6TH International Conference and Workshop on Engineering and Technology Research (ICWETR) LAUFET 2025: 24-39

Intrusion Detection System with Feature Selection on Machine Learning Algorithm

[1]Ajagbe S. A., Akindolani A., Akande T. O., Adeyanju K. I., Aiyeniko O. and Ajiboye G. O.

[1]Department of Computer Engineering, Abiola Ajimobi Technical University, Nigeria,
[2]McAnderson Institute of Technology, United Kingdom.
[3]Department of Mechanical and Mechatronics Engineering, Abiola Ajimobi Technical University, Nigeria.
[4]Department of Computing, Sheffield Hallam University, United Kingdom.
[5]Department of Computer Science, Lagos State University, Ojo, Nigeria.
[6]Department of Computer Science, Precious Cornerstone University, Nigeria.

[1] saajagbe@pgschool.lautech.edu.ng, [2]dayo@mcandersoninstitute.tech,
[3]akande.timileyin@tech-u.edu.ng, [4]deyanjukorede@gmail.com, [5]olukayode.aiyeniko@lasu.edu.ng,
[6]ajiboyegrace@pcu.edu.ng

https://www.laujet.com/

Keywords:

Artificial Intelligence, Computer Networks, Genetic Algorithms, Internet of Things (IoT), Intrusion Detection (IDs), Machine Learning (ML), and Data Analysis

Corresponding Author:
saajagbe@pgschool.lautech.edu.ng

ABSTRACT

The selection of features plays a pivotal role in designing high-performance intrusion detection systems (IDSs), particularly in light of the growing complexity and dimensionality of network traffic data. This study introduces a machine learning-driven feature selection technique that utilizes an enhanced Genetic Algorithm (GA), referred to as GA-based Feature Selection (GbFS), to refine feature subsets for IDSs. The research implemented a fitness function alongside a parameter-tuning strategy to boost the efficiency of the GA., we employed three districting classifiers, namely Support Vector Machine (SVM), Random Forest (RF) cla ssifiers, and Logistic Regression (LR) for this task. Experiments were conducted on three well-known datasets— UNSW-NB15, CIRA-CIC-DOHBrw2020, and Bot-IoT—accessed from Kaggle to highlight significant gains in classifier accuracy. The comparative analysis reveals that GbFS achieves accuracy levels of up to 99.80%, surpassing traditional feature selection techniques. This research underscores the capacity of machine learning to increase the efficacy and efficiency of IDSs by overcoming challenges associated with feature selection in network security.

INTRODUCTION

In today's digital era, where billions of devices are interconnected, safeguarding sensitive data has become a critical priority. When identifying malicious activity, Intrusion Detection Systems (IDSs) are essential for defending networks against cyber threats such as ransomware, botnets, and malware. One of the primary challenges in developing efficient IDSs involves handling the high-dimensional nature of network traffic data, which can negatively impact detection accuracy and computational performance. Feature selection serves as an essential preprocessing step to identify the most relevant attributes, thereby improving the effectiveness of IDSs. The application of machine learning (ML) algorithms is highlighted in this paper for feature selection. Specifically, GA-based Feature Selection (GbFS), an improved Genetic Algorithm (GA)-based technique. By

incorporating an innovative fitness function and refined parameter optimization, this research aims to improve both detection efficiency and accuracy. The proposed approach is validated using widely accepted benchmark datasets, showcasing its potential to address feature selection challenges and bolster IDS performance (Bakro *et al.,* 2024).

Identifying the type of cyber-attack and the vulnerabilities it exploits is a significant challenge in cybersecurity. As security measures become more robust, cybercriminals adopt increasingly sophisticated and creative ways to find their victims. According to Bakro *et al*. (2023), these techniques include embedding assaults in a variety of file types, including Word documents, PDFs, and pictures, as well as more severe strategies like injecting malicious code into devices or deploying ransomware with worm-like behavior capable of infecting hundreds of devices. These types of attack techniques intensify the complexity of detecting cyberattacks. To address these challenges, the emerging field of Cyber Threat Intelligence (CTI) emphasizes the adoption of ML-based security tools to combat cyber threats effectively.

In recent years, significant progress has been made in applying ML in cybersecurity, particularly for CTI purposes (Zouhri *et al.,* 2024). Awotunde *et al.* (2021) opined that ML is now extensively utilized to improve IDS due to its demonstrated effectiveness in malware detection and classification (Jupriyadi *et al.,* 2024). Implementing ML algorithms requires training models capable of differentiating between malicious and benign traffic, and high-quality data is essential for this purpose.

The lack of high-quality data sources significantly impedes the ability of ML algorithms to attain peak performance. To rectify this issue, the current research utilizes a trio of extensively utilized datasets accessed from Kaggle for identifying intrusive activities, which assesses the efficacy of the proposed approach, which incorporates three classification algorithms to optimize the performance of the ML algorithms and identify the effective feature selection. To confirm the performance and efficacy of the suggested solution, it is also compared to three closely similar cutting-edge methodologies (Mhawi *et al.,* 2022).

Managing high-dimensional data presents a significant challenge in the design of IDS. Noise, irrelevant characteristics, and the curse of dimensionality combined within datasets can lead to reduced accuracy and heightened computational demands in ML models (Kareem *et al*., 2022). Feature selection offers a solution by identifying the most informative subsets of features, thereby decreasing complexity and enhancing the effectiveness of IDSs. When supervised learning is involved, feature selection focuses on isolating features that optimize a classifier's ability to differentiate between various classes. According to Li *et al.* 2024, this study introduced an ML-based feature selection approach utilizing an enhanced Genetic Algorithm (GA), GA-based Feature Selection (GbFS), to tackle these issues. This research demonstrates how feature selection can significantly boost IDS accuracy while minimizing false positive rates through the model of a novel fitness function and refined parameter-tuning strategies. The proposed method is validated on benchmark datasets, highlighting its capability to streamline and improve the intrusion detection process.

The creation of efficient IDS necessitates identifying optimal feature subsets from large, high-dimensional datasets. High dimensionality can introduce irrelevant and redundant features, which may degrade model performance, raise false positive and false negative rates, and inflate computational costs. Feature selection techniques, such as wrapper, filter, and embedded techniques, address these issues but come with limitations.

