IJEIT
International Journal of Emerging Information Technology
ISSN: XXXX-XXXX / Volume 1 Issue 1 September-October 2025 / Page No: 1-16
Paper Id: IJEIT-V1I1P101

Original Article

Enhancing the Performance of Intrusion Detection Systems using Hybrid Bio-Inspired optimization Algorithms

Anshul Sharma[1], S. Indra Priyadharshani[2]

VIT Chennai, Chennai, Tamil Nadu, India.

Corresponding Author : anshul.sharma2023@vitstudent.ac.in

Received: 09 June 2025 Revised: 20 July 2025 Accepted: 04 September 2025 Published: 20 September 2025

Abstract *- This research presents a novel hybrid feature selection method, integrating filter-based feature selection using mutual information and ANOVA F-value with the Grey Wolf Optimizer (GWO) and Particle Swarm Optimization (PSO), to enhance the accuracy and efficiency of classification tasks in network intrusion detection systems. Leveraging the UNSW-NB15 dataset, we evaluate four machine learning models—Random Forest, Decision Tree, Logistic Regression, and XGBoost—both before and after feature selection. The optimized feature sets show substantial improvements in performance metrics across models. Random Forest achieved a test accuracy of 95.28%, XGBoost reached 94.84%, Decision Tree recorded 94.01%, and Logistic Regression improved to 90.19% after feature selection. These findings underscore the efficacy of combining filter-based methods with hybrid optimization techniques for feature selection, effectively boosting model accuracy and computational efficiency. This study paves the way for further exploration of hybrid approaches and alternative classifiers to enhance intrusion detection systems.*

Keywords *- Intrusion detection systems, feature selection, filter-based feature selection, Particle Swarm Optimizer, Grey Wolf Optimizer, Hybrid GWOPSO.*

1. Introduction

The rapid advancement of network technologies and the increasing sophistication of cyber threats highlight the critical need for effective Intrusion Detection Systems (IDS). These systems are essential for monitoring network traffic to identify and mitigate malicious activities, thus providing a safeguard against a broad spectrum of security threats. Despite significant progress in IDS methodologies, many existing systems still encounter challenges such as high false positive rates, overfitting, and computational inefficiencies. These issues can compromise the effectiveness of IDS, particularly in dynamic network environments characterized by growing data volumes and complexities.

Feature selection is a vital component in enhancing IDS performance, as it directly affects the accuracy and speed of anomaly detection. Traditional feature selection techniques often rely on single optimization methods, which may not adequately balance exploration and exploitation. This imbalance can result in suboptimal performance in identifying genuine threats while minimizing false alarms. Therefore, innovative solutions are required to improve feature selection processes, leading to more accurate and efficient IDS.

Recent research has explored various optimization algorithms for feature selection in IDS, including Genetic Algorithms (GA), Ant Colony Optimization (ACO), Grey Wolf Optimization (GWO), and Particle Swarm Optimization (PSO). Bio-inspired optimization techniques, particularly GWO and PSO, have gained traction due to their ability to mimic natural processes to solve complex optimization problems. GWO is noted for its strength in exploitation, while PSO excels in exploration, making their integration particularly promising.

In this research, we address the challenges in intrusion detection by proposing a hybrid feature selection algorithm that combines GWO and PSO. By leveraging the strengths of both algorithms, our approach aims to develop an optimal feature set that enhances IDS performance and accurately represents the underlying data. Our key objectives include improving detection capabilities, minimizing detection time, reducing false positive rates, and increasing true positive detection rates to ensure reliable identification of genuine intrusions. To validate the effectiveness of the proposed method, we utilize the UNSW-NB15 benchmark dataset, which offers a diverse range of network traffic features and various malicious attacks, providing a realistic environment for evaluating our hybrid GWO-PSO approach. Additionally, we employ advanced machine and ensemble learning models, including XGBoost and Random Forest, to demonstrate notable enhancements in detection accuracy and overall system performance. Ultimately, this research aims to make significant contributions to the evolution of IDS in response to the growing complexity of emerging cyber threats.

2. Literature Review

Recent advancements in Intrusion Detection Systems (IDS) have focused on enhancing detection accuracy and efficiency through innovative feature selection techniques and the application of machine learning algorithms.

Harbi et al. [1] introduced an IDS tailored for IoT networks, which utilizes a decision tree classifier optimized by a modified firefly algorithm for feature selection. By applying this method to the Edge-IIoTset dataset, which contains 61 features related to various IoT attacks, they reduced the feature set from 74 to 39. This reduction improved detection accuracy from 69.88% to 79.64% while decreasing detection time by 49%, illustrating the effectiveness of bio-inspired algorithms in optimizing IDS for resource-limited IoT environments.

Manokaran and Vairavel [8] presented a deep learning-based IDS combining Autoencoder-Long Short-Term Memory (AE-LSTM) with an Improved Grey Wolf Optimization (IGWO) algorithm, specifically designed for IoT edge computing. Their system, DL-ADS, used a novel testbed dataset generated from Raspberry Pi devices to simulate various attack scenarios. IGWO optimized LSTM hyperparameters, incorporating mechanisms such as Elimination and Opposition-Based Learning to avoid local optima. The model delivered impressive accuracy across multiple datasets: 99.11% on their testbed, 99.85% on CICIDS 2017, 99.47% on DS2OS, and 99.66% on MQTTset, demonstrating the potential of combining deep learning with bio-inspired optimization algorithms for enhanced IoT security.

Keserwani et al. [13] proposed a Smart Anomaly-Based IDS for IoT networks that integrates Grey Wolf Optimization (GWO) and Particle Swarm Optimization (PSO) for feature selection. Their hybrid GWO-PSO-RF model leverages GWO's exploration capabilities alongside PSO's quick convergence, subsequently classifying the selected features using a Random Forest algorithm. Their model achieved an impressive average accuracy of 99.66% for multiclass classification across benchmark datasets like KDDCup99, NSL-KDD, and CICIDS-2017, highlighting the model's robustness and versatility in diverse IoT applications.

Saheed et al. [2] developed a hybrid feature selection method, combining the Bat algorithm with the Residue Number System (RNS) for IDS. This model employed principal component analysis (PCA) for feature extraction and used Naïve Bayes and k-Nearest Neighbors (k-NN) for classification. Applied to the NSL-KDD and UNSW-NB15 datasets, the model achieved a detection accuracy of 99.48%, while significantly improving processing speed, making it twice as efficient compared to standalone feature selection techniques.

Otair et al. [3] devised an enhanced hybrid algorithm combining Grey Wolf Optimizer (GWO) with Particle Swarm Optimization (PSO) for IDS in Wireless Sensor Networks. Their GWO-PSO approach optimized feature selection by updating the position of grey wolves based on PSO, resulting in a detection rate of 95.6% when tested with a Support Vector Machine (SVM) on the NSL-KDD dataset. This hybrid method showcased considerable improvements in both detection accuracy and processing speed.

