Science World Journal Vol. 20(No 3) 2025 https://dx.doi.org/10.4314/swj.v20i3.1
www.scienceworldjournal.org
ISSN: 1597-6343 (Online), ISSN: 2756-391X (Print)
Published by Faculty of Science, Kaduna State University

Full Length Research Article

AN IMPROVED GENETIC ALGORITHM BASED FEATURE SELECTION TECHNIQUE FOR INTRUSION DETECTION IN FOG COMPUTING ENVIRONMENT

[1]Muhammad Yusuf HARUNA, [1]Sahabi A. YUSUF, [1]Mohammed ABDULLAHI, *[2]Jeremiah ISUWA, [1]Abdulaziz SAIDU

[1]Computer Science Department, Ahmadu Bello University, Zaria - Nigeria.
[2]Computer Science Department, Federal University of Kashere, Gombe - Nigeria

*Corresponding Author Email Address: isuwa.jeremiah@fukashere.edu.ng

ABSTRACT
Feature Selection (FS) is critical for reducing the high dimensionality of data, which negatively impacts the classification performance of machine learning models. In the field of Intrusion Detection (ID), where datasets often consist of thousands of attributes and instances, the prevalent issue of data imbalance poses significant challenges, leading to bias in classification tasks. This highlights the pressing need for intelligent techniques to address these challenges effectively. Genetic Algorithm (GA), a widely used evolutionary optimization algorithm for FS, encounters limitations such as slow convergence and a tendency to settle prematurely on suboptimal solutions due to insufficient exploitation capability. These limitations can adversely affect overall performance. Additionally, conventional techniques like the Synthetic Minority Oversampling Technique, commonly employed to handle data imbalance, risk introducing noisy data points into the feature space. To overcome these issues, this study proposes an improved GA-based FS technique featuring an enhanced mutation operator to bolster its exploitation capabilities and deliver improved performance. Furthermore, Adaboost, a more promising machine learning algorithm, is suggested to effectively address data imbalance challenges. The performance of the proposed model was evaluated using benchmark datasets from the Security Laboratory Knowledge Discovery Dataset (NSL-KDD), employing five performance metrics, including accuracy, F1-score, recall, precision, and execution time. The results show that the proposed method outperforms existing techniques across all metrics while effectively tackling the challenges of high-dimensional data and imbalanced datasets, offering a reliable solution for Intrusion Detection.

Keywords: Feature Selection, dimensionality reduction, genetic algorithms, intrusion detection, imbalance data, machine learning

INTRODUCTION

A significant concern in cybersecurity is the sharp increase in network attacks, especially with the growing demand for internet usage and connectivity (Onah et al., 2021). In the field of network security, an Intrusion Detection System (IDS) serves as a safeguard, identifying and blocking intruders from accessing the network (Kumar et al., 2021a). When an intruder attempts to breach the network, the IDS prevents the intrusion, acting promptly to mitigate harm and protect sensitive information (Kumar et al., 2021b). The primary goal of an IDS is to ensure the network's availability, integrity, and confidentiality (Dwivedi et al., 2021). According to researchers such as Dwivedi et al (Dwivedi et al., 2021), IDS is currently considered one of the most effective strategies for network security and protecting against external intruders.

With the advent of fog computing, a paradigm that brings computation, data storage, and networking closer to end-users, new challenges emerge for cybersecurity (Atlam et al., 2018; Tsai, 2002). While fog computing reduces latency and enhances efficiency in distributed environments, it also introduces additional points of vulnerability, as fog devices are often targeted by attackers (Sadaf & Sultana, 2020). These vulnerabilities necessitate integrating IDS into fog computing infrastructures to maintain the security and reliability of network communications, particularly in such distributed and latency-sensitive systems (Onah et al., 2021). By leveraging IDS in fog computing, it is possible to address these threats proactively, ensuring secure and uninterrupted services for end-users.

Recent research on IDS within the context of fog computing has increasingly leveraged the power of machine learning, utilizing both supervised and unsupervised approaches to address evolving cybersecurity challenges. Among these, deep learning methodologies and ensemble-based detection techniques have gained significant traction due to their ability to handle complex and high-dimensional data effectively (Sadaf & Sultana, 2020). These advanced techniques aim to improve the accuracy, efficiency, and adaptability of IDS, making them better suited to the dynamic and distributed nature of fog computing environments (Dwivedi et al., 2021). By incorporating these intelligent approaches, researchers are striving to develop robust systems capable of mitigating the unique security threats posed by the decentralized architecture of fog computing.

While machine learning and deep learning techniques have proven effective in enhancing the accuracy and efficiency of IDS in fog computing, their performance can be significantly hindered by high-dimensional data. This increases computational complexity, causes overfitting, and reduces model accuracy, among other challenges (Isuwa et al., 2023; Shallangwa et al., 2024). To address this issue, metaheuristic algorithms have emerged as powerful tools for optimizing the feature selection process. Inspired by natural phenomena such as biological evolution and swarm intelligence, these algorithms provide a robust framework for identifying the most relevant features, thereby reducing dimensionality and improving model performance (Avinash et al., 2024). Examples of such algorithms include the Genetic Algorithm (GA) (Ali et al., 2020), Whale Optimization Algorithm (WOA) (Tair et al., 2022), Grey Wolf Optimization (GWO) (Chakraborty et al., 2022), Particle Swarm Optimization (PSO) (Isuwa et al., 2022), and Salp Swarm Optimization (SSA) (Tubishat et al., 2021). The application of these algorithms has shown promise in solving real-

world optimization problems across diverse fields, including engineering and healthcare (Dwivedi et al., 2021). Effective feature selection relies on the core components of metaheuristic algorithms, exploitation, and exploration, which are essential for finding optimal solutions (Katoch et al., 2021).

The Genetic Algorithm (GA), one of the earliest and most renowned evolutionary-based metaheuristic algorithms, is grounded in principles of genetic and evolutionary theory. It mimics biological processes like natural selection and reproduction (Azad et al., 2022; Ding et al., 2020). The key components of GA include chromosome representation, fitness selection, and genetic operators such as crossover and mutation (Katoch et al., 2021). Chromosomes, typically represented as binary strings, have loci that carry alleles (0 or 1), which define potential solutions in the search space (Katoch et al., 2021). The fitness function assigns a value to each chromosome, determining its suitability for selection. During the crossover process, chromosomes exchange genetic material, while mutation randomly alters parts of chromosomes to introduce variability (Azad et al., 2022; Ding et al., 2020).

Despite its effectiveness, GA faces challenges such as slow and premature convergence to suboptimal solutions, often due to limited exploitation capabilities (Azad et al., 2022). While mutation operators are designed to introduce diversity, their low probability settings can restrict the algorithm's ability to escape local optima. Researchers have proposed enhancements, such as scramble and bit-flipping mutation operators, to address these limitations. The scramble mutation, effective for large-scale problems, randomly selects and shuffles subsets of genes, though it may disrupt population structure. In contrast, the bit-flipping mutation alters individual genes to create variability, though its incremental nature may limit its impact on large-scale problems. To overcome these challenges, this research aims to develop an improved mutation operator for GA, enhancing its exploitation capability and reducing convergence issues. By addressing these limitations, the proposed model seeks to enhance the performance of IDS in fog computing, ensuring better security and operational efficiency. Specifically, this study aims to:

i. Develop a hybrid scramble and bit-flipping mutation operator in GA to strengthen the exploitation ability of the algorithm.
ii. Utilize the AdaBoost machine learning algorithm to effectively tackle data imbalance challenges.
iii. Implement the improved model and evaluate its performance using the NSL-KDD dataset. Perform a detailed analysis considering F-measure, accuracy, and time complexity, and compare the results with recent studies in the field to determine the effectiveness and advancements of the proposed framework.

The remainder of this paper is organized as follows: Section 2 presents the background of the study and a comprehensive review of relevant literature. Section 3 discusses the proposed approach. Section 4 covers the experiments, comparisons, and detailed analysis of the results. Finally, Section 5 concludes the work by summarizing key findings and suggesting directions for future research.

