuracy, there is no guarantee that it will perform equally well on real-world traffic data. Further testing and validation on actual network data is required to confirm its accuracy and reliability in real-world applications.
Compared to the SQL injection and phishing detection models, the DoS/DDoS model demonstrated the best performance in these experiments. Yet, since the models were trained on different datasets containing distinct traffic categories, it cannot be concluded that the DoS/DDoS model would outperform others in actual networking environments. Additional experiments must be conducted using realistic traffic before any definitive conclusions can be drawn.

Overall Interpretation and Conclusion

The way in which a decision tree works can be better understood in the context of the study investigating how DoS/DDoS attack patterns can be classified using simulated network traffic data. Specifically,
the decision tree structure, feature importance, confusion matrix, and F1-scores identify strengths and weaknesses in classification. However, these findings are limited to the identification of traffic patterns in a simulated environment, and further investigation is needed to evaluate how such a classification model can be applied in practice. In particular, future research can focus on applying the classification model to real-world traffic patterns and determining how well it can identify attacks in practice. As a result, the choice of features and the model itself might be optimized to account for the differences between real-world traffic patterns and attack patterns simulated in this study.

Password-Related Attack Classification Using Decision Tree

A Brute-Force attack is an attempt to gain unauthorized access to an account by guessing the target password. This attack differs from the SQL injection-type since brute-force does not rely on the database's specific vulnerabilities. Instead, the attacker tries different password combinations in an attempt to guess it. Therefore, it is possible to identify such attacks by analyzing the request's frequency, unsuccessful login attempts, and other similar elements. The general idea behind such attacks' analysis is to differentiate brute force from regular login attempts.

Decision Tree Structure and Feature Importance

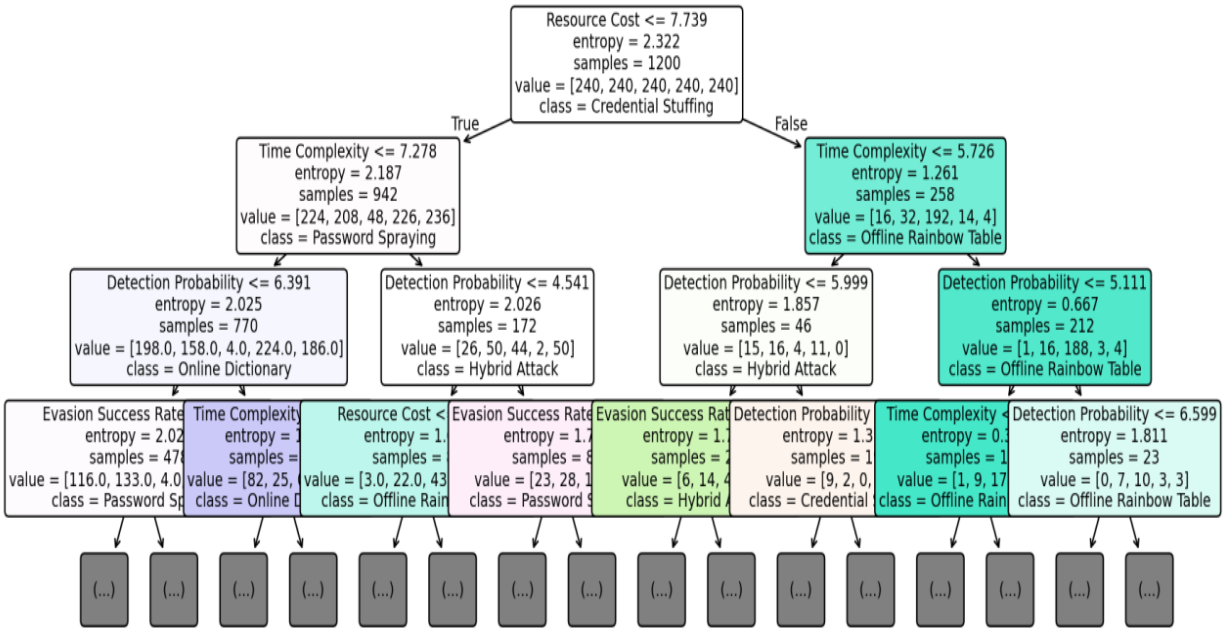


Figure 13: Decision Tree Structure for Password-Attack Classification

Figure 13 shows The Decision Tree's structure is based on the features that were provided to the algorithm for analysis. As seen in the visualization above, the tree classes brute-force attacks in accordance with the specific features that are described below. Therefore, it is possible to see how the brute-force identification model works.