Wrapper methods are effective at identifying relevant features but involve iterative evaluations, leading to high computational overhead (Almomani 2020; Jaw and Wang 2021). Filter methods are computationally efficient but may yield suboptimal feature subsets because they lack feedback from the learning algorithm. Embedded methods attempt to balance these drawbacks but often struggle to scale effectively as data complexity increases. This study presents an ML-based feature selection approach utilizing GA, incorporating an innovative fitness function and parameter-tuning framework to enhance feature selection for IDSs. By focusing on benchmark datasets, this research demonstrates how the proposed approach enhances classification accuracy, reduces computational costs, and strengthens IDS resilience against emerging cyber threats (Yin *et al.,* 2023; Aljabri *et al*, 2024).

This research implemented an ML-driven feature selection framework aimed at improving the performance and precision of IDS. The key contributions and unique aspects of this work include:

i. The GA-based feature selection method was employed alongside an evolutionary computing approach to select relevant features for intrusion detection, utilizing a custom fitness function to optimize feature identification.
ii. Optimization and validation of GA parameters on benchmark datasets (UNSW-NB15, CIRA-CIC-DOHBrw2020, and Bot-IoT), resulting in improved detection accuracy and reduced computational burden.
iii. To underscore how the proposed classification methods outperform state-of-the-art techniques and improve the efficiency of ML classifiers, increasing actualization rates and reducing false positives and false negatives.

REVIEW OF RELATED WORK

Akhiat et al. (2024) proposed a feature selection method combining GA, Principal Component Analysis (PCA), and Multilayer Perceptron (MLP). The goal of the study was to improve the actualization and detection rate of classifiers in identifying intrusions. Their approach was compared against the use of standalone PCA. Employing GA, they identified the principal feature space that optimally enhanced the classifier's sensitivity. Three experiments using feature subsets of 12, 20, and 27 features were used to assess the suggested strategy; the 12-feature subset yields the highest accuracy of 99%. Similarly, GA is used as a feature selection method by Sindhu et al. (2012). They started their optimization process by creating a random population, then they assessed the chromosomes' fitness to create the next population. The fitness function evaluates the quality of the feature subsets by taking into account feature count, sensitivity, and specificity. Using 16 features that were taken from the KDD dataset's initial 41 features, they attained an accuracy of 98.38% when compared to other feature selections (Sayegh et al., 2024).

The research conducted by Barhoush et al. (2023) presented an IDS that utilized a GA for feature selection, incorporating a sophisticated fitness function. The method produced enhanced accuracy while sustaining a minimal false positive rate. The methodology was evaluated utilizing the KDD Cup and UNSW-NB15 datasets, with detailed data for each class provided in their analysis. Additionally, they provide unique statistics for each category and utilize their framework to illustrate its efficacy. Similarly, Aljabri et al. (2024a) developed a feature selection model based on an optimized binary Whale Optimization Algorithm for detecting network intrusions. Their methodology, assessed using the KDD dataset, addresses some difficulties but faces shortcomings, including sluggish convergence and susceptibility to local optima during the update process, which negatively

impacts classification performance. Compared to GA, their method attains superior accuracy, identifying 5 out of 41 features with an accuracy rate of 97.89%, while GA identifies 11 out of 41 features with a lower accuracy of 95.58%. Meanwhile, Amaouche et al. (2024) explore the application of autoencoders as generative models for feature learning in a separate study. Furthermore, they illustrate how autoencoders may effectively learn latent representations and semantic commonalities among dataset features. Their methodology was employed for tasks such as malware classification and intrusion detection. They utilize the Microsoft Malware Classification Challenge (BIG 2015) dataset and the KDD Cup dataset. The Gaussian Naïve Bayes classifier achieves the highest accuracy of 83.3% in intrusion detection. The main points of this earlier research are listed in Table 1, along with the current suggestion (Al-Yaseen et al., 2022).

Their method was based on a popular swarm intelligence methodology called the Krill Herd (KH) algorithm. In their solution, the condition of the krill herd in the search space is updated using a linear closest neighbor lasso step optimization. A globally optimal solution can be found easily through this procedure. Likewise, another swarm intelligence-based method for IDS is the work of Turukmane and Devendiran 2024. Their approach combines an ensemble feature selection method with the grasshopper algorithm, which was also based on swarm intelligence.

Table 1. Key component of the current contribution and earlier works.

Reference	**Model**	**Feature film**	**Result**
Umar et al. (2024)	Decision tree (DT)-GA	34	Detection error rate of 0.095 for U2R and 19.9 for R2L
Yaras and Dener (2024)	Pareto-Optimal Fuzzy Rule Classifier-GA	28	Accuracy of 99.24%
Awad and Fraihat (2023)	MLP-PCA + GA	16	Accuracy of 99%
Das et al. (2022)	DT- GA	18	Accuracy of 98.38%
Zouhri, Idri, and Ratnani (2024)	MLP-GA-Based Kernel Feature Extraction	14	Accuracy 94.22%

To protect a company's network from outside threats, intrusion detection systems, or IDS, are essential. Machine learning is often used to create IDS because of its exceptional performance. However, the detection accuracy of ML models may be adversely affected by the high dimensionality of network traffic data. To mitigate the effects of dimensionality, it is essential to reduce the number of features while retaining critical information, ensuring that the reduced feature set enhances the classifier's detection accuracy. While numerous feature selection techniques have been proposed and are performing effectively, continuous efforts are being made to improve these methods (Fang *et al.,* 2024). Efficient extraction of significant features from datasets, particularly to boost the detection accuracy, can be achieved using unsupervised approaches. However, unsupervised feature selection techniques often involve computational complexity due to the extensive number of feature combinations that must be evaluated to identify the optimal subset. This study focuses on developing an enhanced unsupervised feature

selection technique utilizing Evolutionary Algorithms (EAs) to achieve better detection accuracy, distinguished by low False Positive (FP) and high True Positive (TP) rates (Li *et al.,* 2020).

Evolutionary feature selection for intrusion detection

This study's methodology centres on the implementation of a GA-based Feature Selection (GbFS) approach aimed at improving IDS. The process is structured into four phases, viz-a-viz Data Generation and preparation, GA-Based Feature Selection Module, Classification Module and finally Outcome Evaluation and Comparison. Figure 1 depicts the framework's proposed solution.

