FUDMA Journal of Renewable and Atomic Energy (FJoRAE)
ISSN: 3027-2335
Vol. 1 No 2, June 2024. Pp 54-67

https://doi.org/10.33003/fjorae.2024.0102.04

Genetic Algorithm-Based Feature Selection for Optimizing Website Intrusion Detection Model

[1]Mufidah Lawal, [2]Armaya'u Zango Umar.

[1] Department of Computer Science Federal University Dustin-Ma, Katsina State
[2]Department of Software Engineering and Cybersecurity, Al-Qalam University Katsina

Abstract

The widespread use of the internet and digital platforms has made websites essential for communication, business, and accessing information. However, this reliance has also heightened cybersecurity concerns, as cyber-attacks on web applications pose risks such as data theft, malware installation, and redirection to malicious sites. Cybersecurity is increasingly leveraging machine learning and predictive analysis to proactively identify and mitigate potential web application attacks. Nonetheless, the key challenge in website attack prediction is identifying the most relevant features that balance accurate predictions with manageable computational overhead. While Genetic Algorithm-based feature selection techniques hold promise, their effectiveness should be evaluated on datasets containing common web attacks such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). Additionally, metrics beyond accuracy should be considered to assess their generalizability and overall performance. This research investigates the application of Genetic Algorithms (GA) for feature selection in website attack prediction using Decision Tree and Logistic Regression models. The algorithms were evaluated using accuracy and F1-score metrics. The findings revealed that feature selection using GA significantly reduces the number of features to up to 86.8% while maintaining or improving detection accuracy and reduced overfitting. This 86.8% reduction in features simplifies the model while saving resources. The findings highlight the potential of GA as a valuable method for feature selection in website attack prediction.

1.0 Introduction

The rapid proliferation of internet usage and the subsequent rise of digital platforms have transformed the way we communicate, conduct business, and access information. As the digital landscape continues to expand, websites have become integral to our daily lives, serving as gateways to a myriad of online services. Web applications inherently handle sensitive data and are employed to carry out business-critical activities such as banking and online shopping. Consequently, this widespread reliance on websites has also brought to the forefront a pressing concern: cybersecurity(Desamsetti, 2021; Liu et al., 2022). Cyber-attacks

[1] Corresponding author's e-mail: mufidahlawal26@gmail.com

targeting web applications pose significant risks to online transactions. These risks include the redirection of the user to malicious sites, illegal HTTP requests, theft of personal information through cookies and session, installation of malware and other illegal activities(Xenofontos et al., 2022). The consequences of successful cyberattacks can be severe, encompassing financial losses, reputational damage, and even legal repercussion(Bhakhri et al., 2024).

To counteract threats, the field of cybersecurity has been exploring proactive measures to identify and mitigate potential attacks before they strike(Kandasamy et al., 2022). This pursuit has led to the intersection of data science and cybersecurity, giving rise to predictive analysis techniques that harness the power of machine learning algorithms(Sarker et al., 2020). Predictive models can be trained on web traffic dataset to detect abnormal patterns of web requests as the dataset contains historical records of normal as well as malicious activities. Thus the machine learning algorithms learn to distinguish between normal request and potentially malicious ones. Thus, many types of cyberattacks including SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), phishing attempts, Malware, and Distributed Denial of Service (DDoS) can be accurately predicted and flagged in real time(Umar et al., 2023).

The core challenge in the domain of website attack prediction in lies in the identification of the most pertinent features that optimize accurate predictions while maintaining an acceptable level of computation overhead. Currently, the process of feature selection within the context of vulnerability prediction has not yet yielded an optimal solution. Selecting features too conservatively might lead to the exclusion of important predictors, diminishing the predictive capability of the model. Conversely, incorporating an excessive number of features can not only introduce noise and redundancy but also escalate the computational burden. This trade-off becomes increasingly pronounced as datasets grow in complexity and size (Deepa & Thilagam, 2016).