Feature Importance in Password-Attack

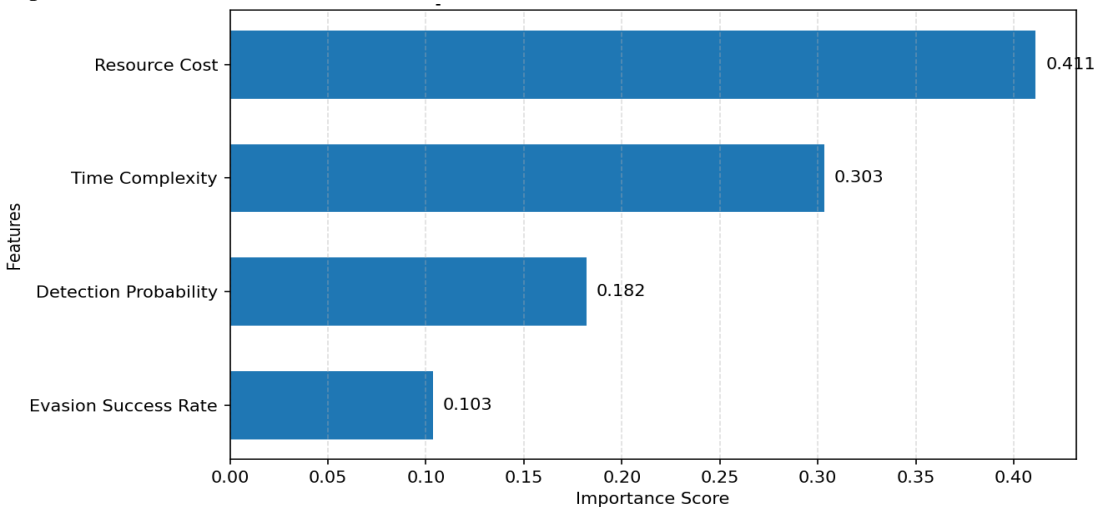


Figure 14: Feature Importance in Password-Attack Classification

Figure 14 shows the results suggest that Resource Cost has the highest value among the other three factors, with a feature importance of 0.411. This was followed by Time Complexity (0.303), Detection Probability (0.182), and Evasion Success Rate (0.103). Resource Cost and Time Complexity sum up to about 71.4% of the total importance, which implies that they have the most significant impact on the model.
Indeed, the model gives more importance to resource costs and time complexity than to detect probability and evasion success rate. These results only represent the trends of the data that were used to train the model. It does not mean that detection probability and evasion success rate have no impact on the real-life cyber-attack detection process. It is necessary to test the model further and use real-world data to see if the results will have the same trends.

Classification Performance and Error Analysis

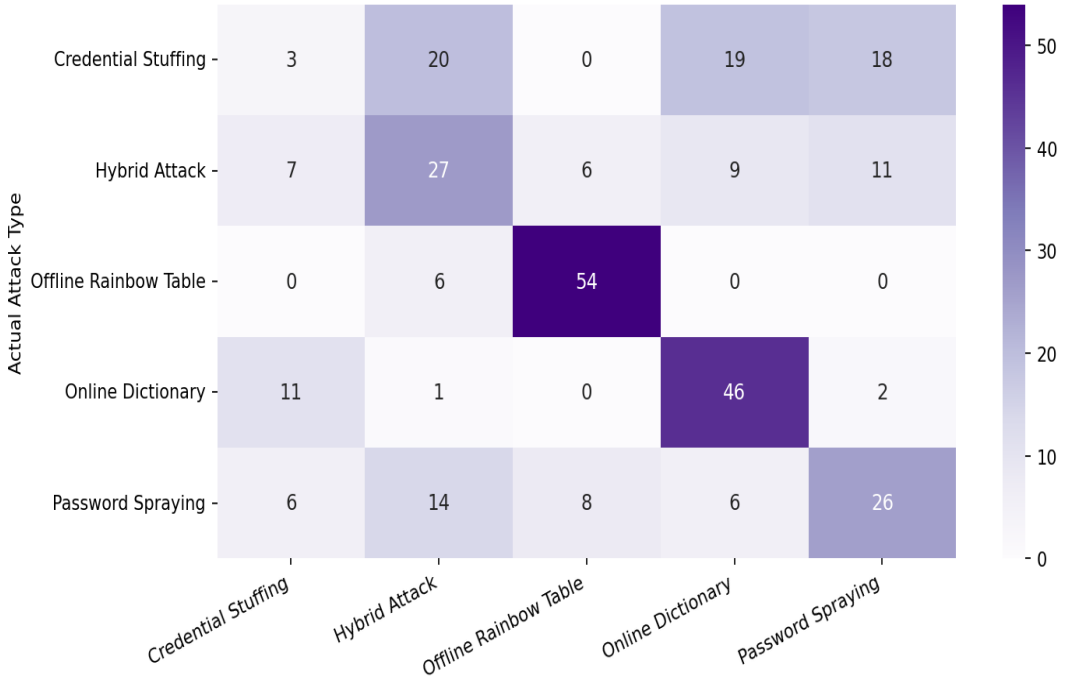


Figure 15: Confusion Matrix for Password-Attack Classification (Predicted Attack Type)

