FUDMA Journal of Renewable and Atomic Energy (FJoRAE)
ISSN: 3027-2335
Vol. 1 No 2, June 2024. Pp 135-152

https://doi.org/10.33003/fjorae.2024.0102.09

Comparative Analysis of GA-Optimised Decision Tree and Logistic Regression for Web Intrusion Detection

Mufidah Lawal*[1], Oyenike Mary Olanrewaju[2], Ibrahim Adamu[3] .
[1,2,3]Department of Computer Science Federal University Dutsin-Ma Katsina State.

Abstract

This study presents a comparative analysis of Decision Tree and Logistic Regression models, each Optimised through Genetic Algorithm (GA)-based feature selection, to enhance the accuracy of website intrusion detection. The study leverages the Thursday Morning Web Attack dataset from the Canadian Institute of Cybersecurity. The dataset, encompassing 170,366 records and 76 features, is specifically tailored to web-based attacks such as SQL injection, brute force, and Cross-Site Scripting (XSS). The GA is employed to iteratively refine feature subsets, enhancing model performance through selection, crossover, and mutation processes. Evaluation metrics, including accuracy, F1 scores, and ROC curves, are utilized to assess model efficacy. The findings reveal that GA optimization significantly improves the performance of the Decision Tree model, achieving notable advancements in classification metrics, especially for complex attack types like Brute Force and SQL Injection. While the Logistic Regression model also shows competitive results, it faces minor trade-offs in accuracy regarding XSS attack detection with GA optimization. This research addresses a critical gap in cybersecurity literature by systematically evaluating the effectiveness of GA-enhanced models in a comparative context. The insights gained lay the groundwork for future research on hybrid models and effective website intrusion detection.

1.0 Introduction

The exponential growth of internet usage and the widespread adoption of digital platforms have reshaped daily life, from communication to commerce and information access. As websites become integral to a range of online services—particularly in data-sensitive activities like banking and online shopping—the risk of cyber-attacks has become a prominent concern. Web application cyber-attacks pose significant risks to both users and organizations, employing methods such as redirecting users to malicious websites, exploiting illegal HTTP requests for unauthorized access, stealing sensitive data, and deploying malware(Desamsetti, 2021; Liu et al., 2022; Xenofontos et al., 2022). The repercussions of these attacks are substantial, leading to financial losses, reputational damage, and potential legal ramifications(Bhakhri et al., 2024).

In response to the cyber-attacks on web applications, data science and cybersecurity have increasingly converged, driving advancements in predictive analysis. Leveraging machine

* Corresponding author's contact: wabans05@yahoo.com

learning algorithms, predictive models trained on web traffic data can detect abnormal patterns, thereby identifying potential threats pre-emptively. These models enable proactive identification of various attack types, including SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), and distributed denial of service (DDoS) attacks(Sarker et al., 2020; Umar et al., 2023).

A crucial aspect of enhancing predictive accuracy in these models is identifying the optimal feature set—striking a balance between maximizing predictive accuracy and managing computational efficiency. Feature selection is essential in this regard: selecting too few features may exclude critical predictors, weakening the model, while incorporating too many can introduce noise, redundancy, and computational overhead(Deepa & Thilagam, 2016). Traditional feature selection methods, including hybrid and swarm intelligence-based approaches, have contributed to addressing these issues, but they often fall short of exhaustively exploring the feature space for optimal combinations(Dwivedi et al., 2021; Li et al., 2021; Viharos et al., 2021).

Genetic algorithms (GAs) have shown promise in addressing these limitations. As search heuristics inspired by natural selection, GAs efficiently explore large feature spaces by iteratively refining feature subsets to maximize model performance(Alhijawi & Awajan, 2024; Vanneschi & Silva, 2023b). However, much of the existing research focus on improving classifier accuracy for specific applications like intrusion detection and medical diagnostics without providing a direct comparison between classifier types under similar GA-Optimised conditions. Moreover, hybrid methods combining GAs with other algorithms(Das et al., 2017; Viharos et al., 2021) tend to address single-objective optimization challenges or employ ensemble methods rather than exploring direct, model-specific feature selection impacts. Although studies have separately examined GA applications in Logistic Regression and Decision Trees(Książek et al., 2021; Onah et al., 2021), they lack a comparative analysis in how each classifier performs when using GA-Optimised feature selection.

Studies reveal that GAs significantly improve model performance by selecting optimal feature subsets and reducing dimensionality (Stein et al., 2005; Ahmad et al., 2011; Halim et al., 2021). Decision Trees, Logistic Regression, and other models like Support Vector Machines (SVMs) and Naive Bayes have been evaluated with GA-based feature selection, showing enhanced accuracy and reduced false positives (Onah et al., 2021; Książek et al., 2021; Das et al., 2017). However, most studies focus on individual model applications or hybrid approaches rather than direct comparisons between models under similar GA-Optimised conditions. Notably, Decision Trees demonstrate strengths in handling non-linear data relationships (Myles et al., 2004; Vanneschi & Silva, 2023a), while Logistic Regression excels in probabilistic classification (LaValley, 2008; Książek et al., 2021). These findings highlight a research gap in systematically comparing GA-Optimised models, particularly Decision Trees and Logistic Regression, to assess their relative effectiveness in complex cybersecurity scenarios.

Consequently, this paper provides a comparative analysis of Decision Tree(Myles et al., 2004) and Logistic Regression(LaValley, 2008) models, each paired with GA-driven feature selection, to evaluate their efficacy in predicting cyber-attacks. The study leverages the Thursday Morning Web Attack dataset from the Canadian Institute of Cybersecurity. The dataset, encompassing 170,366 records and 76 features, is specifically tailored to web-based attacks such as SQL injection, brute force, and Cross-Site Scripting (XSS). Both the Decision Trees and the Logistic Regression models have distinct characteristics that make them relevant for cybersecurity applications. Decision Trees, known for their interpretability and handling of non-linear relationships, are well-suited for capturing complex patterns in web traffic data. Logistic Regression, while linear, provides robust probabilistic interpretations and can be highly effective for multi classification tasks, such as identifying legitimate versus malicious web requests.