The problem of optimal feature selection has been explored by many studies in the literature. For instance, Viharos *et al.* (2021) (Viharos et al., 2021)focused on finding the best combination of feature selection methods to achieve optimal feature ordering. They proposed a hybrid approach that combines different methods for a more generalized solution. This approach aims to overcome the limitations of individual feature selection methods and provide a more robust and effective solution. Similarly, Li et al. (2021)(Li et al., 2021) presented a feature selection approach for network intrusion identification based on the Krill Herd algorithm. Dwivedi et al. (2021)(Dwivedi et al., 2021) proposed a swarm intelligence-based contribution for IDSs using the grasshopper algorithm. Most of these approaches are limited as they do not explore the feature space thoroughly to the fullest.

Due to the limitations of traditional feature selection approaches, numerous studies have explored the application of genetic algorithms (GAs) for feature selection. A genetic algorithm (GA) is a search heuristic inspired by the process of natural selection, used in computer science and optimization to find approximate solutions to complex problems. It operates on a population of potential solutions, evolving them over generations to improve their performance based on a defined fitness function(Alhijawi & Awajan, 2024; Vanneschi & Silva, 2023b). GA feature selection has the potential to reduce the noise and redundancy that may otherwise hinder predictive performance, leading to faster detection and more accurate identification of common web attacks such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). GAs significantly enhance the efficiency of cybersecurity measures by refining the feature space, ultimately resulting in models that are better equipped to detect anomalies in web requests and predict cyberattacks in real-time. This ability to fine-tune models using GAs directly contributes to more effective and proactive cybersecurity strategies for protecting web applications.

Many studies on feature selection techniques based on Genetic Algorithms have been proposed(Ahmad et al., 2011; Das et al., 2017; Dwivedi et al., 2021; Halim et al., 2021; Li et al., 2021; Stein et al., 2005; Viharos et al., 2021). For instance, Stein et al. (2005)(Stein et al., 2005) introduced a GA-based feature selection method using Decision Tree. The authors used a genetic algorithm as part of a feature selection process for improving network intrusion detection systems (IDSs). Experiments were conducted on the KDDCUP99 dataset, and the results demonstrated that the GA-based feature selection improved the performance of Decision Tree classifiers compared to using all available features.

Likewise, Ahmad et al. (2011)(Ahmad et al., 2011) presented a feature selection approach using GA in the context of Multi-Layer Perceptron (MLP). The method improved performance by reducing the number of features and increasing detection rates. These studies have consistently demonstrated the ability of GAs to enhance the accuracy of classifiers by selecting optimal feature subsets.

Most of the feature selection techniques that were proposed above are working quite well but the process of enhancing the techniques still continues. Extracting the useful features from the dataset can be done with better efficiency in regard with increasing detection accuracy(Halim et al., 2021). Consequently, Halim et al (2021)(Halim et al., 2021) proposed a new genetic algorithm-based method for feature selection. The method was applied to network security and intrusion detection to improve classifier accuracy. Part of the proposed method was the tuning of the GA parameters and using a novel fitness function which helps assign fitness values to individuals in the population, leading to the selection of optimal feature sets. The proposed approach recorded better performance than traditional feature selection methods on three benchmark datasets, reaching a maximum accuracy of 99.80%.

While previous studies, such as(Halim et al., 2021), have proposed genetic algorithm-based methods for feature selection to enhance classifier accuracy in network security, challenges remain regarding computational efficiency and the risk of overfitting. The computational complexity of evaluating large combinations of features and the reliance solely on accuracy as a performance metric limit the generalizability and practicality of these approaches for real-time applications. Moreover, accuracy alone does not sufficiently account for potential overfitting, particularly when models are applied to different datasets.

This paper aims to address these limitations by introducing a more comprehensive evaluation framework that incorporates the F1-score alongside accuracy to provide a clearer indication of model performance, specifically in terms of balancing precision and recall. Additionally, by applying the genetic algorithm-based feature selection technique to a dataset[2] containing real-world web attacks—SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF)—the study enhances the generalizability of the approach across multiple types of web vulnerabilities.

The rest of the papers is organized as follows: The next section (Section 2.0) elaborates on the methods employed in the study. Section 3.0 presents the results, followed by discussions. Finally, Section 4.0 concludes the paper.

The next section explains in details on the methods employed in the study.