This section begins by exploring the concepts of Intrusion Detection Systems (IDS) and Fog Computing, emphasizing their complexities and practical applications. It then delves into dimensionality reduction techniques, metaheuristic algorithms, the Adaboost Machine Learning algorithm, and challenges associated with imbalanced data. Finally, a comprehensive review of the current literature is presented, highlighting key strengths and limitations in the field.

Intrusion Detection System (IDS)

An Intrusion Detection System (IDS) is a tool designed to monitor a network for malicious activities or violations of established policies. It automatically identifies intrusions, attacks, or security policy breaches at both the network and host levels using proactive detection mechanisms (Chiba et al., 2022). IDS are classified based on their approach to monitoring intrusive behaviors: Host-Based Intrusion Detection Systems (HIDS) and Network-Based Intrusion Detection Systems (NIDS) (Mishra et al., 2019). NIDS focuses on monitoring network activities by analyzing behaviors reflected through network devices such as switches, routers, and network taps. It detects threats by examining potential attacks hidden within network traffic. In contrast, HIDS monitors and evaluates data on individual or multiple host systems to detect attacks. It typically includes components such as operating systems, system files, and application files (Vinayakumar et al., 2019).

According to Onah et al., (Onah et al., 2021), attacks are classified into four main categories. Denial of Service (DoS) attacks overwhelm a network or system with excessive requests, rendering it inaccessible to legitimate users. Remote-to-user (R2L) attacks involve sending packets to a target system over a network to exploit vulnerabilities and gain unauthorized user privileges. User to Root (U2R) attacks occur when an attacker gains initial access as a regular user and then escalates privileges to achieve root-level control. Lastly, Probing attacks involve scanning networks or systems to identify vulnerabilities that can be exploited later.

Fog Computing

Fog computing, introduced by Cisco, extends cloud computing by enabling the local processing of end-user requests through fog devices, thereby reducing communication latency (Dwivedi et al., 2021). This approach addresses the limitations of traditional cloud computing, particularly in low-latency, real-time data processing scenarios, while improving network bandwidth efficiency. However, ensuring secure and legitimate network communication on fog devices remains critical due to susceptibility to malicious attacks (Sadaf & Sultana, 2020).

Dimensionality Reduction

Dimensionality reduction (DR) is a preprocessing technique designed to enhance learning efficiency by reducing training time and improving accuracy, primarily by eliminating noise (Petinrin et al., 2023). DR is typically divided into two main types: feature extraction and feature selection, as illustrated in Figure 1. Feature extraction generates new features by combining existing ones, with methods such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Multidimensional Scaling. In contrast, feature selection identifies and retains the most relevant features based on specific evaluation criteria, optimizing the set of features used for final classification.

Science World Journal Vol. 20(No 3) 2025 https://dx.doi.org/10.4314/swj.v20i3.1
www.scienceworldjournal.org
ISSN: 1597-6343 (Online), ISSN: 2756-391X (Print)
Published by Faculty of Science, Kaduna State University

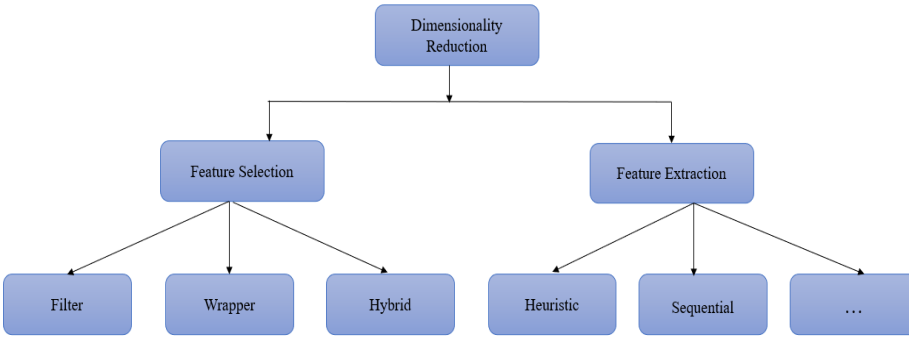

Figure 1: Dimensionality reduction (Effrosynidis & Arampatzis, 2021a)

Feature Selection (FS)
Feature selection (FS) is the process of identifying and selecting the most significant features from a dataset for tasks such as classification, regression, or clustering (Solorio-Fernández et al., 2020). FS offers several benefits, including reduced dataset size, improved classification accuracy, minimized storage requirements, and lower computational costs (Hancer et al., 2020). The FS process typically involves two key stages: subset generation, where a subset of features (considered potentially optimal) is created from the original dataset, and subset evaluation, where the selected subset is assessed to determine its optimality. Additionally, feature selection can be broadly categorized into three types: Filter, Wrapper, and Embedded or Ensemble methods (Moran & Gordon, 2019). Figure 2 illustrates the overall feature selection process.

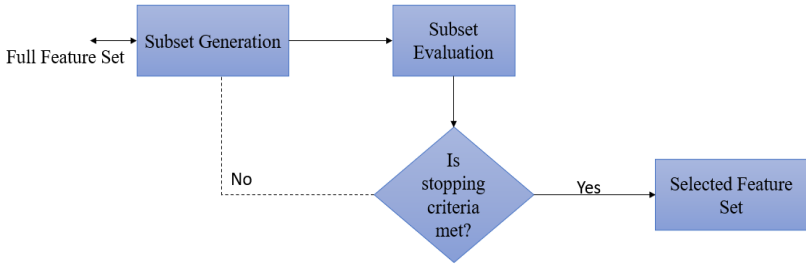

Figure 2: The Feature Selection Process (Hancer et al., 2020)

Filter method
Filter methods typically rank features by assigning scores based on specific evaluation criteria. Features with the highest scores are then selected for further analysis. These evaluation criteria can be categorized into univariate measures, which assess features individually, or multivariate measures, which consider the relationships between multiple features (Hancer et al., 2020).

Wrapper Method
The wrapper method of feature selection involves searching for and selecting feature subsets using a predefined learning algorithm. The model's error rate is evaluated by testing it on a validation set, and each subset is assigned a corresponding score (Jeremiah et al., 2022). The subset with the highest score is ultimately selected. While this approach often yields better performance by tailoring feature selection to the specific algorithm, it is generally more computationally intensive compared to filter methods (Shallangwa et al., 2024). Examples of algorithms in this approach include heuristic and metaheuristic algorithms.

Embedded Method
The embedded method of feature subset selection integrates the strengths of both filter and wrapper approaches by performing feature selection and model training simultaneously, typically using a learning algorithm (Effrosynidis & Arampatzis, 2021b; Hancer et al., 2020). Common classifiers employed in embedded feature selection include Random Forest, and Decision Tree among others.

Metaheuristic Algorithms
Laporte and Osman (Laporte & Osman, 1995) defined metaheuristic algorithms as iterative generation processes that guide subordinate heuristics by strategically combining exploration and exploitation of the search space. These algorithms utilize learning mechanisms to efficiently organize knowledge and identify near-optimal solutions. By balancing exploration and exploitation, metaheuristics provide effective solutions to NP-hard problems with numerous variables and nonlinear objective functions (Hassan et al., 2023). Metaheuristic algorithms are classified based on various criteria, such as their inspiration (nature-inspired vs. non-nature-inspired), search strategy (population-based vs. single-point search), adaptability (dynamic vs. static objective functions), neighborhood exploration (single vs. multiple structures), and memory usage (memory-based vs. memory-less methods) (Hayatu et al., 2024).

Genetic Algorithm
The Genetic Algorithm (GA) is an optimization method inspired by the principles of natural selection and survival of the fittest

Science World Journal Vol. 20(No 3) 2025 https://dx.doi.org/10.4314/swj.v20i3.1
www.scienceworldjournal.org
ISSN: 1597-6343 (Online), ISSN: 2756-391X (Print)
Published by Faculty of Science, Kaduna State University

(Michalewicz & Schoenauer, 1996). It is a population-based search algorithm that iteratively refines solutions by applying genetic operators to evolve a population of potential solutions (Katoch et al., 2021). Key components of GA include chromosome representation, selection, crossover, mutation, and fitness function evaluation.