Vijayanand and Devaraj [5] proposed a feature selection method for IDS in Wireless Mesh Networks, utilizing Whale Optimization Algorithm (WOA) enhanced with genetic operators like crossover and mutation to improve search space exploration. Tested on CICIDS2017 and ADFA-LD datasets, their method achieved over 99.2% accuracy in detecting attacks, particularly excelling in the identification of DoS and reconnaissance attacks.

Ali et al. [4] proposed a Particle Swarm Optimization (PSO)-optimized Fast Learning Network (FLN) for network intrusion detection. By using PSO to optimize the FLN's weights and biases, this model avoided overfitting and suboptimal weight selection. Evaluated on the KDD99 dataset, the approach achieved 98.5% accuracy, maintaining stability even with a reduced number of neurons in the hidden layers.

Hajimirzaei and Navimipour [6] combined a multilayer perceptron (MLP) network with artificial bee colony (ABC) optimization and fuzzy clustering algorithms for cloud computing IDS. Their model used fuzzy clustering to divide the dataset into homogeneous subsets, enhancing MLP training efficiency. ABC optimization then fine-tuned the MLP's weights. Tested on the NSL-KDD dataset, this approach improved classification accuracy by 2.23%, with superior performance in reducing mean absolute error (MAE) and root mean square error (RMSE).

Eesa et al. [7] introduced a feature selection approach utilizing the Cuttlefish Optimization Algorithm (CFA) for IDS. CFA, inspired by cuttlefish color-changing mechanisms, was used to select optimal feature subsets, with a Decision Tree classifier (ID3 algorithm) assessing the chosen features. Using only 10 features from the KDD Cup 99 dataset, the model achieved a detection rate of 92.05% and an accuracy of 92.83%, surpassing models using all 41 features.

Mohammad et al. [9] combined the Bat algorithm with the Residue Number System for feature selection in IDS. This hybrid approach achieved real-time intrusion detection by halving processing speed while maintaining high detection accuracy. The method demonstrated its suitability for real-time applications.

Bakro et al. [10] developed a cloud-based IDS using hybrid bio-inspired feature selection algorithms combined with a Random Forest model. Their system incorporated both Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) for feature selection, achieving an accuracy of 98.45% on NSL-KDD and CICIDS2017 datasets, with reduced computational time suitable for real-time cloud environments.

Mohammad [11] introduced a hybrid feature selection model that combines Grey Wolf Optimization (GWO) and Particle Swarm Optimization (PSO). Two variants of the model, PSO-GWO-NB (Naïve Bayes) and PSO-GWO-ANN (Artificial Neural Networks), were tested on the UNSW-NB15 dataset. The PSO-GWO-NB variant achieved a true positive rate of 90.4% and a false positive rate of 8.8%, outperforming alternative configurations.

Finally, Moghanian, S et al. [12] proposed the GOAMLP, a Network Intrusion Detection System combining the Grasshopper Optimization Algorithm (GOA) with a Multilayer Perceptron (MLP). GOA optimized the MLP's weights and biases, reducing classification errors. When evaluated on NSL-KDD and CICIDS2017 datasets, the system achieved over 99% accuracy, particularly excelling in the detection of DoS and probe attacks.

Network Intrusion Detection Systems (NIDS) have been a crucial component in maintaining network security, especially in detecting potential threats from complex and voluminous network traffic. Traditional NIDS models often face challenges such as high false positive rates and the increasing scale of network data. To address these issues, Shone et al. [14] proposed a novel deep learning approach that combines the strengths of both deep and shallow learning techniques. Their methodology introduces a Non-Symmetric Deep Auto-Encoder (NDAE) for unsupervised feature learning, followed by a stacked NDAE and a Random Forest (RF) classifier. This hybrid model, implemented in TensorFlow with GPU acceleration, demonstrated notable improvements in performance, achieving 94.2% accuracy on the KDD Cup '99 dataset and 83.28% accuracy on the NSL-KDD dataset. Additionally, the model significantly reduced the false positive rate to 2.1% while maintaining high detection rates and reduced training time, outperforming existing models like Deep Belief Networks (DBNs).

In the domain of Wireless Sensor Networks (WSNs), Safaldin et al. [15] presented an optimized intrusion detection system (IDS) by combining the Binary Gray Wolf Optimizer (GWO) with a Support Vector Machine (SVM) classifier. Their approach focused on enhancing detection accuracy and reducing false alarm rates in resource-constrained environments. By utilizing GWO with 3, 5, and 7 wolves, the authors optimized the feature selection process, with the 7-wolf GWOSVM-IDS achieving a remarkable accuracy of 99.65% on the NSL-KDD dataset, surpassing Particle Swarm Optimization-based IDS (PSO-IDS). This model also exhibited a significantly reduced false alarm rate of 1.2% and a detection rate of 97.8%, making it highly suitable for real-time applications.

Ferrag et al. [16] conducted a comprehensive comparative study of deep learning models for intrusion detection, evaluating models such as Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Deep Autoencoders. Tested across two datasets, CSE-CIC-IDS2018 and Bot-IoT, the CNN-based model achieved the highest accuracy of 99.12% on the Bot-IoT dataset and 97.65% on the CSE-CIC-IDS2018 dataset for binary classification tasks. The RNN model also performed well, reaching 98.9% accuracy in multiclass classification with a low false alarm rate of 1.15%. Comparatively, these deep learning models outperformed traditional machine learning approaches such as Random Forest and SVM, which achieved accuracies of 94.5% and 92.3%, respectively.

These studies collectively highlight the effectiveness of bio-inspired optimization algorithms, advanced machine learning techniques, and hybrid approaches in enhancing the performance of Intrusion Detection Systems (IDS). Key themes that emerge include the use of nature-inspired algorithms like the Binary Gray Wolf Optimizer (GWO) for feature selection, which enhances detection accuracy and reduces false alarm rates, especially in resource-constrained environments like Wireless Sensor Networks (WSNs). Additionally, combining multiple algorithms, such as deep learning models with traditional machine learning approaches, leverages their respective strengths to improve accuracy and reduce computational complexity. Deep learning models, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have demonstrated superior performance in handling complex, real-world network traffic, significantly outperforming traditional models such as Support Vector Machines (SVM) and Random Forest.