2.0 Methodology

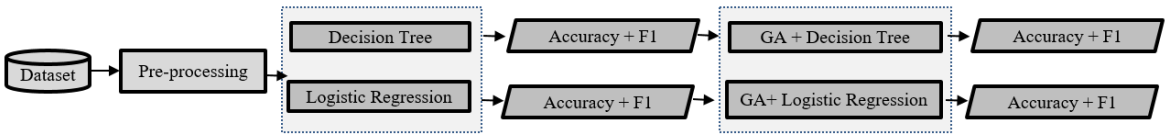

Figure 1: Overview of the methods used in the paper

Figure 1 depicts the overview of the methods used in the paper. Each of the major components of the section is explained in the following subsections:

3.1 Data pre-processing

The dataset used in this study is provided by the Canadian Institute of Cybersecurity(Sharafaldin et al., 2018). Specifically, Thursday Morning Web Attack dataset, was used because it focuses on web-based attacks, such as SQL injection, brute force attacks,

[2] https://www.unb.ca/cic/datasets/ids.html

and Cross-Site Scripting (XSS). The dataset contains 170, 366 records and 76 features related to network traffic flows. The target feature, Label, indicates the category or attack associated with the flow. Algorithm 1 was used in the pre-processing of the dataset. The pre-processing activity involved removing rows with values that were not valid numbers; Missing values were replaced with the mean value of the feature. Similarly, infinite values were replaced with maximum finite values that can be represented by a 64-bit floating-point number.

Algorithm 1: Data Pre-processing

1. IMPORT dataset from CSV file
2. HANDLE infinite and missing values in predictors
    - Replace infinite values with max float values
    - Use mean imputation to handle missing values
3. CONVERT imputed predictors to DataFrame

3.2 Machine learning models

As shown in Figure 1, two (2) machine learning modelling techniques were considered for this study: Decision Trees(Vanneschi & Silva, 2023a), and Logistic Regression(LaValley, 2008). These modelling techniques are elaborated in the following subsections:

3.2.1 Decision Tree

Decision Tree provides a transparent and interpretable approach to modelling complex relationships between variables. A Decision Tree predicts the output by splitting the dataset based on feature values to minimize impurity. In this study, the Gini Index, as presented in Equation (1) was used as the measure of impurity to determine the optimal split point for a node. Gini index is a measure of impurity or entropy and the feature importance is on how much it reduces the impurity across all the nodes of the trees in which it appears because it contributes more to the overall predictive power of the model. The Decision Tree recursively splits the feature space, choosing the feature and threshold that minimize the Gini Index at each node. The tree grows until a stopping criterion is met (e.g., max depth)(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.

3.2.2 Logistic Regression

Logistic Regression is a modelling technique used to predict the probability of an event occurring. It is particularly useful when the outcome variable is binary, meaning it can take only two values, such as 1 or 0. The core idea behind Logistic Regression is to model the

probability of an event occurring as a logistic function of a linear combination of predictor variables. Mathematically, this can be expressed as:

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

3.3 Evaluation Metrics

In this study, Accuracy and F1 scores were used as metrics to evaluate the performances of the models

$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$ X 100 .....(3)

$F1 = 2X\frac{Precision\ X\ Recall}{\text{Precision}+Recall}$..... (4) The Precision and the Recall are given in Equations (7) and (8) respectively

$\text{Precision} = \frac{True\ Positive\ Predictions}{True\ Positive\ Predictions+False\ Positive\ Predictions}$.....(5)

$Recall = \frac{True\ Positive\ Predictions}{True\ Positive\ Predictions+False\ Negative\ Predictions}$.....(6)

3.4 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 sSSolution 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:

i. **Initialization**: A population of individuals (potential solutions) is randomly generated, where each individual encodes a specific feature subset or solution.
ii. **Fitness Evaluation**: The fitness of each individual is assessed using a predefined evaluation function, such as accuracy or precision, depending on the problem.
iii. **Selection**: Based on fitness scores, individuals are selected as parents for reproduction. Higher fitness individuals are more likely to be selected.
iv. **Crossover and Mutation**: Parents undergo crossover (recombination) and mutation to produce offspring, introducing diversity into the population.

v. **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.