1. **Dataset Generation and Preparation**
   i. Standardized datasets for network traffic analysis, including UNSW-NB15, CIRA-CIC-DOHBrw2020, and Bot-IoT, undergo a preprocessing phase to address issues related to incomplete data, disproportionate class distributions, and extraneous noise.
   ii. To guarantee the uniformity and seamless operation of the GA, the characteristics are standardized to prevent any discrepancies that may hinder its performance.

2. **GA-Based Feature Selection Module**

   i. Initialization phase: A collection of feature subset combinations is randomly created, with each combination serving as a potential solution candidate.
   ii. Fitness Assessment: A newly introduced evaluation metric is employed to assess the suitability of feature subsets, taking into account their relevance, redundancy, and impact on the precision of classification outcomes.
   iii. Iterative Refining: The GA employs a sequence of operations, including Selection, Crossover, and Mutation, to progressively refine feature subsets and steer the search towards optimal solutions.
   iv. Optimization of Hyperparameters: The GA performance is fine-tuned by adjusting key parameters, including population size, mutation probability, and crossover probability, to strike a balance between the speed of convergence and the quality of the solution.

3. **Classification Module**

   i. the subsets of features that are carefully chosen by the Genetic Algorithm module are subsequently utilized to educate and fine-tune various ML classification models, including but not limited to SVM, RF, and LR algorithms.
   ii. Evaluative measures such as correctness, exactness, sensitivity, F1-score, and false alarm rate are employed to gauge the performance of classifiers.

4. **Outcome Evaluation and Comparison**

   i. Modern feature selection methods are contrasted with the suggested GbFS approach (Information Gain, PCA) and other optimization algorithms (PSO, ACO).
   ii. Metrics like detection accuracy, computational efficiency, and convergence rate are used for evaluation.

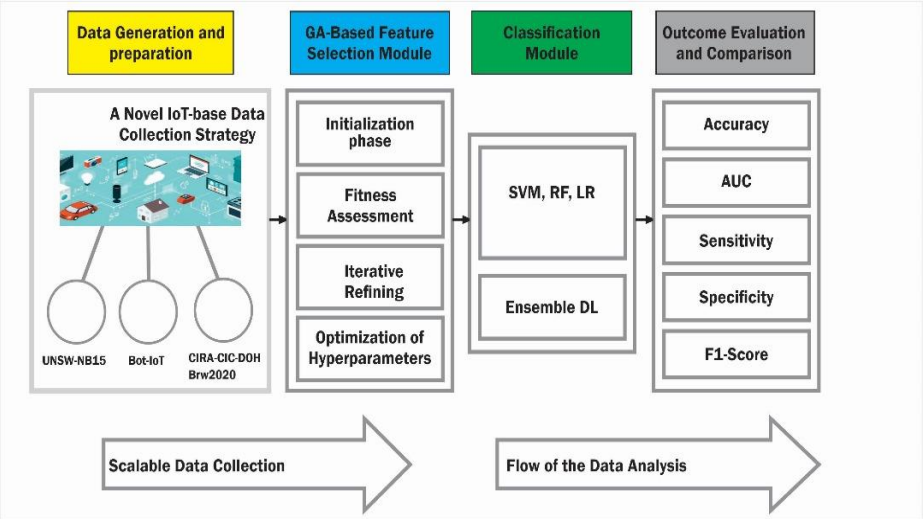

Figure. 1: Block Diagram of the proposed system.

Datasets and preparation

Experiments are performed using three standard intrusion detection datasets, which are integrated into the proposed GA-based framework. These datasets include UNSW-NB15, CIRA-CIC-DOHBrw2020, and Bot-IoT. A summary of these datasets is provided, with further details explained in Table 2. The CIRA-CIC-DoHBrw-2020 dataset includes both harmless and malicious DoH traffic, as well as non-DoH traffic. It combines HTTPS and DoH traffic to develop a comprehensive dataset. This is accomplished by employing browsers and DNS-tunnelling apps that support the DoH protocol for browsers to access the top 10,000 Alexa-ranked websites. There are over 1.4 million samples in the dataset, which includes 34 characteristics and four different classes. To reduce dimensionality, small packets that do not convey enough data are removed during data collection. At UNSW Canberra Cyber, the Bot-IoT dataset is created in a realistic network environment with both typical and botnet traffic. This dataset contains around 72 million records, covering attack classes such as DoS, DDoS, OS and service scans, and data exfiltration (Nazir and Khan 2021).

Table 2: Datasets Summary

Features	**CIRA-CICDoHBrw2020**	**Bot-IoT**	**UNSW NB-15**
Feature size	35	25	47
Class size	6	8	10
Data points	~1.6 million	~3 million	~0.35 million

The UNSW NB-15 dataset was produced using the IXIA Perfect Storm program from the Australian Center for Cyber Security. It is founded on both simulated attack behaviors and actual, everyday network activity. There are nine types of attacks in the dataset: worms, reconnaissance, shellcode, backdoors, DoS, exploits, fuzzers, and analysis (Rani *et al.*, 2024). The dataset comprises 49 characteristics that represent data from several classes. Because of the constraints of the present benchmarks, the maximum number of features examined does not exceed 50, as shown in Table 3, which is a list of datasets currently being considered for testing. The suggested method can still categorize data with more dimensions. Typically, increasing the number of features or input variables affects the prediction outcome negatively, but having too few features also impedes effective learning. In this

study, classifiers were trained with the complete set of features and a reduced set to compare performance. The outcomes demonstrate that a reduced number of features yields better performance. While a robust feature set can lead to classifier overfitting, an insufficient number of features can result in underfitting. Based on this, various feature sets were evaluated, with the optimal outcome identified using just 10 features. The best results were achieved with just 10 features, despite using datasets with up to 50 features.