The GA begins with the random initialization of a population (G) containing n chromosomes. The fitness of each chromosome is then evaluated. Two chromosomes, $Y1$ and $Y2$, are selected from the population G based on their fitness values. A single-point crossover operator with a specified crossover probability (Yp) is applied to $Y1$ and $Y2$, producing an offspring, O. This offspring is subsequently subjected to a uniform mutation operator with a defined mutation probability (Mp), resulting in a mutated offspring, O'. The new offspring O' is added to the next generation of the population. The processes of selection, crossover, and mutation are repeated until the new population is complete. This cycle continues iteratively until a termination condition, such as a maximum number of generations or a satisfactory fitness level, is met. Figure 3 illustrates the flowchart of the GA process.

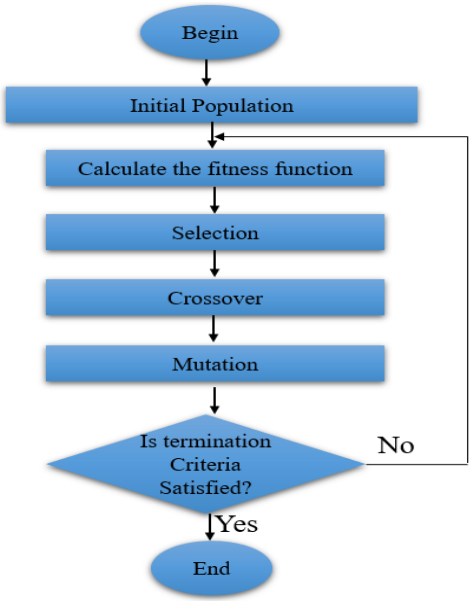

Figure 3: The Genetic algorithm process of feature selection (Katoch et al., 2021)

Imbalanced Data

Data imbalance occurs when one or more class variables have significantly fewer instances compared to others, leading to inaccurate model representations in classification tasks (Thabtah et al., 2020). This issue is prevalent in areas like environmental data, fraud detection, network intrusion detection, and medical diagnostics, where rare or exceptional events are the primary focus of detection (Thabtah et al., 2020). To address this problem, two main approaches have been increasingly employed over the years: data-driven and algorithm-driven (Sahebi et al., 2020). Data-driven methods, such as Random Up-Sampling and Down-Sampling, adjust the class distribution in the dataset, while algorithm-driven methods modify the classification algorithm to help it better handle imbalanced data, enabling the minority class to be effectively learned (Paoletti et al., 2023).

Ensemble Machine Learning

Ensemble learning is a machine learning approach that addresses classification problems by combining multiple weak classifiers to create a stronger, more robust classifier (Tang et al., 2020). Its primary advantage lies in its ability to enhance generalizability and robustness by aggregating predictions from several base estimators, as opposed to relying on a single model. The underlying principle is that while individual weak classifiers have limited predictive capabilities, combining them can produce a more accurate and stable classifier. Ensemble learning algorithms can be categorized based on the dependencies between weak classifiers. In methods like Boosting, weak classifiers are interdependent, with each iteration focusing on correcting the errors of its predecessor. Conversely, in approaches such as Bagging and Random Forest, weak classifiers operate independently, ensuring diversity in predictions (Tang et al., 2020). Boosting, a popular ensemble learning technique involves the following steps: A weak classifier is initially trained on the dataset. Based on its performance, the training sample distribution is adjusted, giving more weight to misclassified samples to improve subsequent classifiers' focus on these challenging cases. This process continues iteratively until a predefined number of weak classifiers, T, are trained. Finally, the outputs of all T weak classifiers are weighted and combined to construct a strong classifier (Sharma et al., 2021).

Freund (Freund, 1995) introduced the AdaBoost algorithm, one of the most powerful and widely used machine learning algorithms. As an ensemble learning method, AdaBoost combines multiple base models, often starting with weak classifiers, to construct a stronger and more accurate predictive model. This approach is particularly effective for handling imbalanced datasets, and improving overall model accuracy (Liu et al., 2019; Long et al., 2021a; Pham et al., 2021; Zharmagambetov et al., 2021).

AdaBoost operates iteratively, where each step focuses on training a base learner to enhance the model's overall performance (Zharmagambetov et al., 2021). Initially, equal weights are assigned to all data instances. During each iteration, these weights are adjusted: instances that are misclassified receive higher weights, while correctly classified instances are assigned lower weights. This reweighting process ensures the algorithm focuses more on challenging cases in subsequent iterations.

Fog computing, a rapidly evolving research field, has gained significant attention due to its low cost, performance, and ability to process data with low latency. This section reviews recent literature on the subject. For instance, Moh and Raju, (Moh & Raju, 2018) conducted an extensive study on various machine learning techniques, such as Decision Tree, KNN, and Random Forest, and explored how machine learning-based intrusion detection systems at the fog layer could detect anomalies and attacks. Peng et al., (Peng et al., 2018) proposed an IDS using a decision tree that detects not only four attack types but also twenty-two variations. The IDS includes data preprocessing, normalization, and decision tree detection, but it does not address the issue of imbalanced data. Bhuvaneswari and Selvakumar (Bhuvaneswari & S., 2020) introduced anomaly detection algorithms for fog environments that classify IoT traffic as normal or malicious, forwarding malicious traffic to the cloud for mitigation. Unlike existing centralized deep learning systems, this approach resolves the class imbalance issue and improves detection accuracy in multiclass classification

scenarios. Similarly, Sadaf and Sultana, (Sadaf & Sultana, 2020) proposed the Auto-If method for intrusion detection using autoencoders (AE) and isolation forests (IF), focusing solely on binary classification and treating all dataset features as equally significant. Onah et al., (Onah et al., 2021) implemented a genetic algorithm-based feature selection combined with Naïve Bayes for anomaly detection in fog computing, but they did not adequately address the imbalanced data issue, potentially biasing the classification results. Despite achieving 99.73% accuracy and a low false positive rate of 0.6%, this limitation remains a concern.

Aliyu et al., (Aliyu et al., 2022) introduced a lightweight, human-immune-inspired, anomaly-based intrusion detection system (IDS) for the fog layer. The system achieves low resource overhead by distributing IDS functions between fog nodes and the cloud, with an accuracy of up to 98.8%. Additionally, it recorded a 10% reduction in energy consumption at the fog node compared to deploying a neural network directly on the fog node. Mohamed and Ismael, (Mohamed & Ismael, 2023) employed genetic algorithms to optimize the network's connecting weights and individual neuronal biases. They conducted experiments using the UNSW-NB15 and ToN_IoT datasets on a Raspberry Pi4 as a fog node for binary-class classification. The results demonstrated that the optimized weights and biases outperformed those of a non-optimized neural network, yielding superior processing speed and detection accuracy. Nabi and Zhou, (Nabi & Zhou, 2024) focused on developing a highly accurate classifier with a low number of false alarms to improve automated intrusion detection. Their work aimed to enhance classifier performance and address challenges posed by high dimensionality in intrusion detection, ultimately leading to more precise and effective detection. They experimented with the NSL-KDD dataset, a popular benchmark in this field, to achieve their goals.

Ensemble methods have gained significant attention in the scientific community due to their power and versatility across various applications. As a result, numerous intrusion detection algorithms based on ensemble learning have been developed. Wang (Wang et al., 2016) addressed the challenge of unbalanced data samples using a Bayesian network and random tree as basic classifiers, proposing a new intrusion dataset, the KDDcup99, to evaluate the model. One common strategy for overcoming restricted datasets and reducing overfitting in deep learning models is artificially enlarging the dataset through label-preserving modifications.

Yuan et al., (Yuan et al., 2016) used a semi-supervised version of Adaboost for network anomaly detection. Han Liu et al., (Liu et al., 2019) proposed PwAdaBoost, an AdaBoost method based on possible worlds, capable of managing uncertain data and handling ambiguity in intrusion analysis. Gao et al., (Gao et al., 2019) introduced an adaptive ensemble machine learning method for identifying network attacks like DoS, Probe, U2R, and R2L, incorporating classifiers such as decision trees, random forests, KNNs, and deep neural networks (DNNs). Their simulations demonstrated that the proposed method outperforms other learning algorithms.