Table 1. Literature review

Year	Title and Author(s)	Proposed Methodologies	Dataset(s) used	Findings and Results
2024	Bio-inspired Intrusion Detection System for Internet of Things Networks Security (Harbi et al.)	Decision tree for classification, modified firefly algorithm for feature selection	Edge-IIoTset	79.64% detection accuracy after feature selection, 49% reduction in detection time
2024	Feature Selection in Intrusion Detection Systems: A New Hybrid Fusion of Bat Algorithm and Residue Number System (Saheed et al.)	Bat algorithm for feature selection, Residue Number System for dimensionality reduction, PCA for feature extraction, Naïve Bayes	NSL-KDD, UNSW-NB15	99.48% overall detection accuracy, 98.23% detection rate, improved F-scores, doubled processing speed

		and k-NN for classification		
2024	Building a Cloud-IDS by Hybrid Bio-Inspired Feature Selection Algorithms Along with Random Forest Model (Bakro et al.)	Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) for feature selection, Random Forest for classification	NSL-KDD, CICIDS2017	98.45% accuracy, reduced computational time
2021	Intrusion Detection Using a New Hybrid Feature Selection Model (Hamdan)	Grey Wolf Optimization (GWO) and Particle Swarm Optimization (PSO) for feature selection, Naïve Bayes and ANN for classification	UNSW-NB15	PSO-GWO-NB outperformed PSO-GWO-ANN in terms of precision and recall, TPR of 90.4%, FPR of 8.8%
2022	An Enhanced Grey Wolf Optimizer Based Particle Swarm Optimizer for Intrusion Detection System in Wireless Sensor Networks (Otair et al.)	GWO-PSO hybrid for feature selection, K-means and SVM for classification	NSL-KDD	Improved detection accuracy and processing speed, accuracy of 94.8%, detection rate of 95.6% with SVM
2018	A New Intrusion Detection System Based on Fast Learning Network and Particle Swarm Optimization (Ali et al.)	PSO-optimized FLN for classification	KDD99	98.5% accuracy, outperformed other optimization techniques (GA, HSO, ATLBO)
2020	A Novel Feature Selection Method Using Whale Optimization Algorithm and Genetic Operators for Intrusion Detection System in Wireless Mesh Network (Vijayanand et al.)	WOA with genetic operators for feature selection, SVM for classification	CICIDS2017, ADFA-LD	99.2% accuracy on CICIDS2017, lower FPR compared to conventional methods
2020	GOAMLP: Network Intrusion Detection With Multilayer Perceptron and Grasshopper Optimization Algorithm	GOA for optimizing MLP parameters	NSL-KDD, CICIDS2017	Over 99% accuracy on NSL-KDD, reduced false positives
2019	Intrusion Detection for Cloud Computing using Neural Networks and Artificial Bee Colony Optimization Algorithm (Hajimirzaei et al.)	MLP with ABC optimization and fuzzy clustering	NSL-KDD	2.23% improvement in correctly classified instances, superior performance in reducing MAE and RMSE
2015	A Novel Feature-Selection Approach Based on the Cuttlefish Optimization Algorithm for Intrusion Detection Systems (Eesa et al.)	CFA for feature selection, ID3 for classification	KDD Cup 99	92.05% detection rate with 10 selected features, reduced FPR to 3.9%
2024	DL-ADS: Improved Grey Wolf Optimization Enabled AE-LSTM Technique for Efficient Network Anomaly Detection in Internet of Things (IoT) Edge Computing (Manokaran et al.)	AE-LSTM with IGWO	Testbed dataset, CICIDS 2017, DS2OS, MQTTset	99.11% accuracy on testbed, lower FAR, improved precision, recall, and F1-score
2021	A Smart Anomaly-Based Intrusion Detection System for the Internet of Things (IoT) Network Using GWO–PSO–RF Model (Keserwani et al.)	GWO-PSO for feature selection, RF for classification	KDDCup99, NSL-KDD, CICIDS-2017	99.66% average accuracy, outperformed other models in terms of accuracy and detection time
2020	A Feature Selection Model for Network Intrusion Detection System Based on PSO, GWO, FFA, and GA Algorithms (Almomani)	PSO, GWO, FFA, and GA for feature selection, SVM and J48 for classification	UNSW-NB15	90.48% accuracy with R12 rule, 90.12% accuracy with R13 rule
2018	A Deep Learning Approach to Network Intrusion Detection (Shone et al.)	Non-Symmetric Deep Auto-Encoder (NDAE) for feature learning, RF for classification	KDD Cup '99, NSL-KDD	94.2% accuracy on KDD Cup '99, 83.28% accuracy on NSL-KDD, reduced false positive rate to 2.1%, outperformed Deep Belief Networks

2021	Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks (Safaldin et al.)	Binary Gray Wolf Optimizer (GWO) for feature selection, SVM for classification	NSL-KDD	99.65% accuracy with 7-wolf GWOSVM-IDS, reduced false alarm rate to 1.2%, outperformed PSO-IDS
2020	Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study (Ferrag et al.)	Comparative study of CNN, RNN, and deep autoencoders	CSE-CIC-IDS2018, Bot-IoT	CNN achieved 99.12% on Bot-IoT, 97.65% on CSE-CIC-IDS2018, RNN reached 98.9% accuracy, deep learning outperformed traditional methods like Random Forest (94.5%) and SVM (92.3%)

3. Methodology

In this research, we focus on enhancing the performance of classification tasks through a novel feature selection approach utilizing a Grey Wolf Optimizer and Particle Swarm Optimization (GWO-PSO) hybrid technique. Feature selection plays a crucial role in reducing dimensionality, improving model accuracy, and accelerating computational efficiency. By identifying the most relevant features in the dataset, we aim to achieve better classification performance across various models, thereby facilitating more effective anomaly detection.

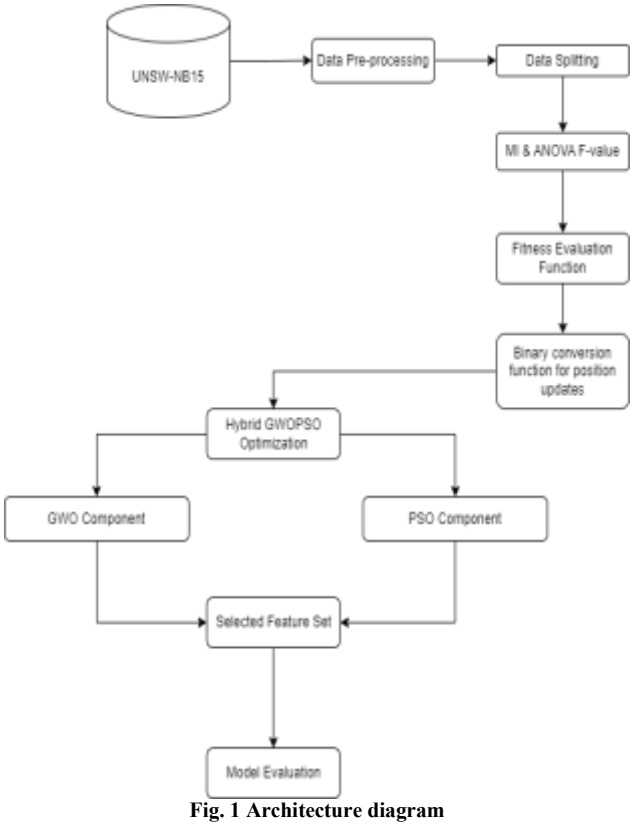

Fig. 1 Architecture diagram

3.1. Dataset

The dataset used in this research is UNSW-NB15, contains the network traffic data collected from a simulated environment, which is comprehensive and specifically designed for intrusion detection systems. The dataset comprises 257,673 instances with 45 features, including both continuous and categorical variables, that represent various aspects of network traffic. It includes nine types of attacks including Fuzzers, Analysis, Backdoors and Denial of Service (DoS), Worms, etc.