Figure 15 shows the confusion matrix below demonstrates that the Decision Tree was moderately successful in classifying the different types of password attacks. Offline Rainbow Table was guessed most frequently (54 out of 60). Then, Online Dictionary was at second place (46), followed by Hybrid Attack (27), Password Spraying (26), and Credential Stuffing (3). Overall, the model guessed 156 samples correctly (52.0% of 300). The main challenge was to differentiate between Credential Stuffing, Hybrid Attack, and Password Spraying. The reason for the poor identification may be related to the fact that the features used for training were not sufficient to make a difference between the five classes of attacks. Therefore, additional experiments must be conducted to evaluate other attributes and use realistic data sets to increase the efficiency of the model.

Class-Wise F1-Score for Password-Attack

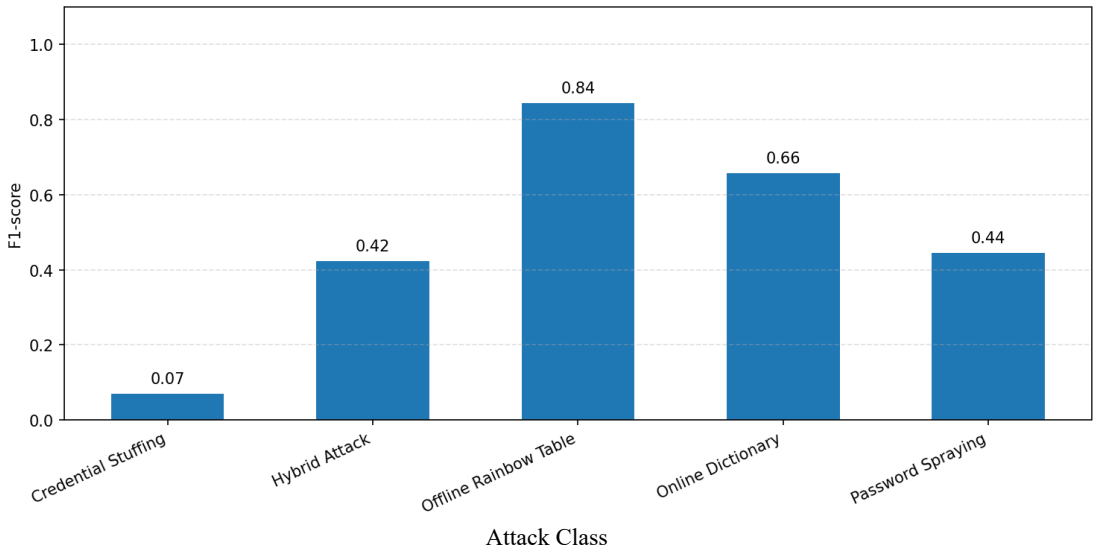


Figure 16: Class-Wise F1-Score for Password-Attack

Figure 16 The model achieved the highest F1-score for Offline Rainbow Table (0.84), followed by Online Dictionary (0.66), Password Spraying (0.44), Hybrid Attack (0.42), and Credential Stuffing (0.07). This outcome demonstrates the model's inability to differentiate between several password-related attack types, especially Credential Stuffing.
Overall, the results suggest that the four selected features were insufficient to distinguish all five classes reliably in this experiment. Therefore, the model requires further improvement through additional discriminative features and evaluation using realistic datasets before its practical reliability can be established.