More specifically, 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):

$X_i = [X_{i1}, X_{i2}, \ldots, X_{in}]$ .....(7). X_i in Equation (7) is the i-th individual in the population, and each $x_{ij} \in \{0,1\}$ 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:

$f(X_i) = Performance\ (M(x_i))$---------- (8)

The Performance in Equation (8) 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 x_i and x_j 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:

$O1 = [x_{i1}, x_{i2}, \ldots, x_{ip}, x_{j(p+1)}, \ldots, x_{jn}]$.....(9)

$O2 = [x_{j1}, x_{j2}, \ldots, x_{jp}, x_{i(p+1)}, \ldots, x_{in}]$.....(10)

Equations 9 and 10 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:

$P(mutation\ of\ x_{ik}) = \mu$ ----------(11)

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:

$X_{best} = arg_{x_{i \in Population}} Max\ f(x_i)$ ----------(1)

Algorithm 2 presents the specific implementation of GA for feature selection in this study

Algorithm 2: 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

ISSN: 3027-2335

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 2. In Algorithm 2, 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.

4.0 Results and Discussion

The plot in Figure 1 illustrates the performance of a genetic algorithm (GA) in terms of F1 score and accuracy over multiple generations in Decision Tree. The x-axis represents the number of generations, while the y-axis represents the corresponding F1 score and accuracy values. In the Figure 1, Both F1 score and accuracy exhibit a general upward trend throughout the generations. This indicates that the GA is effectively optimizing the feature subset and improving the model's performance. Early generations exhibit fluctuations in F1 score and accuracy. This is expected as the GA explores the search space and gradually converges towards better solutions. The results suggest that the genetic algorithm is a promising approach for feature selection in website attack prediction. By iteratively optimizing the feature subset, the GA can significantly improve the model's performance in terms simultaneously both the accuracy and F1 score and living no chance to overfitting.

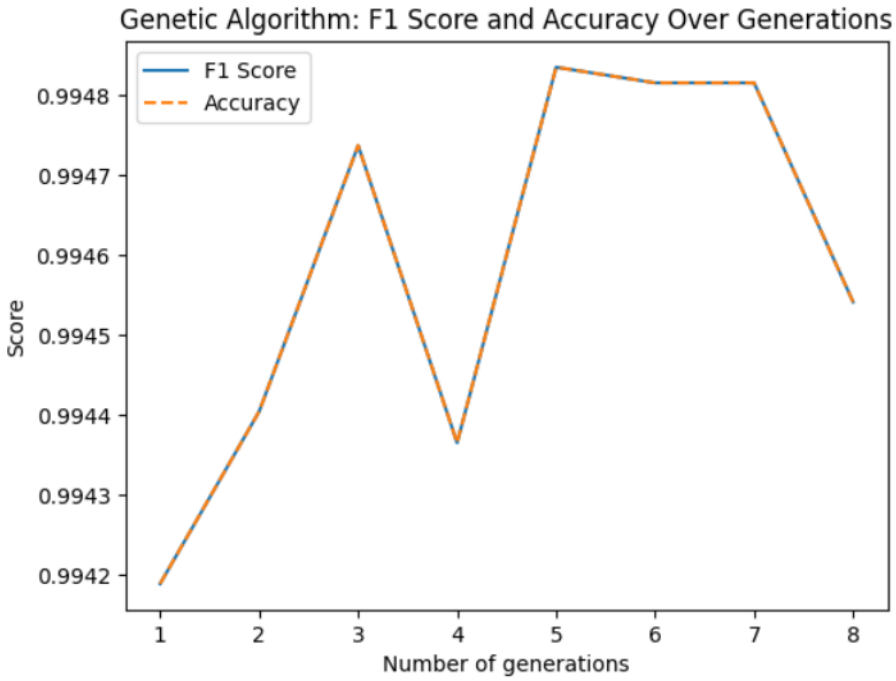

Figure 1: GA used on Decision Tree to select 10 out of 76 features representing 86.8% reduction in features