The dataset is divided into a training set with 175,341 records and a testing set containing 82,332 records, providing a solid foundation for evaluating and developing intrusion detection systems (IDS) in our research. The target variable, label, indicates whether a given instance is a normal connection (0) or an attack (1).

In addition to the primary features, the dataset encompasses categorical attributes such as proto, service, and state, which further enrich the data and provide insights into network behaviours.

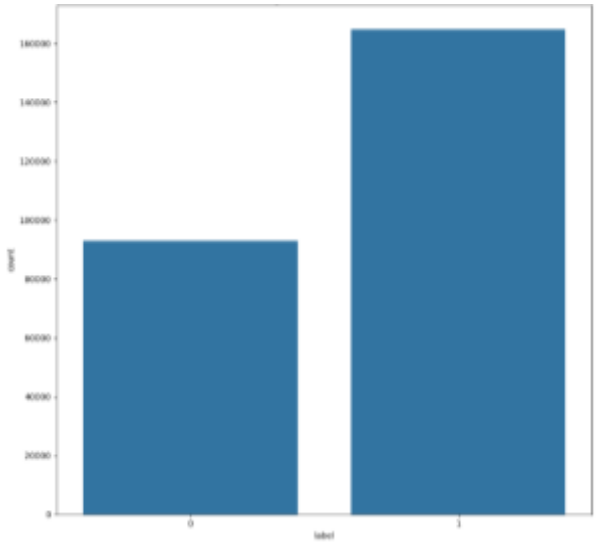

Fig. 2 Distribution of target variable

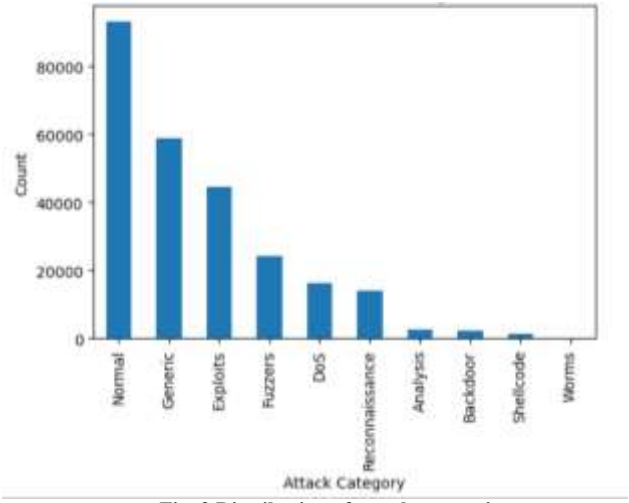

Fig. 3 Distribution of attack categories

a) Data Preprocessing

The preprocessing phase is crucial in preparing the dataset for effective machine learning model training. In this study, the UNSW-NB15 dataset is utilized, consisting of numerous features relevant to intrusion detection. The preprocessing steps are outlined as follows:

- Data Selection: The initial step involves dropping unnecessary columns from the dataset, specifically the *'id', 'attack_cat', and 'label'* columns, as they do not contribute to the feature set for training the model. The features are defined as $\bar{X}$, while the target variable (labels) is denoted as $\bar{y}$.
- Categorical and Numerical Separation: The dataset is divided into numerical and categorical features. Numerical features are selected based on their data type, while categorical features are identified for further processing.
- Encoding Categorical Variables: To convert categorical variables into numerical format, each categorical feature is transformed into category codes. This step is necessary to allow the machine learning algorithms to process the categorical data.

b) Data Splitting

The dataset is split into training and testing sets to evaluate the model's performance. The training set consists of 80% of the data, while the remaining 20% is used for testing. Stratified splitting was applied to ensure that the class distribution in both the training and test sets remains consistent with the original dataset, especially important in cases of imbalanced data.

c) Feature Selection

Feature selection is essential for enhancing machine learning model performance by reducing dimensionality, increasing accuracy, and minimizing overfitting. This study incorporates a filter-based approach alongside meta-heuristic optimization algorithms—Grey Wolf Optimizer (GWO) and Particle Swarm Optimization (PSO)—for feature selection. This combined approach was applied sequentially to identify the most relevant features (MRF) for effective model training.

i) Filter-Based Approach

To initially reduce the feature space, we applied filter-based feature selection techniques, specifically Mutual Information and ANOVA F-value. These statistical methods evaluate each feature's relevance to the target variable independently:

- Mutual Information (MI): MI quantifies the amount of information shared between individual features and the target variable. By measuring the dependency between features and the target, MI identifies features that contribute most to reducing uncertainty in predictions, selecting those with higher MI scores for further analysis.
- ANOVA F-value: The ANOVA F-value evaluates each feature by calculating the variance between different groups (or classes) relative to the variance within each group. Features with high F-values indicate strong discrimination between classes, making them valuable for classification tasks.

The filter-based approach provides a preliminary subset of features, which are then optimized through GWO and PSO.

ii) Grey Wolf Optimizer

The Grey Wolf Optimizer (GWO) is a meta-heuristic algorithm inspired by the social hierarchy and hunting behavior of grey wolves. In GWO, the best solution is modeled as the "alpha," the second and third best solutions as the "beta" and "delta," respectively, and the rest are the "omega" wolves. GWO iteratively updates the positions of these wolves based on their distance from the alpha, beta, and delta solutions.

iii) Particle Swarm Optimization

Particle Swarm Optimization (PSO) is another meta-heuristic algorithm used for feature selection. PSO simulates the social behavior of particles moving in a multi-dimensional space. In this case, each particle represents a potential feature subset. Particles adjust their positions in the search space by considering both their own best-known position and the best-known positions of neighboring particles.