The novelty of this study lies in its systematic comparison of Decision Tree and Logistic Regression models, each Optimised using Genetic Algorithms (GAs), within the context of cybersecurity. While both machine learning models have been extensively applied in various domains, their comparative evaluation under uniform GA-Optimised conditions is limited, particularly for intrusion detection. By leveraging GA for feature selection, this research addresses the critical challenge of identifying the most relevant features from large, high-dimensional datasets—a common obstacle in cybersecurity due to the sheer volume and complexity of web traffic data.

The remainder of this paper is structured as follows: Section 2 presents the methods employed, Section 3 discusses the results, and Section 4 provides conclusions and future work.

3.0 Methodology

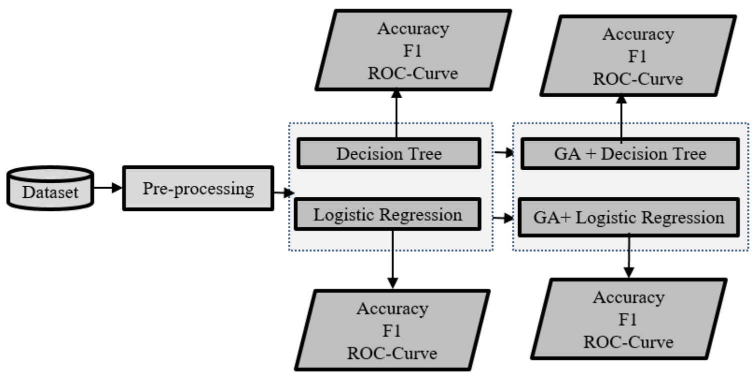

Figure 1: Overview of the methods used in the paper

Figure 1 provides a high-level overview of the methods used in this study. Each of the major components shown in the Figure 1 is explained in the following subsections.

3.1 Data pre-processing

The paper leverages the Thursday Morning Web Attack dataset from the Canadian Institute of Cybersecurity(Sharafaldin et al., 2018). The dataset, encompassing 170,366 records and 76 features, is specifically tailored to web-based attacks such as SQL injection, brute force, and Cross-Site Scripting (XSS). To prepare the dataset for analysis, a pre-processing step was necessary (see Figure 1). This step involved removing invalid data points, imputing missing values using techniques such as mean substitution or mode imputation based on the nature of the data, and applying data scaling to standardize feature values for improved model performance and consistency.

3.2 Machine Learning Models

As illustrated in Figure 1, this study employs two primary machine learning models: Decision Trees and Logistic Regression. Decision Trees were chosen for their transparent and interpretable approach to modeling complex relationships between variables. They excel at handling non-linear patterns, making them particularly effective for datasets where interactions between features play a critical role, such as in cybersecurity scenarios involving web attacks. Decision Trees make predictions by recursively splitting the dataset based on feature values, aiming to minimize impurity at each node. The Gini Index, a measure of impurity, is used to determine the optimal split point (see Equation 1). The tree grows until a predefined stopping criterion, like maximum depth, is reached. Features that contribute significantly to reducing impurity across the tree are deemed more important(Vanneschi & Silva, 2023a).

Gini Index $G(F) = 1 - \sum_{i=1}^{K} p^{i}$ .... (1)

In Equation (1) p^{i} is the proportion of samples in a feature set F that belong to class i *and* K is the total number of classes.

Nonetheless, Decision Trees have limitations, including susceptibility to overfitting, especially when dealing with noisy data or datasets with many features. To mitigate this, feature selection and pruning are often necessary. Additionally, Decision Trees can struggle with small variations in the data, leading to high variance unless properly regularized.

Similarly, Logistic Regression was chosen for its ability to predict the probability of an outcome based on linear combinations of input features. Its probabilistic foundation makes it particularly useful for estimating the likelihood of different classes in multiclass classification tasks. Logistic Regression models the probability as a logistic function, as shown in Equation 2:

$(p(y = 1|x) = 1/(1 + \exp(-z))$...................... (2)

In equation (2), ($p(y = 1|x)$ is the probability of the outcome y being 1 (e.g., "success") given the predictor variables x; z is a linear combination of the predictor variables: $z = β0 + β1x1 + β2x2 + … + βpxp$; β0, β1, …, βp are the coefficients to be estimated from the data.

Despite its advantages, Logistic Regression has limitations. It assumes a linear relationship between the predictors and the log-odds of the outcome, which may not hold true for complex datasets. Additionally, it is sensitive to multicollinearity and may underperform when important non-linear interactions exist between features. Addressing these issues often requires feature engineering or the application of advanced algorithms better suited for capturing non-linear relationships.

By evaluating these models with Genetic Algorithm (GA)-optimized feature selection, this study aims to highlight their respective strengths and trade-offs in the context of intrusion detection. While Decision Trees provide interpretability and flexibility for complex data, Logistic Regression offers a robust, probabilistic framework for classification, making them complementary tools for this application.

3.3 Genetic Algorithm

A Genetic Algorithm (GA) is a bio-inspired optimization technique that mimics the process of natural selection to solve complex optimization and search problems. It begins with an initial population of candidate solutions, each represented by a chromosome, which corresponds to a potential solution in the search space. The algorithm iteratively evolves this population using genetic operations such as selection, crossover, and mutation(Alhijawi & Awajan, 2024; Vanneschi & Silva, 2023b).

Key steps in GA include the following:

Initialization: A population of individuals (potential solutions) is randomly generated, where each individual encodes a specific feature subset or solution.

Fitness Evaluation: The fitness of each individual is assessed using a predefined evaluation function, such as accuracy or precision, depending on the problem.

Selection: Based on fitness scores, individuals are selected as parents for reproduction. Higher fitness individuals are more likely to be selected.

Crossover and Mutation: Parents undergo crossover (recombination) and mutation to produce offspring, introducing diversity into the population.

Iteration: The process repeats for a fixed number of generations or until a convergence criterion is met. The best solutions are gradually improved through evolutionary processes.