Table 3: Parameter Settings

Variable	Values
Aggregate count of properties	52, 41, 68
Number of instances	100
Genetic sequence length	10
Rate of genetic exchange	0.5
Rate of genetic variation	0.6
Maximum node depth	5
Optimization step size	0.1
Quantity of estimators	20
Stimulator	Gradient Boosted DT
SVM core function	Unidirectional
k parameter	9
Objective	multi:softmax

The initial stage of any learning system is data preparation. The system introduced here captures incoming data and stores it in its data repository. A crucial part of this process is data pre-processing, which ensures the data is cleansed of noise and inconsistencies before being fed into the learning module. Normalization, scaling, and input-output coding are examples of pre-processing procedures. It is important to note that some data may not be in a format compatible with the learning module, as the current solution only accepts numerical data and generates output within a range that the fitness function specifies. It's also possible that the input values for various data attributes don't fall on the same scale. Scaling is used to solve this problem. All attributes are scaled in this study. Using the formula in Equation (1)

$$X_i = \frac{(X_i - X_{min})}{(X_{max} - X_{min})} \tag{1}$$

where xi, xi, Xmax, and Xmin stand for the scaled value, maximum, minimum, and original values, respectively. Equation (1) is used to scale the properties first, and then the data is normalized. To keep all of the input values within a reasonable range for the learning module to process, normalization is done. Equation (2) lists the normalization formula.

$$X_i = X_{max} - X_{min} \times \left[\left[\frac{(X_i - X_{min})}{(X_{max} - X_{min})}\right]\right] + X_{min} \tag{2}$$

To change alphanumeric data into numeric data, the 1-of-n encoding algorithm is used. Simulations are conducted utilizing benchmark intrusion detection datasets, such as Bot-IoT, CIRA-CIC- DOHBrw- 2020, and UNSWNB15, to evaluate the efficacy of the suggested framework. The results and evaluation section contains thorough explanations of various datasets. Both textual and numerical data are included in these three databases. A label encoder converts the dataset's category features into a format compatible with ML.

Adaptive GA module

The proposed framework's second module is the Genetic Algorithm-based feature selection component, which plays a crucial role in the architecture. This methodology uses a Genetic Algorithm carried out in an unsupervised manner. The Genetic Algorithm follows a sequential protocol: (a) initializing the population, (b) formulating the fitness function, (c) selecting parents for the next generation, (d) executing crossover, (e) developing mutation, and (f) evaluating the final next generation. This process enables the framework to identify the most relevant features, enhancing the overall system's accuracy and efficacy.

Genetic configuration and founding population

The GA process begins with the creation of an initial population, where chromosomes are randomly formed by selecting unique genes from the dataset attributes. Each chromosome, a one-dimensional array, represents a potential solution to the feature selection problem. The genes are chosen randomly, without duplicates, and based on a predetermined number. This stage sets the foundation for the GA, which shows the chromosome structure. In this study, N is set to 10, though additional experiments are conducted with varying values for N. Every chromosome cell has a numerical value, V, where V is more than or equal to 0 and the length of the feature set is greater than or equal to V. Each gene's numeric value correlates to a certain feature number. As a result, ten features chosen from the feature set are present in each chromosome, creating a subset that will subsequently be optimized to find the best answer.

Fitness function

This study introduces a new fitness function that assesses the relationship between selected features, addressing the challenge of absent class labels. A method for identifying traits that show little resemblance to one another is the fitness function. This makes it possible to choose features that exhibit more diversity and can record a wider variety of data relevant to the dataset. The suggested fitness function calculates the average correlation after determining the correlation between the features that have been provided. Once the resulting value of the average correlation is obtained, optimization is necessary to enhance both the accuracy and average correlation values across the GA generations. When the selected features are varied and show little association, a solution is needed. This is accomplished by converting the average correlation value into an average non-correlation value as presented in Equation 3. To do this, the average correlation for each generation is subtracted from 1, the highest value that may be achieved according to Equation 4. This conversion makes it possible to display the accuracy and correlation averages in ascending or descending order. The accuracy of the particular chromosome and the transformed average correlation value are then used to compute the fitness values. The converted values and accuracy are averaged to establish the chromosome's fitness. The main goal of the suggested strategy is to maximize the accuracy and average correlation. The fitness function is shown in Equation 5.

$$Coor_{avg} = \frac{Sum\ (s)\ of\ values\ above\ the\ diagonal}{Numbers\ of\ Values} \quad (3)$$

$$Corr_{avg}^{t} = (1 - Coor_{avg}) \quad (4)$$

$$F_i = \frac{A_i + (1 - M_i)}{2} \quad (5)$$

The fitness value of the ith chromosome is determined, which involves the average correlation value, transformed average uncorrelation, accuracy, and computed correlation matrix. The fitness function employed in this approach consists of two components: the objective function and the scaling function. The objective function, which is optimized in this study, focuses on accuracy. The optimization is attained through a scaling function, namely the uncorrelation function, that refines the accuracy performance.

Classification

Classification is the last phase in the suggested solution. Following the random selection of genes (features) to create each chromosome, a fresh dataset is made using just the selected features before classification. Only the features that correspond to the optimum chromosome for a particular dataset are kept once the Genetic Algorithm (GA) converges. The dataset is then split into training and testing sets using k-fold cross-validation. Both the training and testing datasets for every chromosome are used for classification. Three classifiers are used in this study: XgBoost, k-NN, and SVM.

Support Vector Machines (SVMs): SVMs are a supervised learning approach used for classification, effectively handling both linear and non-linear data in high-dimensional spaces by creating decision boundaries. Its advantages include strong performance in high-dimensional spaces, flexibility with different kernel functions (including custom kernels), and memory efficiency. Overfitting can occur when features outnumber samples.

K-Nearest Neighbor (k-NN): k-NN is a versatile approach that can be used for both regression and classification, making predictions based on the majority vote of the nearest neighbors. In the simplest case, where k = 1, the class is determined by the closest data point. The optimal k-value can be identified either through data inspection or experimentation. A higher value of k tends to be more beneficial as it reduces noise. The classification decision is based on the distance between points, with similar distance metrics including Euclidean, Manhattan, and Minkowski, chosen based on the data type. XgBoost: XgBoost is an advanced algorithm capable of handling various data irregularities. It offers features such as regularization, parallel processing, increased flexibility in optimization methods, user-defined assessment standards, and the capacity to manage missing values. XgBoost supports parallel tree boosting, allowing for efficient and accurate solutions to a wide range of ML problems.

EXPERIMENTATION OUTCOMES

The proposed approach is evaluated using three benchmark datasets and compared to four advanced feature selection methods, then presents the outcome is presented here. Additionally, three state-of-the-art techniques are also compared. The prediction accuracy of classifiers using proposed and conventional feature selection methods is reported. The solution was implemented in MATLAB, with some built-in libraries used for file processing and metric computation. The development environment used was Spyder, with some code implemented in Python 3.6. Simulations were run on a computer with an Intel Core i5 CPU, Windows 11, and 8GB of RAM.