By using these algorithms, we successfully identified a total of 20 features that collectively contribute to enhanced model performance. The selected features included vital attributes related to the network traffic, enabling the classification algorithms to make informed predictions.

d) Feature Scaling

Feature scaling is applied to standardize the numerical features in the dataset, ensuring that they contribute equally to model performance. The StandardScaler is used, which transforms the data by subtracting the mean and dividing by the standard deviation for each feature, resulting in a standardized dataset with a mean of zero and a variance of one. This ensures that features with different ranges are treated equally, preventing models like XGBoost from being biased towards higher-magnitude features.

e) Classification

In this study, four prominent classifiers were utilized to evaluate model performance:

i) Random Forest

Random Forest is an ensemble learning method that constructs multiple decision trees during the training phase, producing a final classification result based on the majority vote from all trees. This technique is particularly robust against overfitting due to its averaging of predictions, making it ideal for high-dimensional datasets. Key advantages of Random Forest include:

- Ensemble Approach: By aggregating outputs from various trees, Random Forest minimizes variance and enhances generalization.
- Feature Importance: The model provides an assessment of feature importance, helping to identify the most impactful predictors.
- Immunization to Noise: It effectively handles noisy data and outliers, ensuring robust performance across diverse datasets.

ii) Decision Tree

The Decision Tree classifier symbolizes a fundamental approach in machine learning. It builds a tree-like model to make decisions by splitting the data based on feature values, ultimately leading to a final prediction. Notable characteristics include:

- Interpretability: The tree structure offers a clear representation of decision-making processes, which helps in understanding how predictions are made.

- Flexibility: It can easily handle both numerical and categorical data without requiring extensive data preprocessing.
- Overfitting Risk: While decision trees are simple and easy to understand, they can be prone to overfitting, especially with complex datasets.

iii) Logistic Regression

Logistic Regression is a statistical model used for binary classification that employs the logistic function to model the probability of a binary target variable. Its strengths include:

- Simplicity and Efficiency: Logistic Regression is computationally efficient and can be easily interpreted, making it a popular choice for binary outcomes.
- Probabilistic Output: It provides probabilities for class membership, allowing for a nuanced understanding of the predictions.
- Regularization: Techniques such as L1 (Lasso) and L2 (Ridge) regularization can be applied to prevent overfitting, especially in high-dimensional feature spaces.

iv) XGBoost (Extreme Gradient Boosting)

XGBoost is an advanced ensemble technique based on gradient boosting, which builds trees sequentially, with each new tree aimed at correcting the errors of the previous ones. Its advantages include:

- Performance: XGBoost is renowned for its predictive power due to its sequential learning process, effectively improving model accuracy.
- Egularization Methods: Built-in regularization helps prevent overfitting, allowing for better generalization on unseen data.
- Efficiency & Scalability: Optimized for speed, XGBoost leverages parallel processing, making it suitable for both small and large datasets.

Each classifier's performance is evaluated based on various metrics, including accuracy, precision, recall, F1 score, training time, and detection time, AUC-ROC, etc. These metrics provide a comprehensive assessment of the classifiers' effectiveness in accurately predicting outcomes within the context of the dataset.

4. Results

We evaluate machine learning models, including Random Forest, Decision Tree, Logistic Regression, and XGBoost, after feature refinement. Using metrics like accuracy, precision, recall, and F1 score, we analyze how selected features influence model performance. This study highlights the importance of optimization in enhancing predictive accuracy, showcasing the effectiveness of machine learning in real-world network applications.

Table 2. Hybrid GWOPSO with Filter-based Feature Selection

Selection Method	Selected Features	Features
PSO	25	proto, service, spkts, dpkts, sbytes, dbytes, sttl, dttl, dload, sloss, dloss, dwin, tcprtt, smean, trans_depth, response_body_len, ct_srv_src, ct_state_ttl, ct_dst_ltm, ct_src_dport_ltm, ct_dst_sport_ltm, ct_dst_src_ltm, ct_ftp_cmd, ct_srv_dst, is_sm_ips_ports
GWO	24	proto, service, spkts, sbytes, dbytes, dttl, sload, sloss, dloss, swin, dtcpb, dwin, tcprtt, ackdat, smean, dmean, ct_srv_src, ct_state_ttl, ct_src_dport_ltm, ct_dst_sport_ltm, ct_dst_src_ltm, ct_ftp_cmd, ct_src_ltm, ct_srv_dst
Final Feature set (Filter based Hybrid GWOPSO)	25	dur, proto, state, spkts, dpkts, dbytes, rate, sttl, dttl, sloss, sinpkt, dinpkt, sjit, djit, dwin, tcprtt, synack, ackdat, dmean, ct_srv_src, ct_src_dport_ltm, ct_dst_sport_ltm, ct_dst_src_ltm, ct_src_ltm, ct_srv_dst

The Hybrid GWOPSO approach, incorporating filter-based selection, produced a final feature set of 25 key attributes that significantly enhanced the performance of the intrusion detection models. Compared to the original full feature set, models trained on this optimized subset demonstrated improved accuracy and computational efficiency.

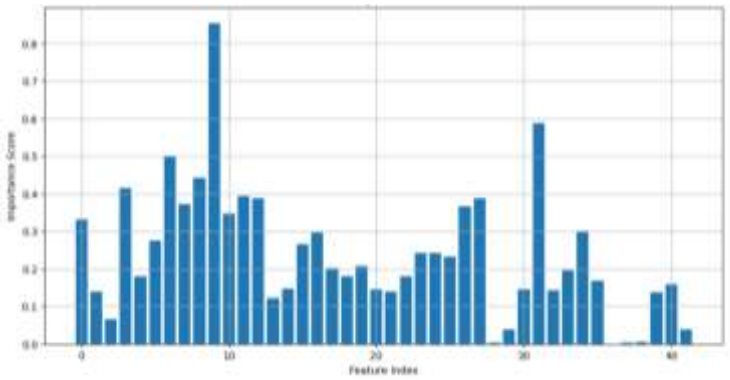

Fig. 4 Feature importance scores

Fig. 4 illustrates the feature importance scores obtained during the feature selection phase of the model. The bar graph represents the importance of each feature (indexed along the x-axis) as determined by the feature selection method. A distinct peak around feature index 10 highlights that this feature significantly contributes to the model's performance, with a high importance score exceeding 0.8. Other features also exhibit varying degrees of importance, with notable contributions around indices 5, 20, and 30. This analysis demonstrates that a subset of features is more relevant, aiding in dimensionality reduction without compromising model accuracy.