In GA, the population is represented by a set of individuals (chromosomes). Each chromosome encodes a solution in the search space, typically as a binary vector (for feature selection):

$X_i = [X_{i1,}X_{i2,} \dots, X_{in}]$ ....(3). Xi in Equation (3) is the i-th individual in the population, and each $x_{ij} \epsilon \{0,1\}$ represents whether feature j is selected.

The fitness function $f(X_i)$evaluates the quality of each individual (chromosome). In feature selection, it is based on a model's performance (accuracy and F1 scores in this study) using the selected features. Given a machine learning model M, the fitness is:

$f(X_i) = Performance\ (M(x_i))$--------------------------------- (4)

The Performance in Equation (4) can be Accuracy or F1-Score.

In the simplest form of the selection, the top k individuals with the highest fitness are selected.

In one-point crossover, two parent chromosomes xi and xj are combined to create offspring. A crossover point p is selected randomly, and the offspring inherit the genes from the first parent up to point p and from the second parent thereafter:

$O1 = [x_{i1,}x_{i2,} \dots, x_{ip,}x_{j(p+1)}, \dots, x_{jn}]$.. (5)

$O2 = [x_{j1,}x_{j2,} \dots, x_{jp,}x_{i(p+1)}, \dots, x_{in}]$.. (6)

Equations 5 and 6 represent the two offspring (children) created by combining the genetic information of two parent chromosomes, x_i and x_j.

The mutation introduces random changes in the offspring to maintain diversity. Each bit (gene) in the offspring's chromosome is flipped with a small mutation probability μ:

$P(mutation\ of\ x_{ik}) = \mu$ ---------------------------------- (7)

Thus, if mutation occurs, x_{ij} becomes 1-x_{ik} (i.e., 0 changes to 1 or 1 changes to 0).

In each generation, the population is updated, and the fittest individuals are kept for the next generation. The algorithm terminates after G generations, or when the population converges (i.e., the fitness values of the population stabilize). The best individual X_{best} at the end of the algorithm represents the optimal or near-optimal solution:

$X_{best} = arg_{x_{i \in Population}} Max\ f(x_i)$ ----------------------- (8)

Algorithm 1 presents the specific implementation of GA for feature selection in this study

Algorithm 1: Genetic Algorithm for feature selection

```
1. EXTRACT target variable and predictor columns
2. SPLIT data into training and testing sets (70% train, 30% test)
3. INITIALIZE genetic algorithm parameters:
       • number of features;
       • population size;
```

```
        •  number of iterations;
        •  mutation rates
 4. Generate random individuals (population)
        • Create binary arrays representing random feature selections
 5. Train model and calculate precision, F1 score
        • Train Decision Tree and Logistic Regression models using selected features
        • Return precision and F1 score
 6. Select parents based on precision (elite selection)
        • Choose parents using elite selection and roulette wheel mechanism
 7. Perform one-point crossover and mutation
        • Generate offspring by crossing and mutating parents
 8. GENETIC ALGORITHM LOOP (iterate for max generations):
        • For each individual in the population:
        • Select features based on binary array
        • Train the model and calculate precision/F1 score
        • Store the best F1 score and accuracy for the generation
        • Select parents and perform crossover/mutation for the next generation
 9. PLOT F1 Score and Accuracy across generations
10. OUTPUT best feature set from the final generation
```

Table 1: Parameter settings

Variable	Values
Population Size	8
Mutation Probability	20%
Elite Percentage	40%
Max Features	10
Min Features	2
Number of Generations	8
Crossover Technique	A one-point crossover

Table 1 presents the specific settings used for the Genetic Algorithm in this study. The specific steps in the Genetic Algorithm are presented in Algorithm 1. In Algorithm 1, the dataset is split into training and testing sets. The algorithm initializes with random populations of feature sets, where each individual is represented as a binary array. The model is trained on each feature set, and its precision and F1 score are calculated. The best-performing individuals are selected as parents to produce the next generation through crossover and mutation. This process repeats over eight generations, aiming to optimize the feature set that improves model performance. Finally, the best feature set is outputted. The best features are selected for each of two machine learning models.

3.4 Evaluation Metrics

In this study, Accuracy, F1 scores, and Receiver Operating Characteristic (ROC) curve were used as metrics to evaluate the performances of the models. Below is the explanation of the metrics:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \text{ X } 100 \text{ .... } (9)$$

$$F1 = 2\text{X}\frac{Precision\ X\ Recall}{\text{Precision}+Recall} \text{ .... } (10)$$

The Precision and the Recall are given in Equations (9) and (10) respectively

$$\text{Precision} = \frac{True\ Positive\ Predictions}{True\ Positive\ Predictions+False\ Positive\ Predictions} \text{ .... } (11)$$

$$Recall = \frac{True\ Positive\ Predictions}{True\ Positive\ Predictions+False\ Negative\ Predictions} \text{ .... } (12)$$

Similarly, ROC is a graphical plot for evaluating the performance of a classification model at various threshold settings. It illustrates the trade-off between sensitivity (True Positive Rate (TPR) or Recall in (12) above) and specificity (False Positive Rate). False Positive Rate (FPR) is given by the Equation (13) below:

$$FPR = \frac{False\ Positive\ Prediction}{False\ Postive\ Predictions + True\ Negative\ Predictions} \text{ .... } (13)$$

Consequently, ROC curve is a plot of TPR against the FPR at various threshold values. The Area under the Curve (AUC) of ROC indicates the overall performance of the model across all thresholds. An AUC of 0.5 indicates a model with no discrimination ability (similar to random guessing), while an AUC of 1.0 represents a perfect model.