Parameter settings

In the second phase of this research, feature selection is performed using the GA, which involves experimenting with different parameter configurations. The varied parameters include the number of iterations, the population size, the length of the genetic sequence, the evaluation metric, and the rates of genetic recombination and mutation. Crossover is a critical parameter for enhancing the GA's ability to converge and identify the optimal solution. Two distinct crossover probabilities are tested: uniform crossover in the first experiment, which yields

the best results, and two-point crossover in the second experiment. Half of the bits are consumed during the crossover phase since the crossover rate is set at 0.5. Together with crossover, mutation serves as the GA's divergence operator, directing the generations toward better convergence in the search for the best answer. Although not all new individuals in the studies experience mutation, the mutation rate is 0.5. The specific parameter settings used in the simulations are listed in Table 3. After processing the input data, the suggested technique, GbFS, produces a subset of ideal features that improve categorization. The performance is then evaluated by running standard classifiers with the characteristics that GbFS has chosen. To assess the effectiveness of GbFS, several experiments are conducted, as outlined in the following sections.

Full-feature experiments

The suggested method was applied to every benchmark dataset, using all its attributes, in an experiment. The three stages of the suggested framework, GbFS, are (a) dataset preparation, (b) GA-based learning module, and (c) classification, as was previously said. To assess classifier accuracy utilizing the entire dataset features, the second phase (b) of the framework was removed for this experiment. In this work, a training set was used for all classification tasks, and a different, untested test set was used for evaluation.

Table 4: Threat types in CIRA-CIC-DoHBrw-2020.

Threat classification	Number of entries	Threat name
Encrypted DNS	296643	HTTP-based DNS protocol
No HTTPS-based DNS	879493	No encrypted DNS
Valid DoH	18907	Valid HTTPS DNS
Virulent	295836	Virulent

For intrusion detection in the CIRA- CIC-DOHBrw- 2020 dataset, three classification algorithms were utilized: SVM, k-NN, and XgBoost. The dataset's 34 features and attack type frequencies are described in Table 4, with classification results reported in Table 5. The results of XGBoost applied to specific classes from the CIRA- CIC-DOHBrw-2 020 data are shown in Figure. 2.

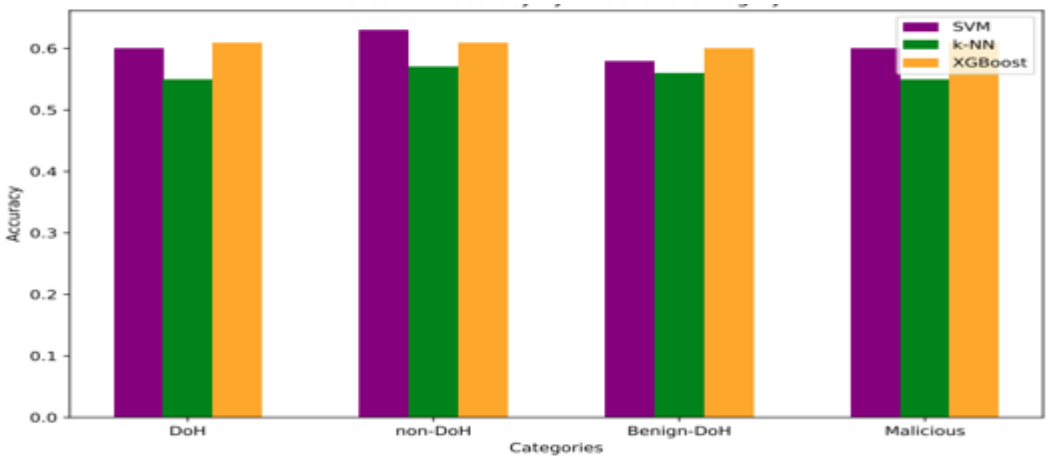

Figure 2: Results achieved using CIRA-CIC-DoHBrw, 2020 dataset.

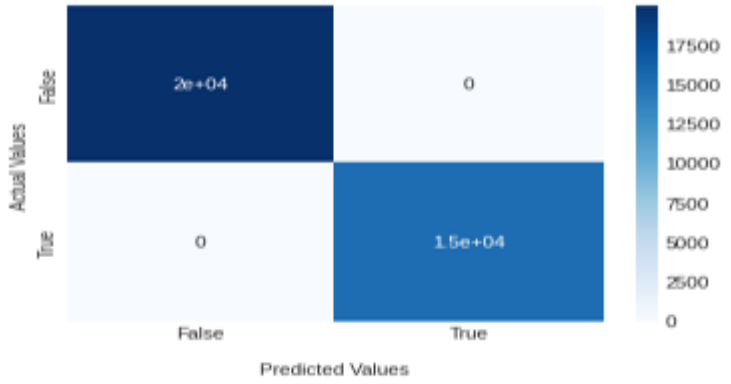

Figure 3: Confusion matrix of seaborn, where the actual value was plotted against the predicted value

According to the experiment results, the non-DoH class uses the three classifiers to obtain detection accuracy ranging from 55% to 63%. For different classes, SVM offers the best detection accuracy among them. XGBoost performs better than k-NN and SVM, with an accuracy of 61%. Figure. 3 shows how PCA reduces the dataset's dimensionality, resulting in 46 features from 190,000 samples.

This study computes the detection accuracies for each class of the 29 features in the Bot-IoT dataset, as shown in Table 6. The confusion matrix results from using SVM, k-NN, and XGBoost as the three classifiers on the Bot-IoT dataset. According to the experimental results, the SVM classifier detects the DDOS class with mediocre performance but reaches the maximum accuracy for the Theft class. For the k-NN classifier, the results for two classes—reconnaissance and theft show significant detection accuracy, as illustrated in Figures 4 and 5. The resulting 46 feature columns with 190,000 samples enhance the work. For k-NN, the DDOS and normal classes achieve detection accuracies of 87% and 83%, respectively. When used in the Bot-IoT dataset for intrusion detection, the XGBoost classifier outperforms the other classifiers. The findings demonstrate that XGBoost produces the best detection results, with the Normal class displaying the lowest accuracy at 72.4% and the Theft class attaining the highest accuracy at 98.8%.