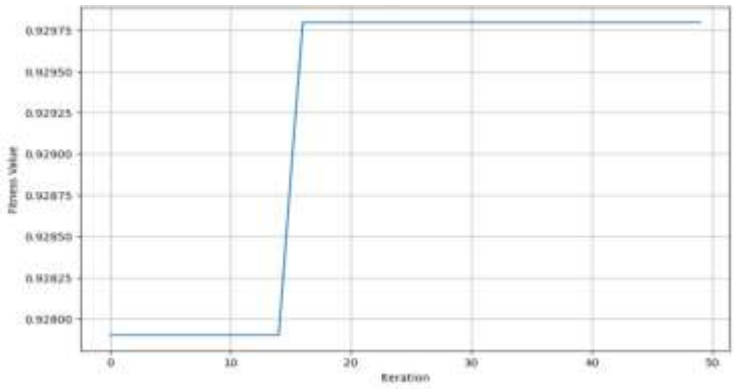

Fig. 5 Convergence Curve

Fig. 5 shows the convergence curve of the optimization algorithm used for feature selection. The curve plots the fitness value against the number of iterations. Initially, the fitness value exhibits minimal improvement but begins to rise significantly around the 10th iteration. By the 20th iteration, the algorithm reaches a near-optimal solution, where the fitness value stabilizes at approximately 0.92975. This rapid convergence indicates the algorithm's efficiency in finding an optimal subset of features within a relatively small number of iterations, thereby reducing computational complexity.

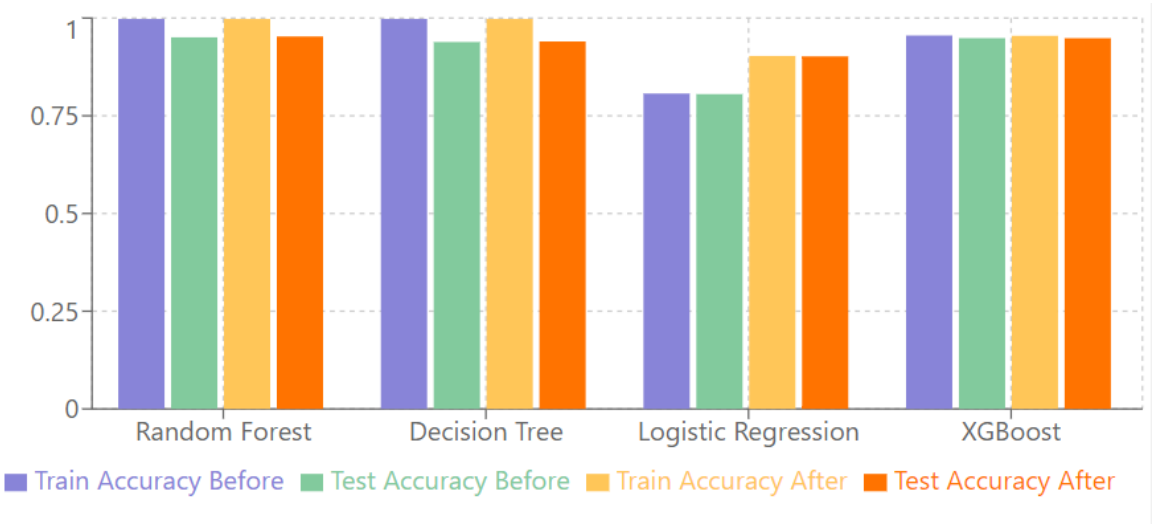

Fig. 6 Accuracy comparison

Table 3. Accuracy comparison

Model	Train Accuracy	Test Accuracy
Random Forest	99.76% → 99.76% (+0%)	95.06% → 95.28% (+0.23%)
Decision Tree	99.76% → 99.76% (+0%)	93.87% → 94.01% (+0.15%)
Logistic Regression	80.69% → 90.30% (+11.92%)	80.53% → 90.19% (+12.01%)
XGBoost	95.53% → 95.46% (-0.07%)	94.86% → 94.84% (-0.02%)

The accuracy comparison of different models before and after feature selection through the integration of Grey Wolf Optimization (GWO) and Particle Swarm Optimization (PSO) followed by MI (Mutual Information) and ANOVA F-value reveals interesting patterns is illustrated in Fig. 6.

The Random Forest and Decision Tree models demonstrate consistently high training accuracies of 99.76% and 99.76% respectively. However, their test accuracies show notable differences. The Random Forest achieved a test accuracy of 95.28%, while the Decision Tree's test accuracy was 94.01%, indicating some performance drop-off in generalization.

Logistic Regression showed improved performance compared to previous results, with a training accuracy of 90.30% and a test accuracy of 90.19%, demonstrating better generalization capabilities. XGBoost maintained strong performance with a test accuracy of 94.84%, indicating robust performance on unseen data.

Fig. 7 Recall and Precision comparison

Fig. 7 presents a comparative analysis of precision and recall metrics for the various models before and after the GWO PSO-based feature selection process.

The comparative analysis of precision and recall metrics for the various models after GWO-PSO feature selection shows:

- Random Forest achieved a precision of 96.43% and recall of 96.17%, demonstrating balanced performance in both metrics
- XGBoost showed strong results with a precision of 96.52% and recall of 95.35%
- Decision Tree maintained solid performance with a precision of 95.28% and recall of 95.32%
- Logistic Regression showed improved balance with a precision of 89.38% and recall of 96.03%

The comparison of precision and recall across models illustrates the effectiveness of the filter based GWO PSO feature selection approach in enhancing the predictive performance of most models, particularly ensemble methods, while also showcasing the limitations of Logistic Regression in this context.

Fig. 8 TPR and FPR Comparison

The True Positive Rate (TPR) and False Positive Rate (FPR) comparison provides crucial insights into the models' performance in correctly identifying positive cases and their tendency to generate false alarms. The Random Forest model shows a slight improvement in both TPR and FPR after feature selection. The TPR increased by 0.079%, indicating a marginal enhancement in correctly identifying positive cases. More notably, the FPR decreased by 7.06%, suggesting a reduction in false alarms.

The Decision Tree model demonstrates modest improvements. The TPR increased by 0.17%, while the FPR decreased by 1.21%. These changes, while small, indicate a slight enhancement in the model's overall performance.

In Logistic Regression model, the TPR increased by 0.49%, but more importantly, the FPR decreased dramatically by 56.27%. This substantial reduction in false positives suggests that the feature selection greatly enhanced the model's ability to distinguish between classes.

XGBoost shows minimal changes after feature selection. The TPR decreased slightly by 0.11%, while the FPR improved marginally by 2.25%. These small changes suggest that the XGBoost model was already well-optimized before feature selection.

The True Negative Rate (TNR) and False Negative Rate (FNR) comparison provides insights into the models' ability to correctly identify negative cases and their tendency to miss positive cases.