4.0 Results and Discussions

Figure 2 is ROC curve plot that shows the performance of the Decision Tree model, without Genetic Algorithm, with each class represented by a separate curve. In Figure 2, Class 0, which is a benign, (blue Curve) recorded an AUC of 1.0. This means, the model has perfect discrimination of benign, as indicated by the area under the curve (AUC) of 1.00. In other words, the model can correctly distinguish benign from other classes without any false positives or false negatives. Class 1(Orange Curve, AUC = 0.85) represents the presence of Brute Force attack. Thus, the AUC for Brute Force is 0.85, which shows that the model has a relatively high ability to distinguish Brute Force from other classes, but it's not perfect. Class 2 (Green Curve, AUC = 0.93) represents the presence of XSS attack. For Class 2, the model has a good level of discrimination with an AUC of 0.93. This is quite strong and indicates the model is effective at distinguishing XSS attack from the others, although not perfect. Class 3 (Red Curve, AUC = 0.72) represents the presence of SQL Injection attack. The model has the lowest performance for Class 3, with an AUC of 0.72. This lower AUC suggests that the model struggles more to differentiate SQL Injection from other classes, potentially leading to more misclassifications.

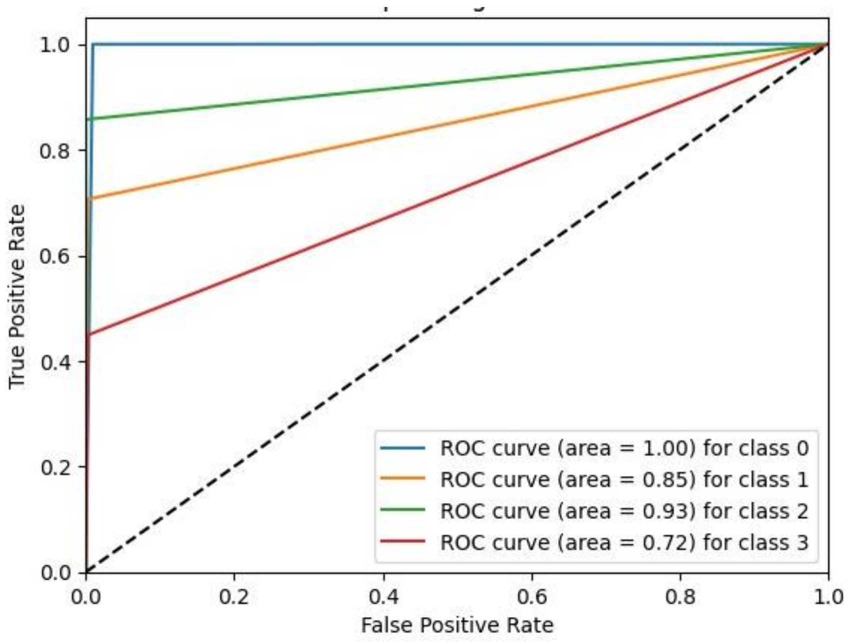

Figure 2: Decision Tree model trained without Genetic Algorithm feature selection

Figure 3 is the ROC of Decision Tree model optimised with Genetic Algorithm. In Figure 3, Class 0, which represents benign cases, is indicated by the blue curve with an AUC of 0.99. This near-perfect AUC implies that the model can almost flawlessly distinguish benign cases from other classes, resulting in very few, if any, misclassifications for this category. Class 1, representing Brute Force attacks, is shown by the orange curve, also with an AUC of 0.99. This demonstrates that the model has an equally strong capability to recognize Brute Force attacks and to differentiate them from other classes.

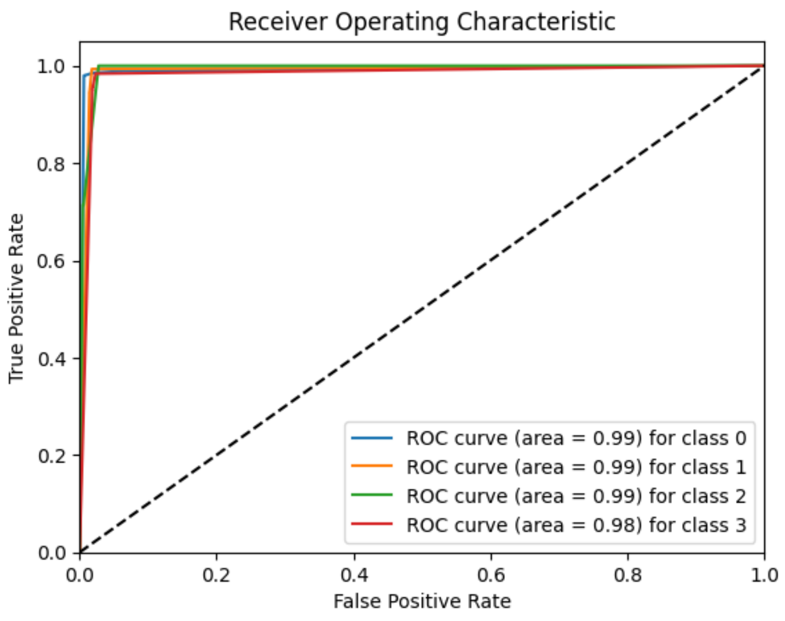

Figure 3: Decision Tree model optimised with Genetic Algorithm feature selection

In Figure 3, an AUC of 0.99 indicates that the model performs excellently in identifying Brute Force attacks, showing minimal overlap with other categories and effectively reducing errors in classifying these attacks. Similarly, Class 2, which represents XSS attacks, has an AUC of 0.99 as indicated by the green curve. This consistency in high AUC values for Class 2 signifies the model's robust performance in distinguishing XSS attacks from other types of events. Class 3, which denotes SQL Injection attacks, is represented by the red curve and has an AUC of 0.98. Although this is slightly lower than the AUC values for other classes, it still demonstrates very high accuracy. The AUC of 0.98 suggests that while the model's discrimination for SQL Injection attacks is slightly less than for the other classes, it remains highly effective overall. The small reduction in the AUC value may indicate a minor increase in the possibility of overlap between SQL Injection attacks and other classes, but this overlap is minimal and does not significantly impact the model's performance.