He et al., (He et al., 2020) applied the AdaBoost algorithm for fault diagnosis, achieving good diagnostic accuracy in simulation results. Tang et al., (Tang et al., 2020) enhanced the AdaBoost method by developing MF-Adaboost to detect low-rate denial of service (LDoS) attacks in communication networks, using a set of key network traffic features for classification. The simulation results confirmed the strategy's effectiveness in detecting LDoS attacks.

Abbas et al., (Abbas et al., 2022) proposed an ensemble-based intrusion detection model using decision trees, logistic regression, and Naive Bayes with a voting classifier. The model, tested on the CICIDS2017 dataset, showed significant improvement in accuracy for both binary and multi-class classification scenarios compared to existing models. This paper presents a stacked ensemble system that utilizes the benchmark NSL-KDD dataset to compare its performance against popular machine learning algorithms such as ANN, CART, random forest, SVM, and other methods proposed in the literature. By combining multiple machine learning algorithms, the ensemble system can identify attacks more effectively than traditional single-algorithm approaches (Rajadurai & Gandhi, 2022). In another study, Alghanam et al., (Abu Alghanam et al., 2023) propose an enhanced version of pigeon-inspired optimization (PIO) called LS-PIO, which incorporates a local search technique to improve optimization performance. They also apply an ensemble learning strategy based on multiple one-class classifiers to further enhance the performance of the proposed network intrusion detection system (NIDS). The LS-PIO and ensemble-based NIDS were evaluated using four benchmark datasets: BoT-IoT, UNSW-NB15, NSL-KDD, and KDDCUP99. Additionally, Jemili et al., (Jemili et al., 2024) conducted a comprehensive set of experiments using datasets such as N-BaIoT, NSL-KDD, and CICIDS2017 to evaluate and compare machine learning methods like Random Forest, XGBoost, and decision trees. Their work culminated in the development of a hybrid intrusion detection model that merges the strengths of these algorithms, with results indicating that the combination of Random Forest and XGBoost yields exceptional performance.

MATERIALS AND METHODS

This chapter outlines the research methodology employed in the study. It includes details on the parameters and settings of the proposed technique, the benchmark datasets used, the system specifications, the proposed methodology, and the experimental design.

Dataset Description

This study used the NSL-KDD dataset to assess the effectiveness of the proposed method. Widely adopted in IDS research, the dataset includes three subsets: the full dataset, 20% for training, and the full testing dataset. Each instance is defined by 41 attributes and a label indicating whether the record represents an attack or is normal. Tables 1 and 2 display the number of normal and attack instances in each testing and training fold.

Table 1: Number of attack cases in the training dataset (Onah et al., 2021).

Types of attack	Number of instances
Normal	67343
DoS	45927
Probe	11656
R2L	995
U2R	52
Total	**125973**

Table 2: Number of attack cases in the testing dataset (Onah et al., 2021)

Types of attack	Number of instances
Normal	9711
DoS	7456
Probe	2421
R2L	2756
U2R	200
Total	**22544**

Classifier (AdaBoost)

As discussed earlier, the Adaboost algorithm will be used in this study to evaluate selected features and address the challenge of data imbalance. This learning algorithm selects m groups of training data at random from the sample space, initializes the test data distribution weight $D_1 = 1/m$, and obtains the first base classifier $h1$ using the initial data distribution; the t^{th} base classifier is constructed iteratively, utilizing the classifier's overall prediction error ε_t, and discover a zero solution for the exponential loss function's $\ell_{\exp}(\beta_t|D_t)$ partial derivative (1) to determine the classifier's weight β_t.

$$e_t = \sum_i D_i\,(i)\ \ (if\ h_t\,(x_i) \neq y_i) \quad (1)$$

$$\frac{\delta l\,(\beta_t|D_t)}{\delta \beta_t} = -e^{-\beta_t}\,(1-\varepsilon_t) + e^{\beta_t}\,\varepsilon_t \quad (2)$$

$$\beta_t = \frac{1}{2}\ln(\frac{1-e_t}{e_t}) \quad (3)$$

Using the pertinent parameters learned during the t^{th} base classifier training, adjust the weights of the test data D_{t+1} for the $t+1^{th}$ iteration, The normalizing factor is β_t, y_i represents the actual classification outcome, y_i^t represents the t^{th} base classifier's predicted classification results.

$$D_{t+1}\,(i) = \frac{D_t(i)}{\beta_t} * \exp\,(-\beta_t y_i y_i^t) \quad (4)$$

Greater weight is given to the fundamental classifier with a low classification error rate, and a lower weight value is assigned to the basic classifier with a high classification error rate. Following that, the linear combination based on T base classifiers is obtained.

$$f(x) = \sum_{t=1}^{T} \beta_t h_t(x) \quad (5)$$

Based on the linear combination, the sign function is transformed to get the result of the strong classifier. Algorithm 1 presents the overall stages of the Adboost learning algorithm.

$$H(x) = sign(f(x)) = sign(\sum_{t=1}^{T} \beta_t h_t(x)) \quad (6)$$

Algorithm 1: AdaBoost Algorithm (Long et al., 2021b)

Given $(x_1, y_1),\ldots,(x_m, y_m)$ Where $x_1, \in x, y_i \in \{-1, +1\}$,
Initialize: $D_1 = 1/m$ for $i = 1, \ldots, m$.
For $t = 1, \ldots, T$:

1 Train weak learner using distribution D_1.
2 Get weak hypothesis $\varkappa \longrightarrow \{-1, +1\}$.
3 Aim: select h_t with low weighted error:

$$e_t = \sum_i D_i\,(i)\ \ (\text{if } h_t\,(x_i) \neq y_i)$$

4 Choose $\beta_t = \frac{1}{2}\ln(\frac{1-e_t}{e_t})$
5 Update, for $i = 1, \ldots m$:

$$D_{t+1}\,(i) = \frac{D_t(i)}{\beta_t} * \exp\,(-\beta_t y_i y_i^t)$$

Where β_t is a normalization factor (chosen so that D_{t+1} will be a distribution).
Output the final hypothesis:

$$H(x) = \text{sign}(\sum_{t=1}^{T} \beta_t h_t(x))$$

Description of the Proposed Improved Genetic Algorithm

This study introduces an enhanced mutation operator in the Genetic Algorithm (GA), combining the strengths of scramble and bit-flipping mutation techniques. In scramble mutation, a randomly selected subset of non-contiguous genes within a solution vector is shuffled, promoting exploration by diversifying the population. While effective for large-scale problems, its tendency to disrupt population stability can sometimes degrade solution quality. Bit-flipping mutation, on the other hand, alters a single gene in the solution vector with a specified probability, flipping its value (e.g., 0 to 1 or 1 to 0). This method supports exploitation by fine-tuning solutions but is less effective in large-scale problems due to its limited impact on the solution space.

To address the limitations of individual mutation techniques, this study proposes a hybrid approach combining scramble and bit-flipping mutations as presented in algorithm 2. The hybrid method enhances the Genetic Algorithm's exploitation capability, mitigating issues like slow convergence and premature stagnation in suboptimal solutions, particularly in feature selection tasks. The proposed hybrid method operates as follows:

1. For mutation, apply the scramble mutation operator. After applying the scramble mutation operator, two new solution vectors are generated say $C1$ and $C2$.
2. Fitness values of the generated solution vectors $C1$ and $C2$ are calculated.
3. If the fitness of either solution vector deteriorates or fails to improve, select a random bit (gene) based on some probability from the initial subset of genes selected in (1) and flip.

Science World Journal Vol. 20(No 3) 2025 https://dx.doi.org/10.4314/swj.v20i3.1
www.scienceworldjournal.org
ISSN: 1597-6343 (Online), ISSN: 2756-391X (Print)
Published by Faculty of Science, Kaduna State University

4. Re-calculate the fitness value of the newly created solutions say C_{1bf} and C_{2bf} after bit flipping.
5. If the fitness of either C_{1bf} and C_{2bf} is better than $C1$ and $C2$, replace solution vectors with the new solution,
6. Otherwise, return to the initial population.