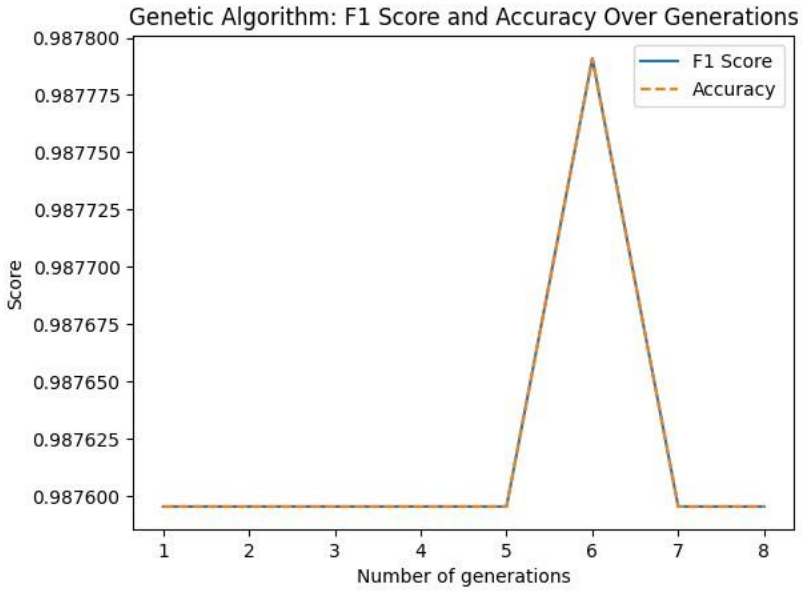

Figure 2: GA used on Logistic Regression to select 10 out of 76 features representing 86.8% reduction in features

The plot in Figure 2 also illustrates the performance of a genetic algorithm (GA) in terms of F1 score and accuracy over multiple generations in Logistic Regression. The x-axis represents

the number of generations, while the y-axis represents the corresponding F1 score and accuracy values.

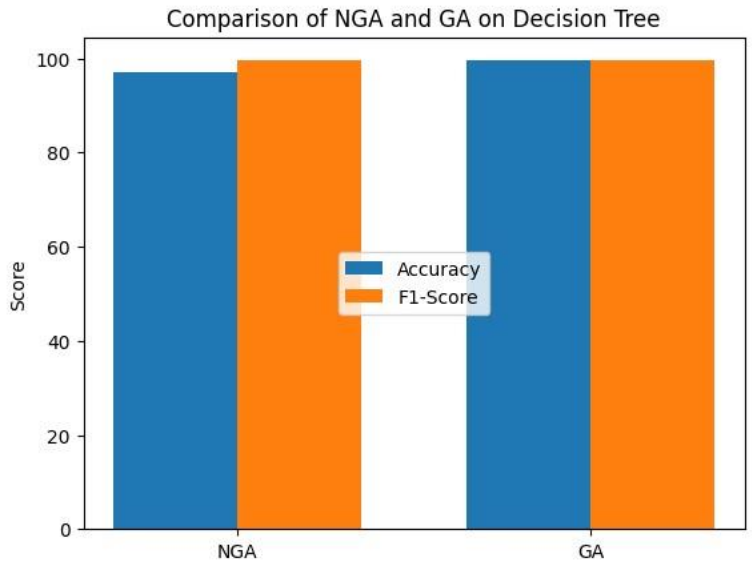

Figure 3: Accuracy and F1 scores of Non-GA and GA in Decision Tree

Figure 3 chart compares the performance of a Non-Genetic Algorithm (NGA) and a Genetic Algorithm (GA) when used for feature selection in a Decision Tree model. The x-axis represents the two algorithms, while the y-axis represents the evaluation metrics: accuracy and F1-score. The GA matches the NGA in terms both accuracy and outperforms NGA in terms of F1-score thereby suggesting a more robust and less over fitted model.