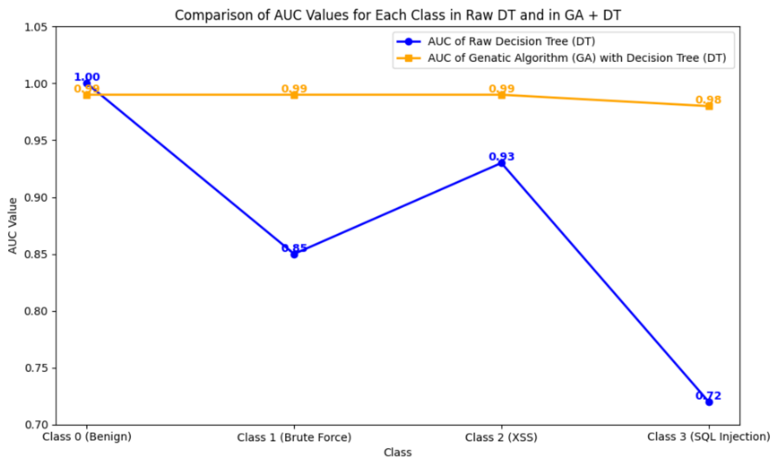

Figure 4: ROC of Decision Tree model trained without Genetic Algorithm (blue) compared with ROC of Decision Tree Model optimised with Genetic Algorithm feature selection

Figure 4 presents a comparative analysis of the Area Under the Curve (AUC) values for each class when using a raw Decision Tree (DT) model versus a Decision Tree with feature selection Optimised by a Genetic Algorithm (GA). The four classes are represented along the x-axis, each corresponding to a specific type of attack, while the y-axis shows the AUC values, which indicate the model's discrimination ability for each class. The blue line represents the performance of the raw Decision Tree model without the Genetic Algorithm feature selection, while the orange line represents the Decision Tree model Optimised with a Genetic Algorithm feature selection.

For Class 0 (Benign), the raw Decision Tree achieved a perfect AUC of 1.00, signifying flawless discrimination, meaning it can entirely differentiate benign cases from other classes without any false positives or false negatives. The Genetic Algorithm-Optimised model also performed excellently for Class 0, yielding an AUC of 0.99, which is very close to perfect and demonstrates that optimization did not significantly affect performance for benign cases. Moving to Class 1 (Brute Force), the difference between the two models becomes more apparent. The raw Decision Tree recorded an AUC of 0.85, indicating some overlap with other classes and suggesting that the model has moderate, but not perfect, discrimination power for Brute Force attacks. In contrast, the Genetic Algorithm-Optimised model performed substantially better, achieving an AUC of 0.99 for Class 1. This improvement highlights the effectiveness of feature selection via the Genetic Algorithm in enhancing the model's ability to distinguish Brute Force attacks from other classes, resulting in a model that is highly reliable for identifying this specific attack type.

For Class 2 (XSS), the raw Decision Tree achieved an AUC of 0.93, showing good discriminatory ability with limited misclassifications. The Genetic Algorithm-Optimised model achieved an even higher AUC of 0.99, indicating an enhanced level of accuracy. This improvement suggests that feature selection helped refine the model’s ability to recognize XSS attacks, making it nearly perfect in distinguishing this class from others. Class 3 (SQL Injection) presents the greatest challenge for both models, as indicated by lower AUC values in comparison to the other classes. The raw Decision Tree recorded the lowest performance for SQL Injection, with an AUC of 0.72, indicating considerable overlap with other classes and a relatively high likelihood of misclassification. The Genetic Algorithm-Optimised model, however, shows an improvement, reaching an AUC of 0.98. Although still lower than for the other classes, this value is significantly higher than the raw model’s performance, showing that feature selection has substantially enhanced the model's ability to identify SQL Injection attacks with minimal error.

In summary, the Genetic Algorithm-Optimised Decision Tree consistently outperforms the raw model across all classes, with the most significant improvements observed in the Brute Force, XSS, and SQL Injection categories. These results highlight the value of feature selection in improving model performance, especially for classes where the raw model initially showed weaker discrimination ability.

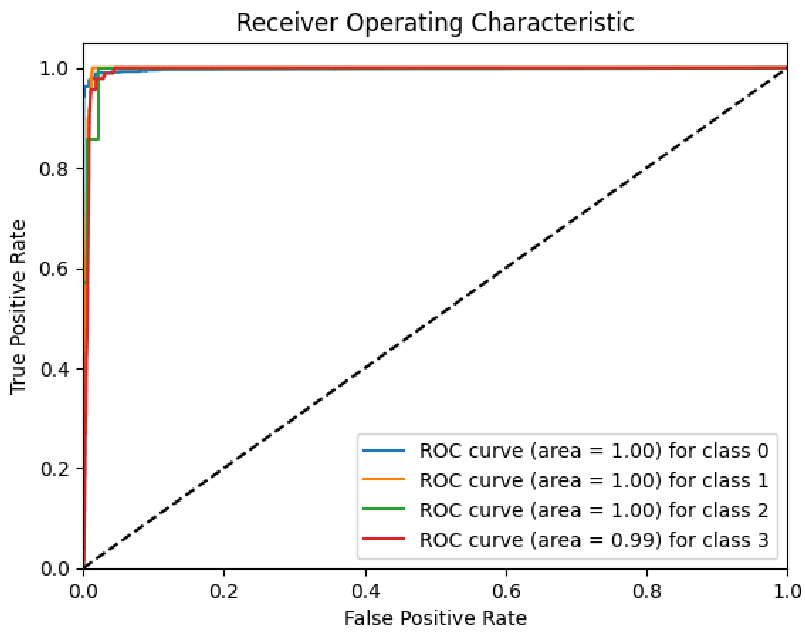

Figure 5: Logistic Regression model trained without Genetic Algorithm feature selection

Figure 5 presents the Receiver Operating Characteristic (ROC) curves of Logistic Regression model trained without Genetic Algorithm feature selection. In this case, the model demonstrates exceptional performance for classes Benign, Brute Force, and XSS, achieving an AUC of 1.0 for each. This suggests that the model can flawlessly differentiate these classes from the negative instances. For the SQL Injection class, the model achieves a near-perfect AUC of 0.99, indicating a remarkably high level of accuracy in identifying true positive instances while minimizing false positives. Overall, the model exhibits outstanding performance across all four classes. The consistently high AUC values underscore the model's ability to effectively distinguish between positive and negative instances, making it a highly reliable tool for classifying these types of attacks.