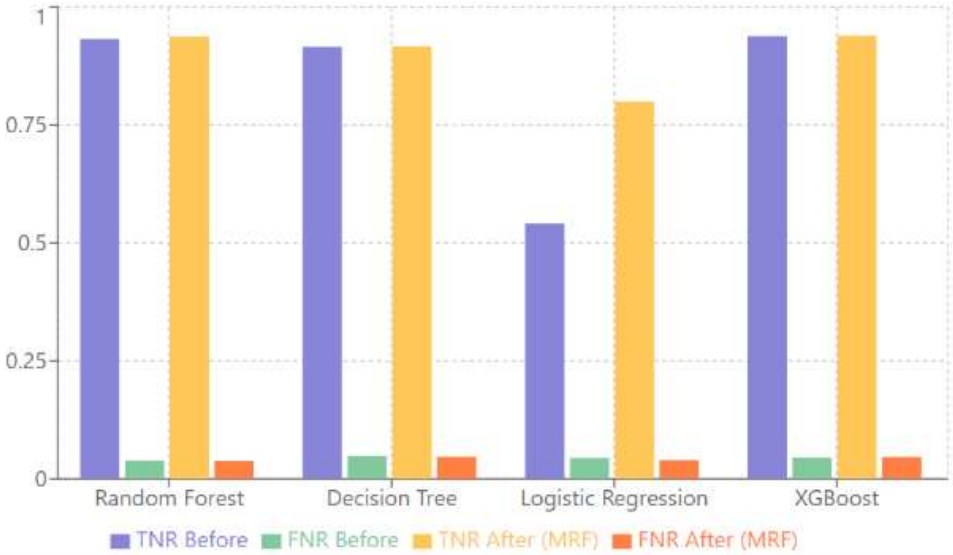

Fig. 9 TNR and FNR Comparison

The Random Forest model shows improvements in both TNR and FNR. The TNR increased by 0.51%, indicating better identification of true negatives. The FNR decreased by 1.95%, suggesting fewer missed positive cases.

The Decision Tree model shows slight improvements. The TNR increased by 0.11%, while the FNR decreased by 3.27%. These changes indicate a small enhancement in the model's ability to correctly identify negative cases and reduce missed positives.

Logistic Regression demonstrates the most substantial improvements. The TNR increased dramatically by 47.75%, indicating a significant enhancement in correctly identifying negative cases. The FNR decreased by 10.55%, suggesting fewer missed positive cases. These improvements align with the significant FPR reduction observed in Fig. 8.

XGBoost shows minimal changes after feature selection. The TNR improved slightly by 0.15%, while the FNR increased marginally by 2.34%. These small changes further support the observation that XGBoost was already well-optimized before feature selection.

Logistic Regression consistently shows the most significant improvements across all metrics, suggesting that it benefits the most from the Filter based GWO-PSO feature selection method. Random Forest and Decision Tree show consistent, albeit modest, improvements across all metrics. XGBoost demonstrates the least change after feature selection, indicating that it was already well-optimized for the given feature set.

Table 4. TNR and FNR, TPR and FPR Comparison

Metric	Random Forest	Decision Tree	Logistic Regression	XGBoost
TPR	96.09% → 96.17% (+0.08%)	95.15% → 95.32% (+0.17%)	95.56% → 96.03% (+0.49%)	95.45% → 95.35% (-0.11%)
FPR	6.75% → 6.27% (-7.06%)	8.40% → 8.30% (-1.19%)	45.91% → 20.07% (-56.27%)	6.19% → 6.05% (-2.26%)
TNR	93.25% → 93.73% (+0.51%)	91.60% → 91.70% (+0.11%)	54.09% → 79.93% (+47.78%)	93.81% → 93.95% (+0.15%)
FNR	3.91% → 3.83% (-2.05%)	4.84% → 4.68% (-3.31%)	4.44% → 3.97% (-10.59%)	4.54% → 4.65% (+2.42%)

The model's discriminative capability is demonstrated through the Receiver Operating Characteristic (ROC) curve and Precision-Recall (PR) curve analysis.

Table 5. AUC-ROC and AUC-PR Comparison

Model	AUC-ROC	AUC-PR
Random Forest	99.20% → 99.25% (+0.05%)	99.52% → 99.56% (+0.04%)
Decision Tree	93.60% → 93.74% (+0.15%)	93.90% → 94.03% (+0.14%)
Logistic Regression	86.29% → 97.56% (+13.04%)	89.47% → 98.55% (+10.14%)
XGBoost	99.19% → 99.18% (-0.01%)	99.55% → 99.55% (+0%)

The ROC curve exhibits exceptional performance with an Area Under the Curve (AUC) of 0.989, indicating the model's robust ability to distinguish between classes. The curve shows a sharp vertical rise in the True Positive Rate at very low False Positive Rates, reaching approximately 0.8.

TPR at nearly zero FPR, before continuing to climb more gradually to 1.0. This steep initial ascent demonstrates the model's strong discriminative power, particularly at strict classification thresholds. The substantial separation from the diagonal reference line (shown as a dashed blue line) further confirms the model's performance far exceeds random classification.