Algorithm 2: Improved Genetic Algorithm with Hybrid Mutation Operator

Input: Dataset; Population size: N; Maximum number of iterations: $\mathrm{M_{max}}$
Output: Class labeled test instance

Randomly generate initial solution X(0)
For $\mathrm{K} = 1$ to N do
  Evaluate fitness of individual of X(0);
End for
$\mathrm{m} = 0$
While $\mathrm{X(m} \leq \mathrm{M_{max}})$**do**
  Select two individuals in X(0) based on their fitness
  Apply crossover operator to selected individual from step 3 resulting in new individuals
  Apply scramble mutation operator to new individuals from step 5
  Evaluate the fitness of individual of X(t)
  If fitness deteriorates then
    Select a random gene from the initial subset of the gene
    Flip the selected gene based on some probability
    Re-evaluate the fitness of individual of X(t) after bit flipping
    If fitness improves then
      Replace the original individual X(t) with the mutated one
    End if
  End if
  $\mathrm{t} = \mathrm{t} + 1$:
End While
Return individual with the highest fitness

Model Building

The process begins with two pre-processing phases: encoding categorical variables into numeric values and normalizing the dataset. These steps aim to reduce complexity, minimize misclassification errors, and prepare the data for effective use in the proposed model. Before the optimization algorithm is applied, key parameters are initialized, and necessary configurations are established. This includes randomly initializing the population of search agents in the improved GA. Additionally, the sigmoid transfer function is employed to convert continuous values into discrete ones, with 1 representing a selected feature and 0 indicating otherwise. The chromosomal genes are encoded as bits based on the characteristics of the NSL-KDD dataset. Each individual is evaluated based on its fitness function, which guides the search for optimal solutions.

The entire population is then subjected to selection and recombination operations, each with a probability of 0.6, to explore new solutions within the search space. To further refine the solutions, random modifications are applied using the hybrid scramble and bit-flipping mutation operators, with a mutation probability of 0.033. After dimensionality reduction, as illustrated in Algorithm 2, the NSL-KDD dataset's features are reduced to 17. In this study, the AdaBoost classifier is utilized to evaluate the performance of the selected feature subset, enabling both the evaluation of feature subsets and the classification of attacks based on the selected features. The overall architecture of the proposed model is illustrated in Figure 4.

Model Performance Evaluation

To evaluate the performance of the proposed model against existing methods in the literature, the accuracy, precision, recall, F1-score, and execution time are used as performance metrics in this study. These metrics are defined as follows:

i. Accuracy: the proportion of total correct predictions

$$Accuracy = \frac{TP*TN}{TP*TN*FP*FN} \quad (7)$$

ii. Precision (P): the ratio of correctly identified positive instances to the total instances predicted as positive.

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

Science World Journal Vol. 20(No 3) 2025 https://dx.doi.org/10.4314/swj.v20i3.1
www.scienceworldjournal.org
ISSN: 1597-6343 (Online), ISSN: 2756-391X (Print)
Published by Faculty of Science, Kaduna State University

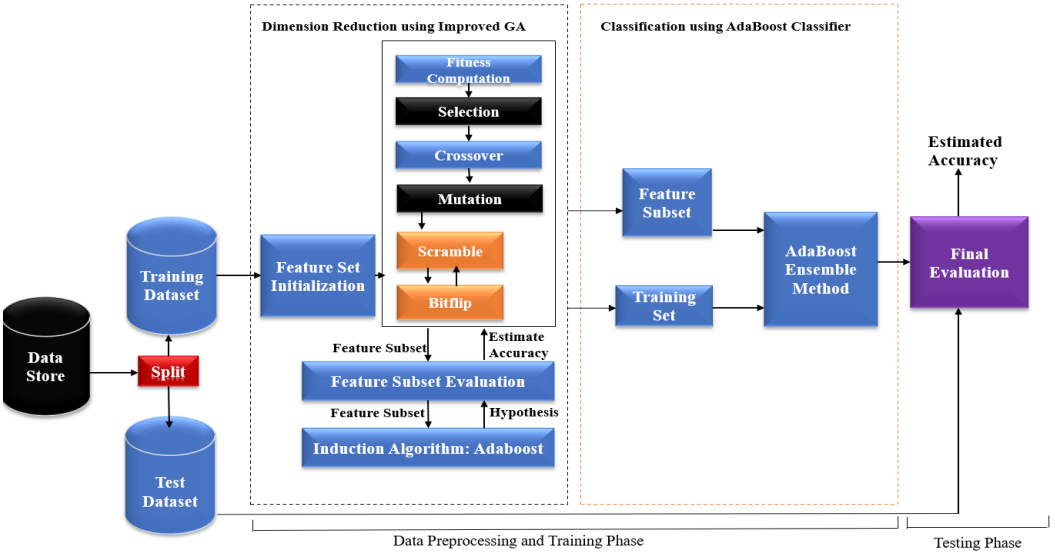

Figure 4: Architecture of the proposed model.

i. Recall (*R*): the ratio of correctly identified positive instances to the total actual positive instances.

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

ii. F1 score: the harmonic mean of precision and recall, balancing their contributions to measure a model's accuracy.

$$F1\ score = \frac{2PR}{P+R} \quad (10)$$

iii. The execution time: the computation time of the propo sed method.

RESULTS AND DISCUSSION

All experiments were carried out on the Google Colab platform using Python, with k-fold cross-validation applied for performance evaluation. The computations were performed on a machine equipped with an Intel Core i5-3360M processor (2.80 GHz) and 8GB of RAM. The proposed model's effectiveness was evaluated through three experimental stages. In the first stage, the model's performance was assessed on a 20% subset of the dataset as in the study by (Onah et al., 2021). The second stage involved the use of optimal hyperparameters for the Adaboost algorithm, determined through grid search, still applied to the subset. Finally, in the third stage, the evaluation was extended to the entire dataset, retaining the optimal settings and comparing the results with those from existing studies in the literature.

Experiment Stage I:

This experiment evaluated the model's performance using only 20% of the entire dataset with the following default hyperparameter settings of the Adaboost algorithm (Table 3). The model achieved impressive results, with accuracy, precision, recall, F1-score, and execution time of 99.10%, 99.30%, 99.21%, 99.25%, and 2 seconds, respectively, as displayed in Table 4.

Table 3: Hyperparameters for the experiment stage 1

Parameters	Values
Learning _rate	0.1
n_ estimator	50
Base _estimator	Decision Tree

Table 4: Performance of the proposed model using 20%of the dataset

Measure	Values (%)
Accuracy	99.10
Precision	99.30
Recall	99.21
F1-Score	99.25
Execution time	2 secs

Experiment Stage 2:

In this stage of the experiment, grid search was employed to explore different combinations of hyperparameter settings while still maintaining 20% of the dataset. The analysis revealed the following optimal configurations for the hyperparameters (Table 5).

Table 5: Selected trainable hyperparameters by the grid search for the experiment.

Parameters	Values
Learning _rate	0.7
n_ estimator	900
Base _estimator	Decision Tree

The performance achieved in this experiment, as measured by accuracy, precision, recall, F1-score, and time taken, were 99.72%, 99.69%, 99.70%, 99.70%, and 11 seconds, respectively, as shown in Table 6.

Table 6: Performance of the proposed system using optimal parameters and 20%of the dataset

Measure	Values (%)
Accuracy	99.72
Precision	99.69
Recall	99.70
F1-Score	99.70
Execution time	11seconds

Science World Journal Vol. 20(No 3) 2025 https://dx.doi.org/10.4314/swj.v20i3.1
www.scienceworldjournal.org
ISSN: 1597-6343 (Online), ISSN: 2756-391X (Print)
Published by Faculty of Science, Kaduna State University

Experimental Stage 3:
In this stage, we maintained the optimal hyperparameter settings from Stage 2 but evaluated the model using the entire dataset. The model achieved exceptional results across all performance metrics, with an accuracy of 99.69%, precision of 99.31%, recall of 99.50%, F1-score of 99.40%, and a time taken of 192 seconds. These results highlight the model's effectiveness and scalability when applied to the full dataset. Table 7 displays these findings and compares them with the results from Onah et al. (Onah et al., 2021). The proposed method outperforms the existing approach in all metrics, demonstrating its superior performance and efficiency.