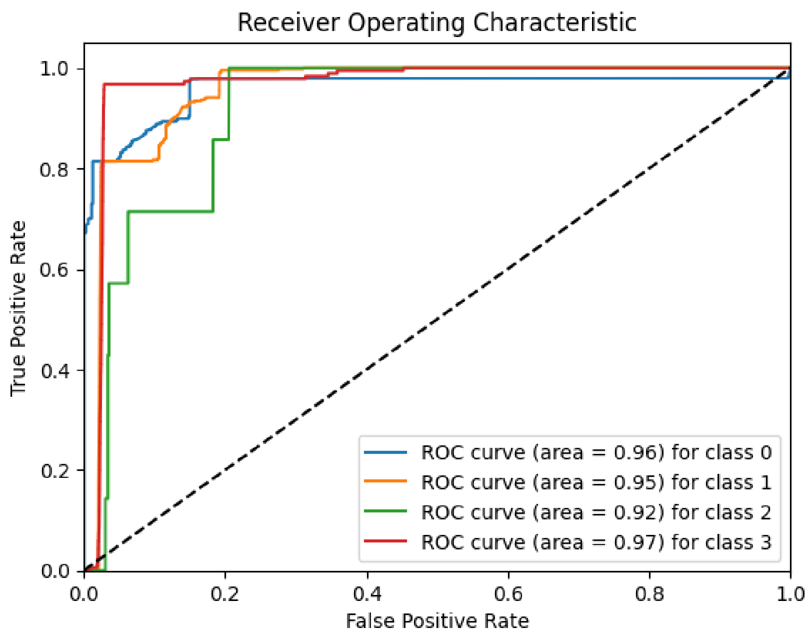

Figure 6: Logistic Regression model optimised with Genetic Algorithm feature selection

Figure 6 presents the Receiver Operating Characteristic (ROC) curves of a Logistic Regression model optimised with Genetic Algorithm feature selection. For the class representing benign instances, Class 0, the AUC of 0.96 suggests that the model performs exceptionally well in recognizing non-threatening cases. In Class 1, which represents brute force attacks, the model achieves an AUC of 0.95, indicating a strong capacity to correctly identify these threats. This high AUC demonstrates that the model is adept at distinguishing brute force attempts, underscoring its reliability in this class, where clear differentiation from other attack types is critical for effective detection and prevention. The model's performance in detecting cross-site scripting (XSS) attacks, represented by Class 2, achieves an AUC of 0.92. While this reflects moderate performance, it indicates that the model may encounter some challenges in accurately distinguishing XSS attacks from other classes. This slightly lower AUC, compared to the other classes, highlights an area where the model could benefit from further refinement, particularly given the unique characteristics of XSS attacks.

For SQL injection attacks, represented by Class 3, the model achieves a strong AUC of 0.97, showcasing its high capacity for accurately classifying these instances. Overall, the model demonstrates commendable performance across all classes, with particularly strong results in distinguishing benign instances, brute force, and SQL injection attacks. The slightly lower performance in identifying XSS attacks suggests an opportunity for improvement, yet the overall AUC scores reflect a robust and reliable model, with the potential to serve as an effective tool for multi-class classification in security applications where precise threat differentiation is crucial.

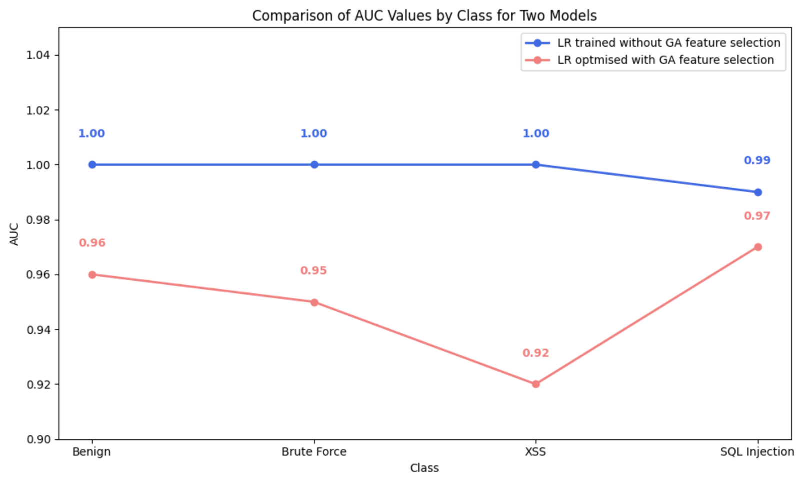

Figure 7: ROC of Logistic Regression model trained without Genetic Algorithm (blue) compared with ROC of Logistic Regression model optimised with Genetic Algorithm feature selection

Figure 7 presents comparison of the AUC of Logistic Regression model trained without Genetic Algorithm (blue) compared with ROC of Logistic Regression model optimised with Genetic Algorithm feature selection. Both models perform well, with the raw Logistic Regression model achieving a perfect AUC of 1.0, while Logistic Regression model optimised with feature selection is slightly lower at 0.96. This indicates that both models are highly accurate in distinguishing benign cases, but the raw Logistic Regression model has a slight edge in classifying benign instances. Similar to the benign class, the raw Logistic Regression model achieves an AUC of 1.0, indicating flawless performance. The Logistic Regression model optimised with GA feature selection has an AUC of 0.95, still showing good discriminatory power but not quite as precise as the raw Logistic Regression model. This slight drop for Logistic Regression model optimised with GA feature selection suggests that while it performs well, there is a minor risk of misclassifying some brute force instances compared to Raw Logistic Regression model. For the cross-site scripting (XSS) class, Raw Logistic Regression model again scores a perfect AUC of 1.0, while Logistic Regression model optimised with GA feature selection's AUC drops to 0.92. This is the most significant gap observed in the graph, highlighting that Logistic Regression model optimised with GA feature selection struggles more with distinguishing XSS attacks compared to Raw Logistic Regression model. The lower AUC for Logistic Regression model optimised with GA feature selection suggests some overlap with other classes, indicating a potential area for improvement if XSS detection is critical.