The UNSW NB-15 dataset includes various threat classes, Figure 6 presents the results of the experiment conducted on this dataset, the experiment utilizing the k-NN classifier with nine neighbours. The generic assault class uses the SVM classifier to attain an accuracy of 98.1% out of the ten classes in the UNSW NB-15 dataset. Two groups, however, exhibit detection accuracies that are nearly 0%. Figure 7 depicts the observed seaborn confusion matrix for the dataset. Like SVM, k-NN achieves the highest accuracy of 98.3% for the generic threat class. The generic class continues to produce the greatest detection accuracy (96.8%) when using the XGBoost classifier. This result is supported (Akinlade *et al*., 2023; Oyekunle *et al*., 2025; Benedit, 2023; Ugbomeh *et al*., 2024; Ajagbe *et al*., 2024).

CONCLUSION AND FUTURE DIRECTION

IDS performance depends on selecting key features from network traffic data. This study proposes a GA-based method to reduce dimensionality, optimizing feature subsets with a novel fitness function to boost classifier accuracy.

Table 5: The CIRA-CIC-DOHBrw 2020 data yielded a confusion matrix for the three classifiers.

		Encrypted DNS			**No HTTPS-based DNS**			**Valid DoH**			**Virulent**		
		SVM	**KNN**	**XgBoost**	**SVM**	**KNN**	**XgBoost**	**SVM**	**KNN**	**XgBoost**	**SVM**	**KNN**	**XgBoost**
DoH	**SVM**	175571			22147			23686			65539		
	KNN		194632			121239			5268			12504	
	XgBoost			163719			101894			6598			13832
non-DoH	**SVM**	175852			535496			121478			96567		
	KNN		186243			439654			102010			151586	
	XgBoost			10107			596173			174852			196461
Benign-DoH	**SVM**	1247			4071			13221			4368		
	KNN		1578			6574			12053			1502	
	XgBoost			3650			132			12247			5087
Malicious	**SVM**	10247			14578			15298			290713		
	KNN		39799			22445			10714			108578	
	XgBoost			14022			15671			37689			156854

Table 6: Attack classes in the Bot-IoT dataset

Classification of Threat/Attack	**Number of entries**	**Attack name**
DoS	224,785	DoS
DDoS	204,000	DDos
Theft	160	Unauthorized data theft and keystroke logging
Reconnaissance	128,165	Service and Operating System Detection

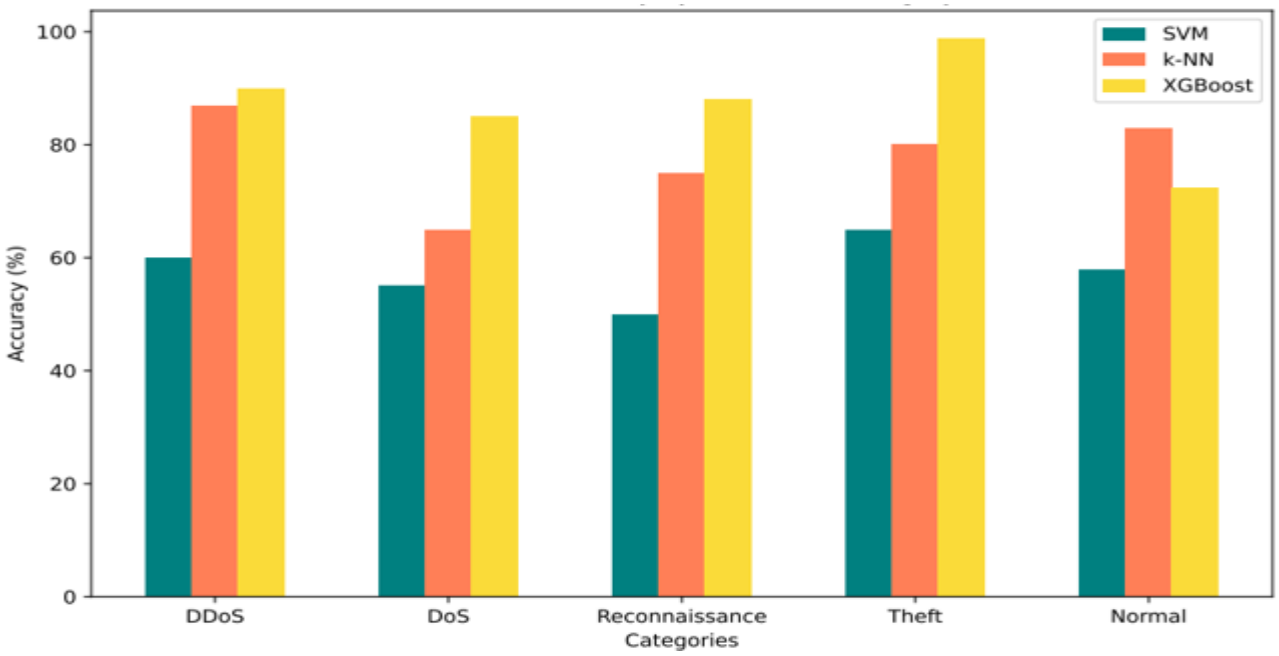

Figure 4: Results achieved using the Bot-IoT dataset.

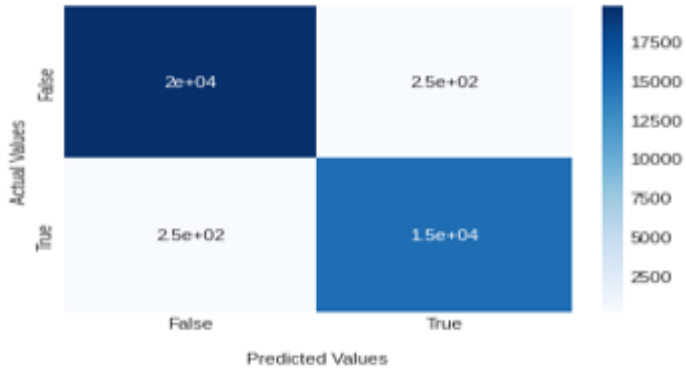

Figure. 5: Confusion matrix of seaborn, where the actual value was plotted against the predicted value