Overall Interpretation

The current section sought to provide an interpretation of the results that were obtained using the Decision Tree algorithm. First, the analysis relied on the brute-force attacks classification in accordance with the features that were used in the experiments. Following that, feature importance demonstrated that certain characteristics of attacks are more common than others. Moreover, the confusion matrix and classification scores were used to highlight how the model classifies different types of login attempts.
Overall, the current analysis is limited by the data provided and the selection of features that were analysed. If the model was built on the fabricated dataset, it is necessary to question the results' accuracy. In other words, if the Decision Tree only relies on the presence of certain features that signify a brute-force, then its real-life application might be flawed. In future research, it would be interesting to test the model's performance on various authentication logs and analyze how it differentiates between benign and malicious login attempts

5. Discussion

This study evaluated the use of the Decision Tree algorithm for the classification of four categories of attack: SQL injection, phishing, DoS/DDoS, and password-related attack techniques. The models were compared in terms of accuracy, precision, recall, F1-scores, confusion matrices, and feature importance.
For the classification of SQL injection, the model achieved an accuracy of 95.5%. The class-wise F1-scores were equal to 1.00 for Error-Based, 0.99 for UNION-Based, 0.97 for Benign, 0.92 for Tautology-Based, and 0.89 for Comment-Based queries. The model misclassified seven Comment-Based samples as Tautology-Based. The most informative features were the appearance of UNION indicators and database error signs.
The phishing model achieved an accuracy of 85.0%, with an F1-score of 0.87 for Safe and 0.83 for Phishing. According to the confusion matrix, 17 safe URLs were incorrectly identified as phishing, and 14 phishing URLs were incorrectly identified as safe. Thus, the model is likely to have high false-negative rates in the case of unverified URLs. The number of subdomains and the presence of a login form were the most informative features for phishing detection. The findings suggest the necessity of additional experimentation with verified URLs.

The DoS/DDoS model achieved an accuracy of 99.0%, with an F1-score of 1.0 for ICMP Flood, 1.0 for SYN Flood, 0.99 for Benign Traffic, and 0.97 for HTTP GET Flood. The model misclassified two benign traffic samples as HTTP GET Flood. The most informative feature was the average packet size, followed by the urgency score and packet rate. However, the high accuracy is explained by the fact that the model was trained on synthetic data with clearly defined patterns. Thus, the results should not be directly generalized to the case of real traffic.

Finally, the attack classification model achieved an accuracy of only 52.0%. The class-wise F1-scores were equal to 0.84 for Offline Rainbow Table, 0.66 for Online Dictionary, 0.44 for Password Spraying, 0.42 for Hybrid Attack, and 0.07 for Credential Stuffing. The model had significant difficulties in distinguishing between different types of attacks. According to the confusion matrix, Credential Stuffing was frequently misclassified as other attack types. However, according to the inspection of feature importance, the differentiation between these classes was not aided by the consideration of resource cost and time complexity features.

In general, the experiments demonstrate that Decision Trees allow for obtaining human-interpretable classification rules, which is a critical advantage of the algorithm. At the same time, the analysis of confusion matrices and class-wise F1-scores indicates the possibility of errors that accuracy might fail to reflect. In particular, precision and recall might be useful metrics to report for each class to report false-positive and false-negative rates. Furthermore, the experiments did not directly compare the performance of different classification models; Therefore, the choice of Decision Trees was justified by convenience rather than performance. Moreover, the accuracy scores were obtained using different synthetic datasets with various features and attack classes, which do not allow for direct comparisons between experiments. The experiments could be extended by testing the models on real-world or benchmarks and comparing the performance of different classification algorithms.

6. Conclusion

The paper has explored the use of the Decision Tree algorithm for analyzing and classifying four types of cyberattacks: SQL Injection, phishing, DoS/DDoS, and brute-force attacks. The study highlights how the characteristics peculiar to each attack type can be utilized in machine learning to classify samples and identify attack patterns.

The paper's findings are based on experiments with classification trees, which incorporated the characteristics used for classification, computed feature importance, and evaluated the performance of the algorithm using confusion matrices and F1-scores. The algorithm's transparency is its primary advantage, as it allows one to analyze how the model classifies observations and identify the features that contribute to misclassifications. This is especially important for the given task, as it allows one to distinguish between attack types based on the characteristics that differentiate them from each other.

The research's limitation is the use of synthetic data for experiments, which does not reflect real-world scenarios. Therefore, while the paper shows that Decision Trees can be used to classify and detect cyberattacks, its findings might not be generalized to different data sets. Future research should consider conducting similar experiments with actual data sets and addressing other aspects of attack classification, such as the impact of depth and breadth of analyzed features on the results. In addition, future studies should compare the performance of the Decision Tree algorithm to other machine learning algorithms and evaluate their effectiveness in solving the given task.

7. References

Ahmadi, S. (2024). AI in the detection and prevention of distributed denial of service (DDoS) attacks. International Journal of Advanced Computer Science and Applications, 15(10). https://doi.org/10.14569/IJACSA.2024.0151004

Al-Odat, Z. (2025). Optimized distributed gradient boosting with explainable artificial intelligence for real-time ransomware detection in networked environments. International Journal of Electrical and Electronic Engineering & Telecommunications, 14(4).