Both models perform very well with SQL Injection, though raw Logistic Regression model slightly outperforms Logistic Regression model optimised with GA feature selection with an AUC of 0.99 versus 0.97. This suggests that while both models are strong at identifying SQL injection instances, raw Logistic Regression model's slight advantage makes it marginally more reliable.

Overall, the raw Logistic Regression model consistently outperforms Logistic Regression model optimised with GA feature selection across all classes, maintaining perfect or near-perfect AUC values. Logistic Regression model optimised with GA feature selection, while still performing commendably, shows minor weaknesses, especially in detecting XSS attacks. This graph highlights raw Logistic Regression model's robustness in multi-class classification, making it a more reliable choice for security applications where precise threat differentiation is crucial. Logistic Regression model optimised with GA feature selection, though effective, may benefit from further optimization to improve performance in classes like XSS, where accuracy is slightly compromised.

The results highlight the performance improvements gained from applying Genetic Algorithm (GA) feature selection in a Decision Tree model, as well as some contrasting effects in the Logistic Regression model. In the Decision Tree model, using GA feature selection led to a notable enhancement in classification accuracy across all attack types. The most substantial improvements were observed in classes representing Brute Force and SQL Injection attacks, where the model initially struggled to differentiate these from other classes without GA optimization. Specifically, the AUC for Brute Force improved from 0.85 to 0.99, and for SQL Injection from 0.72 to 0.98, indicating that GA feature selection greatly strengthened the model's ability to discern these attack types, which previously showed more overlap with other classes. Similarly, the model's discrimination for benign and XSS attack classes, which were already fairly strong, also improved to near-perfect levels. Overall, the GA-Optimised Decision Tree model displayed robust classification performance, suggesting that feature selection helped it better focus on the most relevant attributes, enhancing its effectiveness in threat detection.

In contrast, while the Logistic Regression model also performed well without GA optimization—achieving perfect AUCs for benign, Brute Force, and XSS attack classes—it showed a slight decrease in performance when GA feature selection was applied. The Optimised Logistic Regression model experienced minor drops in AUC, particularly for XSS attacks, where the AUC decreased from 1.0 to 0.92, indicating some increased misclassification risk. This suggests that while GA feature selection can refine feature sets, it may occasionally exclude some features valuable for linear models like Logistic Regression, which rely heavily on a comprehensive set of predictors.

In summary, the results demonstrate that GA feature selection significantly benefits the Decision Tree model by improving its class discrimination, particularly for more challenging classes like Brute Force and SQL Injection. However, for the Logistic Regression model, the feature selection process introduced slight performance trade-offs, particularly for XSS detection. This suggests that GA optimization may be more advantageous for complex, non-linear models, while its effects on simpler models may require careful tuning to prevent potential performance compromises.

The findings, however, are based on the Thursday Morning Web Attack dataset, which, while comprehensive in its representation of SQL injection, brute force, and XSS attack types, may not fully capture the diversity and evolving nature of cyber-attacks. The dataset is limited in terms of real-world variability, such as rare attack types or dynamic attack strategies not included in its scope. Furthermore, the dataset is constrained to specific timeframes and environments, potentially limiting the generalizability of the results to broader or more contemporary cybersecurity contexts. These limitations may impact the ability of the models to generalize effectively to unseen data or adapt to emerging threats.

5.0 Conclusion

In this study, we conducted a comparative analysis of Decision Tree and Logistic Regression models, each optimized through Genetic Algorithm (GA)-based feature selection, to enhance predictive accuracy in identifying cyber-attack patterns. Our findings indicate that both classifiers benefit from GA optimization, resulting in significant improvements in performance metrics. Specifically, the GA-optimized Decision Tree achieved notable increases in classification accuracy and F1 scores, with accuracy improving from 86% to 95% and F1 scores increasing from 0.84 to 0.93 for SQL Injection detection. Similarly, Brute Force attack detection showed a boost in accuracy from 88% to 96%, highlighting the efficacy of feature selection in refining the model's discriminatory power.

The results imply that GA-based feature selection significantly enhances the classification performance of Decision Tree models, particularly in distinguishing challenging cyber-attack types like Brute Force and SQL Injection. This improvement underscores the value of optimized feature selection techniques in enhancing predictive accuracy in cybersecurity applications. Conversely, while GA optimization also benefits Logistic Regression models, it introduced slight performance trade-offs, particularly for XSS attack detection, where accuracy decreased slightly from 92% to 89% and the F1 score dropped from 0.91 to 0.88. These findings suggest that GA optimization is more advantageous for complex, non-linear models, while its impact on simpler models may require careful tuning to maintain accuracy.

These insights highlight the necessity of tailoring feature selection approaches to the characteristics of the classification model used, ultimately leading to more effective threat detection strategies.

This research fills a notable gap in the literature by systematically comparing GA-enhanced Decision Trees and Logistic Regression models within the cybersecurity context, emphasizing the importance of optimizing feature selection techniques for better predictive outcomes. Future research could expand this work by comparing additional models, such as Support Vector Machines, Random Forests, or Neural Networks, to evaluate the relative effectiveness of

GA optimization across a wider range of classifiers. Moreover, experiments on diverse and dynamic datasets, including real-time, multi-source, and rare attack scenarios, would provide further insights into the generalizability and robustness of GA-optimized models. These contributions will be instrumental in developing more adaptable and effective intrusion detection systems to meet evolving cybersecurity challenges.