Table 7: Performance of the proposed system in comparison with an existing study using the entire dataset

Models	Accuracy	Precision	Recall	F1-Score	Execution time
Proposed Model	**99.69%**	**99.31%**	**99.50%**	**99.40%**	**192 seconds**
(Onah et al., 2021)	95.24%	59.40%	57.62%	58.42%	189 seconds

Discussion of Results: Comparison between the Proposed System and Existing Systems Using the Entire Dataset
The performance of the proposed model was evaluated using the entire dataset and compared with the results of existing methods, specifically, the model proposed by Onah et al., (Onah et al., 2021). The evaluation demonstrated the superior effectiveness and efficiency of the proposed system in terms of multiple performance metrics, including accuracy, precision, recall, F1-score, and execution time as shown in Figures 5 and 6 respectively.

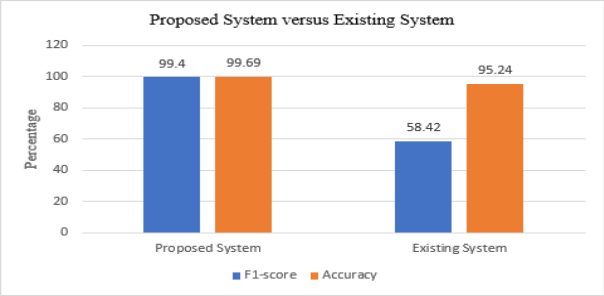

Figure 5: Evaluation of the Existing System vs. the Proposed System Using the Entire Dataset: F1-Score and Accuracy Comparison

The proposed system achieved remarkable results: an accuracy of 99.69%, precision of 99.31%, recall of 99.50%, F1-score of 99.40%, and an execution time of 192 seconds. These results highlight the model's robustness and ability to scale effectively when applied to the entire dataset. In contrast, Onah et al., (Onah et al., 2021) reported lower performance across the same metrics, with an accuracy of 95.24%, precision of 59.40%, recall of 57.62%, F1-score of 58.42%, and an execution time of 189 seconds.

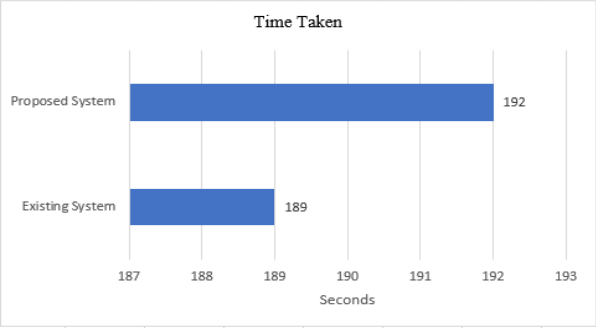

Figure 6: Evaluation of the Existing System vs. the Proposed System Using the Entire Dataset: Execution Time Comparison

This comparison demonstrates that the proposed model outperforms the existing approach by a significant margin in terms of accuracy and the ability to effectively detect relevant instances (higher precision and recall). The enhanced mutation operator, which combines the benefits of scramble and bit-flipping techniques, contributes to the model's ability to avoid slow convergence and premature stagnation, resulting in improved feature selection outcomes. Moreover, the reduced impact of imbalanced data, achieved through a more robust Adaboost technique, strengthens the overall performance of the model, making it more reliable for real-world applications.
In conclusion, the proposed method not only provides better predictive performance but also addresses key challenges such as imbalanced data handling and GA's exploitation ability, making it a promising approach for feature selection in IDS in Fog computing environments. The significant improvements over Onah et al., (Onah et al., 2021) underline the effectiveness of the proposed modifications, and future work could further explore the integration of additional techniques to further optimize the system's performance.

Discussion of results: Comparison between the Proposed System and Existing Systems Using Subset of Dataset
The performance of the proposed model was also evaluated using the subset of the dataset (selected features) and compared with the results of the existing methods, specifically, the model proposed by Onah et al., (Onah et al., 2021). The evaluation demonstrated the superior effectiveness and efficiency of the proposed system in terms of multiple performance metrics, including accuracy, F1-score, and execution time as shown in Figures 7 and 8 respectively.
It can be seen from Figure 7 that the proposed method attains significantly superior performance in F1-score (99.70%) compared to the existing study, which achieved 63.03%. This can be attributed to the utilization of the Adaboost learning algorithm in addressing the data imbalance problem. Adaboost enhances the model's ability to focus on misclassified instances by iteratively adjusting the weights of samples, thereby improving classification performance in minority classes. This targeted approach ensures a more balanced contribution of all classes to the overall model performance, ultimately leading to a higher F1 score and better overall results.
Although the existing study demonstrates a marginally higher accuracy of 99.73% compared to the proposed method's 99.72%, this difference is statistically negligible and unlikely to have a practical impact. Furthermore, the proposed method offers additional advantages, such as greater robustness as

https://dx.doi.org/10.4314/swj.v20i3.1

demonstrated with the F1 score, making it a strong alternative for the task at hand. Such trade-offs underscore the importance of evaluating algorithms beyond accuracy alone.

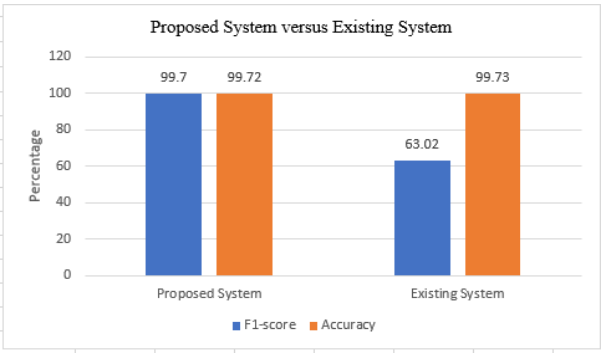

Figure 7: Evaluation of the Existing System vs. the Proposed System Using Subset of the Dataset: F1-Score and Accuracy Comparison

It is also worth pointing out that even though the proposed model did not achieve the best execution time, it obtained competitive results in two key evaluation metrics: F1-score and accuracy. This performance can be attributed to the integration of advanced optimization techniques, such as the hybrid mutation operator and the Adaboost learning algorithm, which enhance the model's ability to explore and exploit the solution space effectively. While these techniques improve classification accuracy and F1-score by addressing challenges like data imbalance and premature convergence, they also increase computational complexity, which may explain the slightly longer execution time.

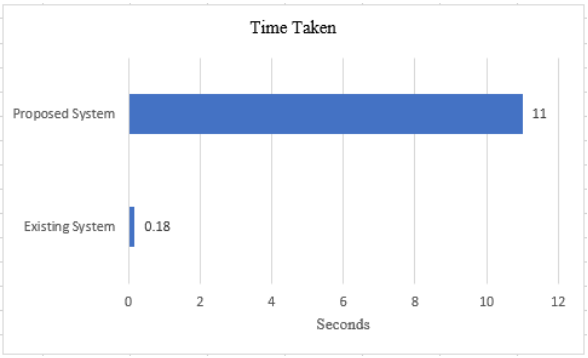

Figure 6: Evaluation of the Existing System vs. the Proposed System Using Subset of the Dataset: Execution Time Comparison

Conclusion and Future Works

This study presents a novel hybrid feature selection approach that integrates scramble and bit-flipping techniques within the mutation operator of a genetic algorithm (GA). The hybrid mutation operator effectively identifies relevant features for the classification process while eliminating redundant and irrelevant ones, thereby optimizing the feature selection process. By enhancing the conventional GA, the proposed method introduces greater diversity in the population of chromosomes across generations, addressing key limitations of GA, such as slow convergence and premature stagnation at suboptimal solutions. In this research, AdaBoost was employed as the classifier to tackle issues of data imbalance and overfitting. The dataset was partitioned into training and testing sets using K-fold cross-validation to ensure robust evaluation. The system's performance was measured using five critical metrics, showcasing its effectiveness and reliability. Beyond improving classification accuracy, this work has broader implications for strengthening the security infrastructure of Fog Computing environments. By enhancing intrusion detection systems to combat evolving cyber threats, the study contributes significant insights to the field of Fog Computing security. It underscores the value of integrating intelligent feature selection techniques to fortify distributed computing systems against sophisticated attacks.