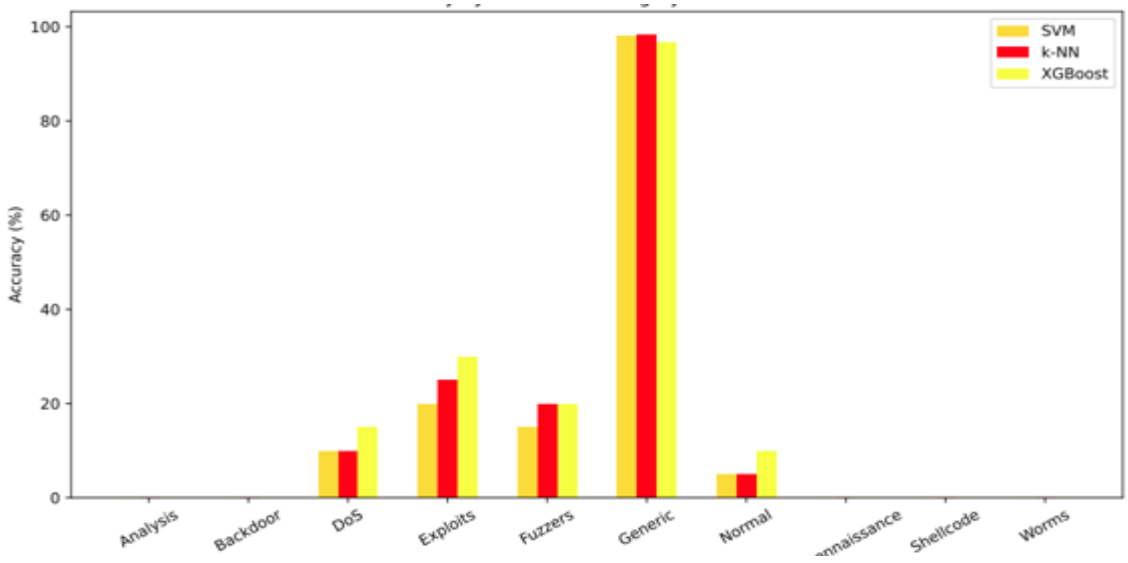

Figure 6 – Results were obtained using the UNSW NB-15 dataset

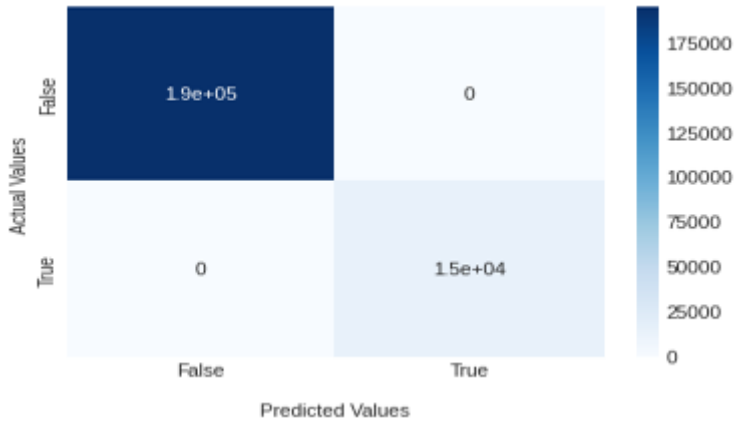

Figure 7 – Observed Seaborn Confusion Matrix for the dataset.

Experimental results on benchmark datasets—UNSW-NB15, CIRA-CIC-DOHBrw- 2020, and Bot-IoT—showed that the GbFS approach outperformed traditional feature selection techniques, achieving better detection rates and fewer false positives. Despite the success of the proposed method, depending on GA, which can be computationally expensive, there are some drawbacks. Future research could investigate faster optimization strategies, such as binary chaotic genetic algorithms or (1 + 1)-Evolutionary Strategies, to speed up convergence. Moreover, while this study employed supervised learning for classification, incorporating unsupervised techniques like clustering could further improve the adaptability of IDSs to detect emerging attack patterns. This work represents a significant advancement in using ML and optimization techniques to enhance IDS performance in managing complex, high-dimensional data.

REFERENCES

Aljabri, A., Jemili, F. and Korbaa, O. (2024) Convolutional neural network for intrusion detection using blockchain technology, International Journal of Computers and Applications, 46:2, 67-77, DOI: 10.1080/1206212X.2023.2284443

Aljabri, A., Jemili, F. and Kobara, O. (2024a). Real-Time Data Fusion for Intrusion Detection in Industrial Control Systems Based on Cloud Computing and Big Data Techniques. *Cluster Computing,* 27(2): 2217–2238.

Akhiat, Y., Kaouthar T., Ahmed Z. and Mohamed C. (2024). IDS-EFS: Ensemble Feature Selection-Based Method for Intrusion Detection System. *Multimedia Tools and Applications*, 83(5): 12917–37.

Al-Yaseen, W. L., Ali, K. I. and Faezah, H. A. (2022). Wrapper Feature Selection Method Based on Differential Evolution and Extreme Learning Machine for Intrusion Detection System. *Pattern Recognition*, 132: 108912.

Almomani, A. and Omar, A. (2020). A Feature Selection Model for Network Intrusion Detection System Based on Pso, Gwo, Ffa and Ga Algorithms. *Symmetry,* 12(6): 1–20.

Amaouche, S., Azidine, G., Said, B., and Mourade, A. (2024). IDS-XGbFS: A Smart Intrusion Detection System Using XGboost with Recent Feature Selection for VANET Safety. *Cluster Computing,* 27(3): 3521–35.

Awad, M. and Salam, F. (2023). Recursive Feature Elimination with Cross-Validation with Decision Tree: Feature Selection Method for Machine Learning-Based Intrusion Detection Systems. *Journal of Sensor and Actuator Networks*, 12(5). https://doi.org/10.3390/jsan12050067.

Awotunde, J. B., Chinmay, C. and Abidemi, E. A. (2021). Intrusion Detection in Industrial Internet of Things Network Based on Deep Learning Model with Rule-Based Feature Selection. *Wireless Communications and Mobile Computing*

Bakro, M., Rakesh, R.K., Amerah, A., Zubair, A., Md Nadeem A., Mohammad S. and Ahmed, A. (2023). An Improved Design for a Cloud Intrusion Detection System Using Hybrid Features Selection Approach with ML Classifier. *IEEE Access* 11, 64228–47.

Bakro, M., Rakesh, R. K., Mohammad, H., Zubair, A., Arshad, A., Syed, I, Y., Mohammad N. A. and Nikhat P. (2024). Building a Cloud-IDS by Hybrid Bio-Inspired Feature Selection Algorithms Along with Random Forest Model. *IEEE Access*, 12, 8846–74.