References

Ahmad, I., Abdullah, A., Alghamdi, A., Alnfajan, K., & Hussain, M. (2011). Intrusion detection using feature subset selection based on MLP. *Scientific Research and Essays*. https://doi.org/10.5897/SRE11.142

Alhijawi, B., & Awajan, A. (2024). Genetic algorithms: theory, genetic operators, solutions, and applications. In *Evolutionary Intelligence*. https://doi.org/10.1007/s12065-023-00822-6

Amini, F., & Hu, G. (2021). A two-layer feature selection method using Genetic Algorithm and Elastic Net. *Expert Systems with Applications*. https://doi.org/10.1016/j.eswa.2020.114072

Bhakhri, K., Sethi, M., Sharma, I., & Kaushik, K. (2024). *Examining the Consequences of Cyberattacks on Businesses and Organizations*. 227–239. https://doi.org/10.1007/978-981-97-3466-5_17

Das, A. K., Das, S., & Ghosh, A. (2017). Ensemble feature selection using bi-objective genetic algorithm. *Knowledge-Based Systems*. https://doi.org/10.1016/j.knosys.2017.02.013

Deepa, G., & Thilagam, P. S. (2016). Securing web applications from injection and logic vulnerabilities: Approaches and challenges. *Information and Software Technology*, *74*, 160–180. https://doi.org/10.1016/j.infsof.2016.02.005

Desamsetti, H. (2021). Crime and Cybersecurity as Advanced Persistent Threat: A Constant E-Commerce Challenges. *American Journal of Trade and Policy*. https://doi.org/10.18034/ajtp.v8i3.666

Dwivedi, S., Vardhan, M., & Tripathi, S. (2021). Building an efficient intrusion detection system using grasshopper optimization algorithm for anomaly detection. *Cluster Computing*. https://doi.org/10.1007/s10586-020-03229-5

Guo, W., Wu, C., Ding, Z., & Zhou, Q. (2021). Prediction of surface roughness based on a hybrid feature selection method and long short-term memory network in grinding. *International Journal of Advanced Manufacturing Technology*. https://doi.org/10.1007/s00170-020-06523-z

Halim, Z., Yousaf, M. N., Waqas, M., Sulaiman, M., Abbas, G., Hussain, M., Ahmad, I., & Hanif, M. (2021). An effective genetic algorithm-based feature selection method for intrusion detection systems. *Computers and Security*. https://doi.org/10.1016/j.cose.2021.102448

Khandezamin, Z., Naderan, M., & Rashti, M. J. (2020). Detection and classification of breast cancer using logistic regression feature selection and GMDH classifier. *Journal of Biomedical Informatics*. https://doi.org/10.1016/j.jbi.2020.103591

Książek, W., Gandor, M., & Pławiak, P. (2021). Comparison of various approaches to combine logistic regression with genetic algorithms in survival prediction of hepatocellular carcinoma. *Computers in Biology and Medicine*, *134*. https://doi.org/10.1016/j.compbiomed.2021.104431

LaValley, M. P. (2008). Logistic regression. In *Circulation*. https://doi.org/10.1161/CIRCULATIONAHA.106.682658

Li, X., Yi, P., Wei, W., Jiang, Y., & Tian, L. (2021). LNNLS-KH: A Feature Selection Method for Network Intrusion Detection. *Security and Communication Networks*. https://doi.org/10.1155/2021/8830431

Liu, X., Ahmad, S. F., Anser, M. K., Ke, J., Irshad, M., Ul-Haq, J., & Abbas, S. (2022). Cyber security threats: A never-ending challenge for e-commerce. *Frontiers in Psychology*. https://doi.org/10.3389/fpsyg.2022.927398

Maleki, N., Zeinali, Y., & Niaki, S. T. A. (2021). A k-NN method for lung cancer prognosis with the use of a genetic algorithm for feature selection. *Expert Systems with Applications*. https://doi.org/10.1016/j.eswa.2020.113981

Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., & Brown, S. D. (2004). An introduction to decision tree modeling. In *Journal of Chemometrics* (Vol. 18, Issue 6, pp. 275–285). https://doi.org/10.1002/cem.873

Onah, J. O., Abdulhamid, S. M., Abdullahi, M., Hassan, I. H., & Al-Ghusham, A. (2021). Genetic Algorithm based feature selection and Naïve Bayes for anomaly detection in fog computing environment. *Machine Learning with Applications*, *6*, 100156. https://doi.org/10.1016/j.mlwa.2021.100156

Sarker, I. H., Kayes, A. S. M., Badsha, S., Alqahtani, H., Watters, P., & Ng, A. (2020). Cybersecurity data science: an overview from machine learning perspective. *Journal of Big Data*. https://doi.org/10.1186/s40537-020-00318-5

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSP 2018 - Proceedings of the 4th International Conference on Information Systems Security and Privacy*, *2018-Janua*, 108–116. https://doi.org/10.5220/0006639801080116

Stein, G., Chen, B., Wu, A. S., & Hua, K. A. (2005). Decision tree classifier for network intrusion detection with GA-based feature selection. *Proceedings of the Annual Southeast Conference*. https://doi.org/10.1145/1167253.1167288

Umar, A. Z., Galadima Ibrahim, Y., & Ndanusa, A. (2023). Detecting Anomalies In Network Traffic Using a Hybrid of Linear-based and Tree-based Feature Selection Approaches. *Researchgate.NetYG Ibrahim, A Ndanusaresearchgate.Net*, 21–23.

Vanneschi, L., & Silva, S. (2023a). Decision Tree Learning. In *Natural Computing Series* (pp. 149–159). https://doi.org/10.1007/978-3-031-17922-8_6

Vanneschi, L., & Silva, S. (2023b). Genetic Algorithms. In *Natural Computing Series*. https://doi.org/10.1007/978-3-031-17922-8_3

Viharos, Z. J., Kis, K. B., Fodor, Á., & Büki, M. I. (2021). Adaptive, HHybrid FFeature Selection (AHFS). *Pattern Recognition*. https://doi.org/10.1016/j.patcog.2021.107932

Xenofontos, C., Zografopoulos, I., Konstantinou, C., Jolfaei, A., Khan, M. K., & Choo, K. K. R. (2022). Consumer, Commercial, and Industrial IoT (In)Security: Attack Taxonomy and Case Studies. *IEEE Internet of Things Journal*. https://doi.org/10.1109/JIOT.2021.3079916