For future research directions, the proposed model can be further explored using a wider variety of datasets to assess its generalizability. Additionally, experimenting with alternative classifiers beyond AdaBoost could reveal new performance benchmarks. Investigating advanced techniques for addressing data imbalance may also lead to a reduction in execution time, making the model more computationally efficient without compromising its effectiveness.

REFERENCES

Abbas, A., Khan, M. A., Latif, S., Ajaz, M., Shah, A. A., & Ahmad, J. (2022). A New Ensemble-Based Intrusion Detection System for Internet of Things. *Arabian Journal for Science and Engineering*, *47*(2), 1805–1819. https://doi.org/10.1007/S13369-021-06086-5/TABLES/12

Abu Alghanam, O., Almobaideen, W., Saadeh, M., & Adwan, O. (2023). An improved PIO feature selection algorithm for IoT network intrusion detection system based on ensemble learning. *Expert Systems with Applications*, *213*, 118745. https://doi.org/10.1016/J.ESWA.2022.118745

Ali, Z. A., Rasheed, S. A., & No, N. (2020). *An enhanced hybrid genetic algorithm for solving traveling salesman problem*. *18*(2), 1035–1039. https://doi.org/10.11591/ijeecs.v18.i2.pp1035-1039

Aliyu, F., Sheltami, T., Deriche, M., & Nasser, N. (2022). Human Immune-Based Intrusion Detection and Prevention System for Fog Computing. *Journal of Network and Systems Management*, *30*(1). https://doi.org/10.1007/S10922-021-09616-6

Atlam, H. F., Walters, R. J., & Wills, G. B. (2018). Fog Computing and the Internet of Things: A Review. *Big Data and Cognitive Computing 2018, Vol. 2, Page 10*, *2*(2), 10. https://doi.org/10.3390/BDCC2020010

Avinash, N., Sinha, S. K., & Shivamurthaiah, M. (2024). An Improved Gannet Optimization Algorithm Based on Opposition-Based Schemes for Feature Selection Problems in High-Dimensional Datasets. *SN Computer Science*, *5*(1), 1–15. https://doi.org/10.1007/S42979-023-02487-5/METRICS

Azad, C., Bhushan, B., Sharma, R., Shankar, A., Singh, K. K., & Khamparia, A. (2022). Prediction model using SMOTE, genetic algorithm and decision tree (PMSGD) for classification of diabetes mellitus. *Multimedia Systems*, *28*(4), 1289–1307. https://doi.org/10.1007/s00530-021-00817-2

Bhuvaneswari, B. A., & S., S. (2020). Anomaly detection framework for Internet of things traffic using vector convolutional deep learning approach in fog environment. *Future Generation Computer Systems*, *113*, 255–265. https://doi.org/10.1016/J.FUTURE.2020.07.020

Chakraborty, C., Kishor, A., & Rodrigues, J. J. P. C. (2022). Novel Enhanced-Grey Wolf Optimization hybrid machine learning technique for biomedical data computation. *Computers and Electrical Engineering*, *99*, 107778.

https://doi.org/10.1016/J.COMPELECENG.2022.107778

Chiba, Z., Abghour, N., Moussaid, K., Lifandali, O., & Kinta, R. (2022). A Deep Study of Novel Intrusion Detection Systems and Intrusion Prevention Systems for Internet of Things Networks. *Procedia Computer Science*, *210*(C), 94–103. https://doi.org/10.1016/j.procs.2022.10.124

Ding, Y., Zhou, K., & Bi, W. (2020). Feature selection based on hybridization of genetic algorithm and competitive swarm optimizer. *Soft Computing*, *24*(15), 11663–11672. https://doi.org/10.1007/s00500-019-04628-6

Dwivedi, S., Vardhan, M., & Tripathi, S. (2021). Building an efficient intrusion detection system using grasshopper optimization algorithm for anomaly detection. *Cluster Computing*, *24*(3), 1881–1900. https://doi.org/10.1007/S10586-020-03229-5/METRICS

Effrosynidis, D., & Arampatzis, A. (2021a). An evaluation of feature selection methods for environmental data. *Ecological Informatics*, *61*(December 2020), 101224. https://doi.org/10.1016/j.ecoinf.2021.101224

Effrosynidis, D., & Arampatzis, A. (2021b). An evaluation of feature selection methods for environmental data. *Ecological Informatics*, *61*, 101224. https://doi.org/10.1016/J.ECOINF.2021.101224

Freund, Y. (1995). Boosting a weak learning algorithm by majority. *Information and Computation*, *121*(2), 256–285. https://doi.org/10.1006/INCO.1995.1136

Gao, X., Shan, C., Hu, C., Niu, Z., & Liu, Z. (2019). An Adaptive Ensemble Machine Learning Model for Intrusion Detection. *IEEE Access*, *7*, 82512–82521. https://doi.org/10.1109/ACCESS.2019.2923640

Hancer, E., Xue, B., & Zhang, M. (2020). A survey on feature selection approaches for clustering. *Artificial Intelligence Review*, *53*(6), 4519–4545. https://doi.org/10.1007/S10462-019-09800-W/METRICS

Hassan, I. H., Mohammed, A., Ali, Y. S., Jeremiah, I., & Abdulraheem, S. A. (2023). Metaheuristic algorithms in text clustering. *Comprehensive Metaheuristics*, 131–152. https://doi.org/10.1016/B978-0-323-91781-0.00007-7

Hayatu, I., Abdullahi, M., Isuwa, J., Ali, S., & Tetengi, I. (2024). *Franklin Open A comprehensive survey of honey badger optimization algorithm and meta-analysis of its variants and applications*. *8*(July).

He, Y. L., Zhao, Y., Hu, X., Yan, X. N., Zhu, Q. X., & Xu, Y. (2020). Fault diagnosis using novel AdaBoost based discriminant locality preserving projection with resamples. *Engineering Applications of Artificial Intelligence*, *91*, 103631. https://doi.org/10.1016/J.ENGAPPAI.2020.103631

Isuwa, J., Abdullahi, M., & Abdulrahim, A. (2022). *Hybrid particle swarm optimization with sequential one point flipping algorithm for feature selection*. *July*, 1–18. https://doi.org/10.1002/cpe.7239

Isuwa, J., Abdullahi, M., Ali, Y. S., Kim, J., Hassan, I. H., & Buba, J. R. (2023). Optimizing Microarray Cancer Gene Selection using Swarm Intelligence : Recent Developments and An Exploratory Study Optimizing Microarray Cancer Gene Selection using Swarm Intelligence : Recent. *Egyptian Informatics Journal*, *24*(4), 100416. https://doi.org/10.1016/j.eij.2023.100416

Jemili, F., Meddeb, R., & Korbaa, O. (2024). Intrusion detection based on ensemble learning for big data classification. *Cluster Computing*, *27*(3), 3771–3798. https://doi.org/10.1007/S10586-023-04168-7

Jeremiah, I., Abdullahi, M., Yusuf, S. A., & Idris, M. N. (2022). *Integration of Specific Local Search Methods in Metaheuristic Algorithms for Optimizing the Feature Selection Process : A Survey*. *4*(1), 34–48.

Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, *80*(5), 8091–8126. https://doi.org/10.1007/s11042-020-10139-6

Kumar, P., Gupta, G. P., & Tripathi, R. (2021a). A distributed ensemble design based intrusion detection system using fog computing to protect the internet of things networks. *Journal of Ambient Intelligence and Humanized Computing*, *12*(10), 9555–9572. https://doi.org/10.1007/S12652-020-02696-3/METRICS

Kumar, P., Gupta, G. P., & Tripathi, R. (2021b). A distributed ensemble design based intrusion detection system using fog computing to protect the internet of things networks. *Journal of Ambient Intelligence and Humanized Computing*, *12*(10), 9555–9572. https://doi.org/10.1007/S12652-020-02696-3/METRICS

Laporte, G., & Osman, I. H. (1995). Routing problems: A bibliography. *Annals of Operations Research 1995 61:1*, *61*(1), 227–262. https://doi.org/10.1007/BF02098290

Liu, H., Zhang, X., & Zhang, X. (2019). PwAdaBoost: Possible world based AdaBoost algorithm for classifying uncertain data. *Knowledge-Based Systems*, *186*, 104930. https://doi.org/10.1016/J.KNOSYS.2019.104930

Long, Z., Zhang, X., Zhang, L., Qin, G., Huang, S., Song, D., Shao, H., & Wu, G. (2021a). Motor fault diagnosis using attention mechanism and improved adaboost driven by multi-sensor information. *Measurement*, *170*, 108718. https://doi.org/10.1016/J.MEASUREMENT.2020.108718

Long, Z., Zhang, X., Zhang, L., Qin, G., Huang, S., Song, D., Shao, H., & Wu, G. (2021b). Motor fault diagnosis using attention mechanism and improved adaboost driven by multi-sensor information. *Measurement: Journal of the International Measurement Confederation*, *170*, 108718. https://doi.org/10.1016/j.measurement.2020.108718

Michalewicz, Z., & Schoenauer, M. (1996). Evolutionary Algorithms for Constrained Parameter Optimization Problems. *Evolutionary Computation*, *4*(1), 1–32. https://doi.org/10.1162/EVCO.1996.4.1.1

Mishra, P., Varadharajan, V., Tupakula, U., & Pilli, E. S. (2019). A detailed investigation and analysis of using machine learning techniques for intrusion detection. *IEEE Communications Surveys and Tutorials*, *21*(1), 686–728. https://doi.org/10.1109/COMST.2018.2847722

Moh, M., & Raju, R. (2018). Machine learning techniques for security of internet of things (IoT) and fog computing systems. *Proceedings - 2018 International Conference on High Performance Computing and Simulation, HPCS 2018*, 709–715. https://doi.org/10.1109/HPCS.2018.00116

Mohamed, D., & Ismael, O. (2023). Enhancement of an IoT hybrid intrusion detection system based on fog-to-cloud computing. *Journal of Cloud Computing*, *12*(1), 1–13. https://doi.org/10.1186/S13677-023-00420-Y/TABLES/7

Moran, M., & Gordon, G. (2019). Curious Feature Selection. *Information Sciences*, *485*, 42–54. https://doi.org/10.1016/J.INS.2019.02.009

Nabi, F., & Zhou, X. (2024). Enhancing intrusion detection systems through dimensionality reduction: A comparative study of machine learning techniques for cyber security. *Cyber Security and Applications*, *2*, 100033. https://doi.org/10.1016/J.CSA.2023.100033

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

Paoletti, M. E., Mogollon-Gutierrez, O., Moreno-Alvarez, S., Sancho, J. C., & Haut, J. M. (2023). A Comprehensive Survey of Imbalance Correction Techniques for Hyperspectral Data Classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *16*, 5297–5314. https://doi.org/10.1109/JSTARS.2023.3279506

Peng, K., Leung, V. C. M., Zheng, L., Wang, S., Huang, C., & Lin, T. (2018). Intrusion Detection System Based on Decision Tree over Big Data in Fog Environment. *Wireless Communications and Mobile Computing*, *2018*(1), 4680867. https://doi.org/10.1155/2018/4680867

Petinrin, O. O., Saeed, F., Salim, N., Toseef, M., Liu, Z., & Muyide, I. O. (2023). Dimension Reduction and Classifier-Based Feature Selection for Oversampled Gene Expression Data and Cancer Classification. *Processes*, *11*(7), 1–13. https://doi.org/10.3390/pr11071940

Pham, B. T., Nguyen, M. D., Nguyen-Thoi, T., Ho, L. S., Koopialipoor, M., Kim Quoc, N., Armaghani, D. J., & Le, H. Van. (2021). A novel approach for classification of soils based on laboratory tests using Adaboost, Tree and ANN modeling. *Transportation Geotechnics*, *27*, 100508. https://doi.org/10.1016/J.TRGEO.2020.100508

Rajadurai, H., & Gandhi, U. D. (2022). A stacked ensemble learning model for intrusion detection in wireless network. *Neural Computing and Applications*, *34*(18), 15387–15395. https://doi.org/10.1007/S00521-020-04986-5/METRICS

Sadaf, K., & Sultana, J. (2020). Intrusion detection based on autoencoder and isolation forest in fog computing. *IEEE Access*, *8*, 167059–167068. https://doi.org/10.1109/ACCESS.2020.3022855

Sahebi, G., Movahedi, P., Ebrahimi, M., Pahikkala, T., & Plosila, J. (2020). GeFeS : A generalized wrapper feature selection approach for optimizing classification performance. *Computers in Biology and Medicine*, *125*(August), 103974. https://doi.org/10.1016/j.compbiomed.2020.103974

Shallangwa, Y., Ahmad, A. A., Isuwa, J., & Yahaya, E. B. (2024). SWARM INTELLIGENT OPTIMIZATION ALGORITHMS FOR PRECISION GENE SELECTION IN MICROARRAY-BASED CANCER CLASSIFICATION. *World Science Journal*, *19*(3), 842–854. https://doi.org/https://dx.doi.org/10.4314/swj.v19i3.32

Sharma, D., Willy, C., & Bischoff, J. (2021). Optimal subset selection for causal inference using machine learning ensembles and particle swarm optimization. *Complex & Intelligent Systems*, *7*(1), 41–59. https://doi.org/10.1007/s40747-020-00169-w

Solorio-Fernández, S., Carrasco-Ochoa, J. A., & Martínez-Trinidad, J. F. (2020). A review of unsupervised feature selection methods. *Artificial Intelligence Review*, *53*(2), 907–948. https://doi.org/10.1007/S10462-019-09682-Y/METRICS

Tair, M., Bacanin, N., Zivkovic, M., & Venkatachalam, K. (2022). A Chaotic Oppositional Whale Optimisation Algorithm with Firefly Search for Medical Diagnostics. *Computers, Materials and Continua*, *72*(1), 959–982. https://doi.org/10.32604/cmc.2022.024989

Tang, D., Tang, L., Dai, R., Chen, J., Li, X., & Rodrigues, J. J. P. C. (2020). MF-Adaboost: LDoS attack detection based on multi-features and improved Adaboost. *Future Generation Computer Systems*, *106*, 347–359. https://doi.org/10.1016/J.FUTURE.2019.12.034

Thabtah, F., Hammoud, S., Kamalov, F., & Gonsalves, A. (2020). Data imbalance in classification: Experimental evaluation. *Information Sciences*, *513*, 429–441. https://doi.org/10.1016/J.INS.2019.11.004

Tsai, C. (2002). *A New Approach for Solving Large Traveling Salesman Problem Using Evolutionary Ant Rules*. 00*, 1540–1545.

Tubishat, M., Ja'afar, S., Alswaitti, M., Mirjalili, S., Idris, N., Ismail, M. A., & Omar, M. S. (2021). Dynamic Salp swarm algorithm for feature selection. *Expert Systems with Applications*, *164*. https://doi.org/10.1016/j.eswa.2020.113873

Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep Learning Approach for Intelligent Intrusion Detection System. *IEEE Access*, *7*, 41525–41550. https://doi.org/10.1109/ACCESS.2019.2895334

Wang, Y., Shen, Y., & Zhang, G. (2016). Research on Intrusion Detection Model using ensemble learning methods. *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS*, *0*, 422–425. https://doi.org/10.1109/ICSESS.2016.7883100

Yuan, Y., Kaklamanos, G., & Hogrefe, D. (2016). A novel semi-supervised adaboost technique for network anomaly detection. *MSWiM 2016 - Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 111–114. https://doi.org/10.1145/2988287.2989177

Zharmagambetov, A., Gabidolla, M., & Carreira-Perpiñán, M. (2021). IMPROVED MULTICLASS ADABOOST FOR IMAGE CLASSIFICATION: THE ROLE OF TREE OPTIMIZATION. *Proceedings - International Conference on Image Processing, ICIP*, *2021-September*, 424–428. https://doi.org/10.1109/ICIP42928.2021.9506569