Barhoush, M., Bilal H. Abed-alguni, and Nour Elhuda A. Al-qudah. (2023). Improved Discrete Salp Swarm Algorithm Using Exploration and Exploitation Techniques for Feature Selection in Intrusion Detection Systems. *Journal of Supercomputing*. Vol. 79. Springer US.

Das, S., Sajal, S. A., Tahsin, P., Etee, K. R., Frederick, T. S., Anwar H. and Sajjan S. (2022). Network Intrusion Detection and Comparative Analysis Using Ensemble Machine Learning and Feature Selection. *IEEE Transactions on Network and Service Management*, 19(4): 4821–33.

Fang, Y., Yu, Y., Xiaoli, L., Jiaxuan, W. and Hao Z. (2024). A Feature Selection Based on Genetic Algorithm for Intrusion Detection of Industrial Control Systems. *Computers and Security* 139, 103675.

Jaw, E., and Xueming W. (2021). Feature Selection and Ensemble-Based Intrusion Detection System: An Efficient and Comprehensive Approach. *Symmetry* 13(10): 1764. https://doi.org/10.3390/sym13101764.

Jupriyadi, A. B., Eki Z. H., Syaiful A. and Ridha M. N. (2024). Wrapper-Based Feature Selection to Improve the Accuracy of Intrusion Detection System (IDS). *Proceedings of 2024 the 10th International Conference on Wireless and Telematics, ICWT*, 1–5.

Kareem, S. S., Reham R. M., Fatma A. H. and Hazem M. E (2022). An Effective Feature Selection Model Using Hybrid Metaheuristic Algorithms for IoT Intrusion Detection. *Sensors*, 22(4): 1–23.

Li, J., Mohd S. O., Hewan C. and Lizawati, M. Y. (2024). Optimizing IoT Intrusion Detection System: Feature Selection versus Feature Extraction in Machine Learning. *Journal of Big Data* 11, 1.

Li, X. K., Wei C., Qianru Z., and Lifa Wu. 2020. 'Building Auto-Encoder Intrusion Detection System Based on Random Forest Feature Selection'. Computers and Security 95: 101851. https://doi.org/10.1016/j.cose.2020.101851.

Mhawi, D. N., Ammar A., and Soukeana H. (2022). Advanced Feature-Selection-Based Hybrid Ensemble Learning Algorithms for Network Intrusion Detection Systems, Symmetry 14, 7.

Nazir, A., and Rizwan A. K. (2021). A Novel Combinatorial Optimization-Based Feature Selection Method for Network Intrusion Detection. Computers and Security 102: 102164. https://doi.org/10.1016/j.cose.2020.102164.

Rani, B., Selva, S. Vairamuthu, and Suresh, S. (2024). Archimedes Fire Hawk Optimization Enabled Feature Selection with Deep Maxout for Network Intrusion Detection. Computers and Security 140, October 2023: 103751.

Sayegh, H. R., Wang, D. and Al-madani.M. A. (2024) Enhanced Intrusion Detection with LSTM-Based Model, Feature Selection, and SMOTE for Imbalanced Data. Applied Sciences (Switzerland) 14, 2: 1–20.

Turukmane, A. V., and Ramkumar D. (2024). M-MultiSVM: An Efficient Feature Selection Assisted Network Intrusion Detection System Using Machine Learning. Computers and Security 137, October 2023: 103587.

Umar, M. A., Zhanfang C., Khaled S., and Yan L. (2024). Effects of Feature Selection and Normalization on Network Intrusion Detection. Data Science and Management. https://doi.org/10.1016/j.dsm.2024.08.001.

Yaras, S and Murat D. (2024). IoT-Based Intrusion Detection System Using New Hybrid Deep Learning Algorithm. Electronics (Switzerland) 13, 6

Yin, Y., Jang-Jaccard, J., Xu, W., Singh, A., Zhu, J., Sabrina, F. and Kwak, J. (2023). IGRF-RFE: A Hybrid Feature Selection Method for MLP-Based Network Intrusion Detection on UNSW-NB15 Dataset. Journal of Big Data 10, 1. https://doi.org/10.1186/s40537-023-00694-8.

Zouhri, H., Ali, I. and Ratnani, A. (2024). Evaluating the Impact of Filter-Based Feature Selection in Intrusion Detection Systems. International Journal of Information Security 23, 2: 759–85.

Akinlade, O., Vakaj, E., Dridi, A., Tiwari, S. and Ortiz-Rodriguez, F. (2023). Semantic Segmentation of the Lung to Examine the Effect of COVID-19 Using UNET Model. In: Jabbar, M.A., Ortiz-Rodríguez, F., Tiwari, S., Siarry, P. (eds) Applied Machine Learning and Data Analytics. AMLDA 2022. Communications in Computer and Information Science, vol 1818. Springer, Cham. https://doi.org/10.1007/978-3-031-34222-6_5 (52–63)

Oyekunle, D. O., Esseme, A. C., Oladipupo, M. A., Oseni, V. E., Adebola, N. T., Nwaiku, M., Nwanakwaugwu, A. C., and Matthew, U. O. (2025). Artificial Neural Network Algorithm in Nutritional Assessment: Implications for Machine Learning Prediction in Nutritional Assessments. Precision Health in the Digital Age: Harnessing AI for Personalized Care (253-276). IGI Global Scientific Publishing. https://doi.org/10.4018/979-8-3693-4422-4.ch013

Benedit, J. (2023). An Appraisal of Database Security in a Business Organization (Case Study: Fintrak Software Company Limited). University of East London, United Kingdom.

Ugbomeh, O., Yiye, V., Ibeke, E., Ezenkwu, C. P., Sharma V. and A. Alkhayyat, (2024). Machine Learning Algorithms for Stroke Risk Prediction Leveraging on Explainable Artificial Intelligence Techniques (XAI). 2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT), Greater Noida, India, 1-6, doi: 10.1109/ICEECT61758.2024.10739320.

Ajagbe, S.A., Awotunde, J.B. and Florez, H. (2024). Intrusion Detection: A Comparison Study of Machine Learning Models Using Unbalanced Datasets. *SN COMPUT. SCI.* **5**, 1028https://doi.org/10.1007/s42979-024-03369-0