Blockeel, H., Devos, L., Frénay, B., Nanfack, G., & Nijssen, S. (2023). Decision trees: From efficient prediction to responsible AI. Frontiers in Artificial Intelligence, 6, Article 1124553. https://doi.org/10.3389/frai.2023.1124553

Chakraborty, A., Biswas, A., & Khan, A. K. (2023). Artificial intelligence for cybersecurity: Threats, attacks and mitigation. In Artificial intelligence for societal issues (pp. 3–25). Springer. https://doi.org/10.1007/978-3-031-12419-8_1

Ferdous, J., Islam, R., Mahboubi, A., & Islam, M. Z. (2024). AI-based ransomware detection: A comprehensive review. IEEE Access, 12, 136666–136695. https://doi.org/10.1109/ACCESS.2024.3461965

Foley, J. (2026). Cybersecurity future threats in artificial intelligence: A comprehensive analysis. Journal of Advanced Artificial Intelligence, Engineering and Technology, 2(1). https://doi.org/10.56147/aaiet.2.1.113

George, R., & Zeratsion, E. (2025). Detection of SSH brute-force attacks using machine learning: A comparative study with Fail2Ban and PAM Tally2 [Master's thesis, University West].

Hamza, A. A., & Al-Janabi, R. J. S. (2024). Detecting brute force attacks using machine learning. BIO Web of Conferences, 97, Article 00045. https://doi.org/10.1051/bioconf/20249700045

Ismaeel, M., Lokhande, P., & Kotiyal, B. (2025). Enhancing web security detection with AI-driven mitigation techniques. In Proceedings of the MULTINOVA: First International Conference on Artificial Intelligence in Engineering, Healthcare and Sciences (ICAIEHS-2025) (pp. 181–194). Atlantis Press. https://doi.org/10.2991/978-94-6463-852-3_12

Kalgude, K. (2022). Bug detection using artificial intelligence and cyber security [Master's dissertation, Solent University].

Kang, G. (2025). Artificial intelligence for threat detection: Leveraging deep learning to identify zero-day attacks in real time. Universal Library of Engineering Technology, 2(4), 74–79. https://doi.org/10.70315/uloap.ulete.2025.0204013

Kaushal, H. (2024). XSS exploit generation using AI. International Journal of Research Publication and Reviews, 5(3), 4875–4881.

Kumari, P., Raj, R., Bhandari, H., Goenka, K., & Anjana, R. (2026). AI based cyber threats and vulnerability detection, prevention and prediction system. International Journal for Research in Applied Science & Engineering Technology, 14(6), 85–95. https://doi.org/10.22214/ijraset.2026.83342

Mereani, F. A., & Howe, J. M. (2018). Detecting cross-site scripting attacks using machine learning. In Advanced machine learning technologies and applications (Vol. 723, pp. 200–210). Springer. https://doi.org/10.1007/978-3-319-74690-6_20

Njie, B., & Gabriouet, L. (2024). Machine learning for cross-site scripting (XSS) detection: A comparative analysis of machine learning models for enhanced XSS detection [Bachelor's thesis, Dalarna University]. DiVA. https://www.diva-portal.org/smash/get/diva2%3A1883365/FULLTEXT01.pdf

Patil, P. J., Kulkarni, S., & Kulkarni, V. T. (2026). AI-generated phishing attacks and detection methods. International Journal of Scientific Research in Engineering and Management, 10(6).

Pamarthi, K. (2020). Artificial intelligence and machine learning techniques to control SQL injection attacks. Journal of Scientific and Engineering Research, 7(4), 259–267.

Radaković, M. U., & Marjanović, M. D. (2023). Analyzing a Python programming example of building an artificial intelligence machine learning model for detecting cyber-attacks—Benefits and challenges. Tehnika, 78(2), 187–194. https://doi.org/10.5937/tehnika2302187R

Sinchana, H. C., & Thejaswini, M. N. (2025). AI based phishing attack detection. International Research Journal of Modernization in Engineering Technology and Science, 7(10), 218–221.

Sreedharan, R., & Sampath, A. K. (2026, July 28). Prediction of SQL injection using machine learning [Preprint]. Research Square. https://doi.org/10.21203/rs.3.rs-10122490/v1

Zunaid, M., Puneeth, M. P., Abhishek, P., Kumar, R. P., & Rahaman, M. T. R. (2026). Phishing website detection using URL-based machine learning for real-time browser security. International Journal of Advanced Research in Computer and Communication Engineering, 15(4). https://doi.org/10.17148/IJARCCE.2026.154274