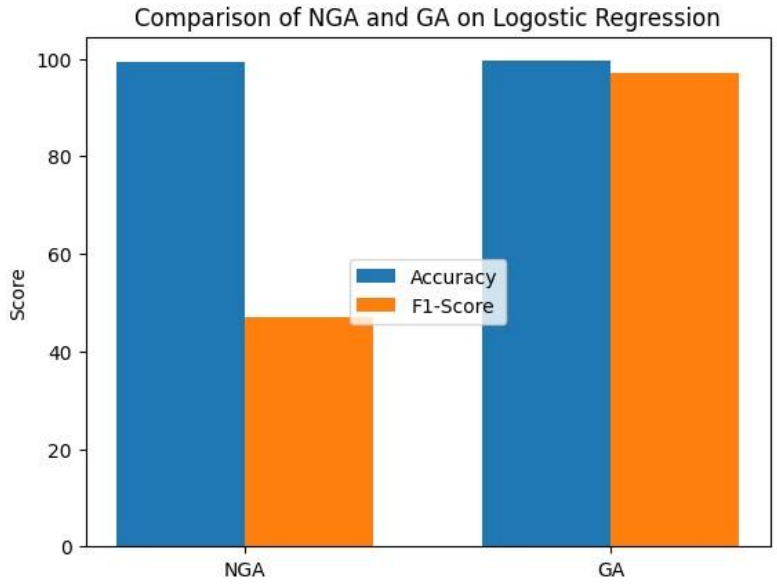

Figure 4: Accuracy and F1 scores of Non-GA and GA in Logistic Regression

ISSN: 3027-2335

The provided bar chart in Figure 4 compares the performance of a Non-Genetic Algorithm (NGA) and a Genetic Algorithm (GA) when used for feature selection in a Logistic Regression model. The x-axis represents the two algorithms, while the y-axis represents the evaluation metrics: accuracy and F1-score. As seen in the Figure the GA matches the performance NGA in terms of accuracy and significantly outperforms NGA in terms of F1-score**.** This suggests that the GA produces a more robust and less over fitted model.

The Genetic Algorithm (GA) for feature selection significantly reduced the number of features while maintaining or improving the models' performance. Initially, the dataset contained 76 features, but after applying GA, the number of selected features was reduced by up to **86.8%**. Despite this drastic reduction, the models did not experience any significant performance degradation, suggesting that the GA effectively eliminated redundant or non-informative features while retaining the most important ones.

A common issue in machine learning is overfitting, especially when models are trained on a large number of features. GA's ability to reduce feature dimensionality helps mitigate this by selecting only the most important features. This can be observed in the improved F1-scores for both models, which indicates a better balance between precision and recall, and a reduced risk of overfitting to noise in the data.

The application of the GA prevented overfitting in both models. In the Decision Tree, pruning through feature reduction led to a more generalizable model with better test-set performance. Similarly, Logistic Regression exhibited reduced variance, as evidenced by the higher F1-score post-GA selection.

The performance gains observed in this study align with the findings of prior research that demonstrated the effectiveness of GA for feature selection. Similar to prior studies(Halim et al., 2021; Stein et al., 2005), our results confirm that GA can enhance the accuracy of classifiers while reducing feature space dimensionality. However, this study extends the work by incorporating F1-score as a key evaluation metric, which ensures that the models selected by GA not only perform well in terms of accuracy but also handle imbalanced classes effectively.

5.0 Conclusion

This study explored the effectiveness of Genetic Algorithm (GA)-based feature selection in improving the performance of machine learning models for intrusion detection, using Decision Tree and Logistic Regression classifiers. The results demonstrated that GA significantly reduced the feature space—by as much as 86.8%—while simultaneously improving both accuracy and F1-scores for the models. In addition to the performance enhancements, the GA-based approach minimized the risk of overfitting, ensuring that the

models maintained robust predictive power across diverse data conditions. This reinforces the practical relevance of GA in scenarios where datasets contain high-dimensional feature spaces, but only a subset is truly informative for the task at hand. In conclusion, this research has successfully demonstrated that GA-based feature selection is a highly effective method for improving the performance of machine learning models in detecting intrusions, offering both accuracy and efficiency gains, making it a promising approach for deployment in security-critical environments.

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

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

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

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

Kandasamy, K., Srinivas, S., Achuthan, K., & Rangan, V. P. (2022). Digital Healthcare - Cyberattacks in Asian Organizations: An Analysis of Vulnerabilities, Risks, NIST Perspectives, and Recommendations. *IEEE Access*. https://doi.org/10.1109/ACCESS.2022.3145372

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

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