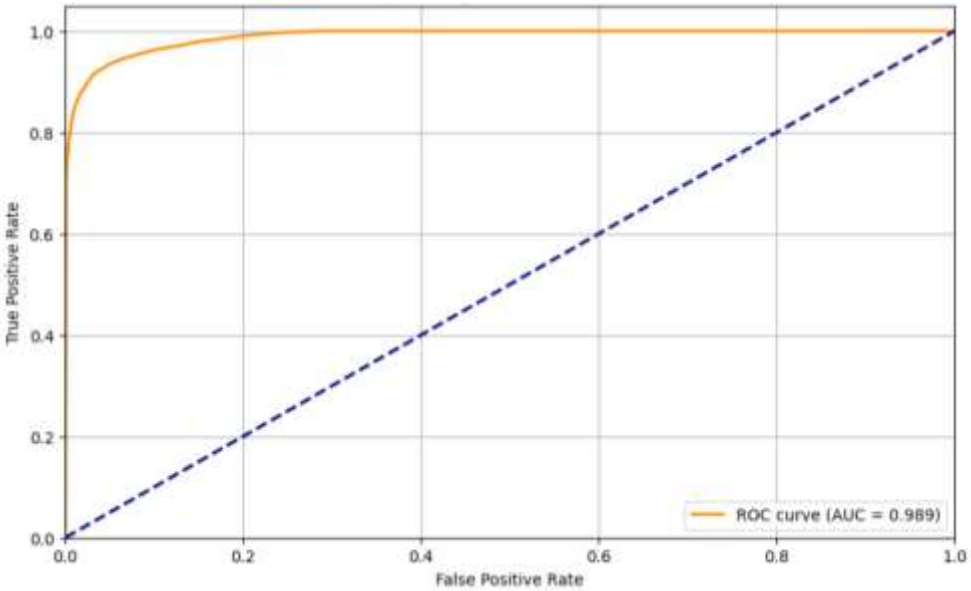

Fig. 10 AUC-ROC curve

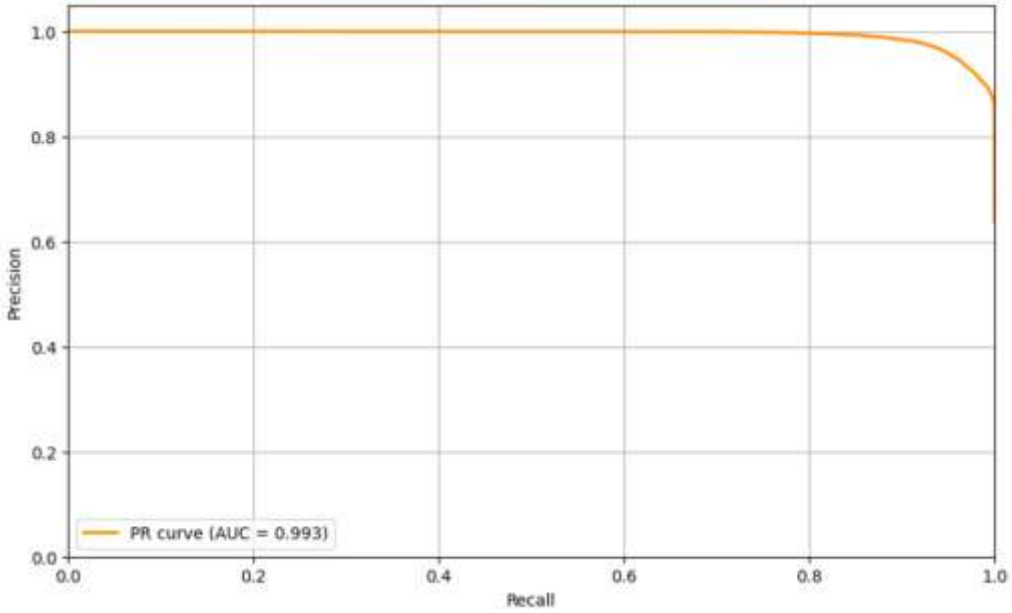

Fig. 11 AUC-PR Curve

The Precision-Recall curve complements these findings, displaying an AUC of 0.993, which indicates exceptional performance in maintaining high precision across different recall values. The curve maintains a near-perfect precision of approximately 1.0 across most of the recall range, with only a slight decline in precision at very high recall values (beyond 0.8).

This slight decline occurs in the high-recall region where the model must make more difficult classification decisions. The remarkably high and stable precision up to 0.8 recall suggests that the model makes very few false positive predictions while correctly identifying the majority of positive cases.
The curve's shape, maintaining high precision even at moderate to high recall values, indicates that the model achieves a favorable balance between precision and recall, making it particularly suitable for applications where both metrics are crucial.

Now let’s analyse the confusion matrices for four machine learning models (Random Forest, Decision Tree, Logistic Regression, and XGBoost) applied to the MRF (Mutual Random Forest) feature selection method. Each matrix provides insights into the models' classification performance on a binary classification task, likely related to network intrusion detection.

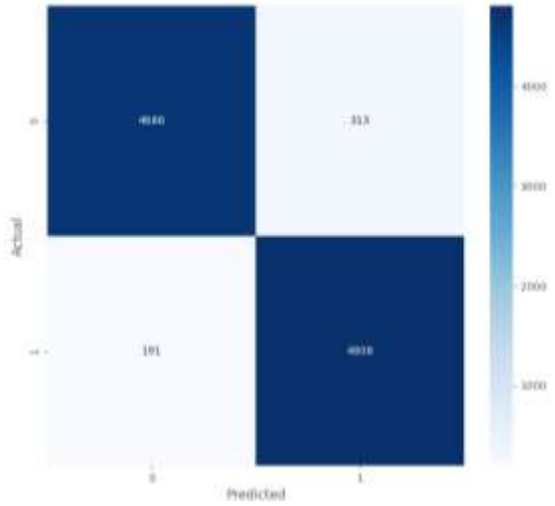

Fig. 12 Random Forest

The Random Forest model demonstrates high accuracy with 4686 true negatives (TN) and 4808 true positives (TP). It shows a low false positive rate with only 313 false positives (FP) and 191 false negatives (FN). This indicates excellent performance in correctly identifying both normal and anomalous network behaviour.

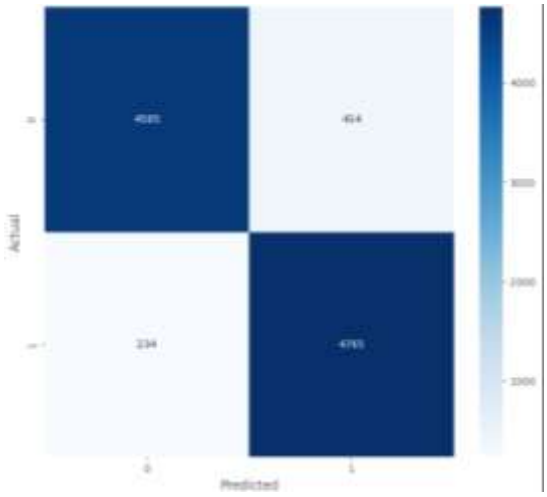

Fig. 13 Decision tree

The Decision Tree model performs similarly to Random Forest, with 4585 TN and 4765 TP. It has slightly higher misclassifications with 414 FP and 234 FN. While still effective, it shows marginally lower precision in identifying normal traffic compared to Random Forest.

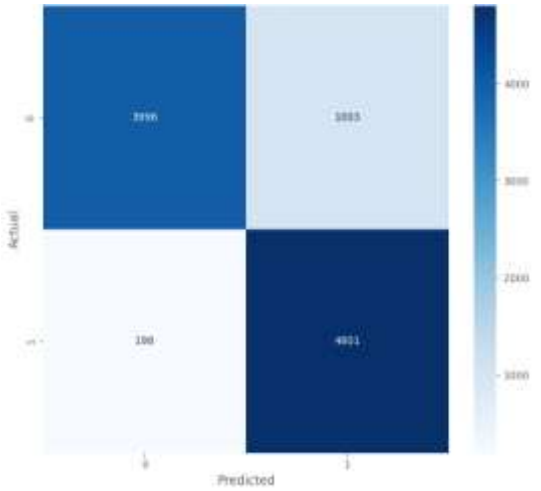

Fig. 14 Logistic Regression

Logistic Regression model exhibits the highest number of false positives (1003) among all models, with 3996 TN and 4801 TP. It maintains a low false negative rate (198), suggesting a tendency to overclassify normal traffic as anomalous. This could lead to higher false alarm rates in a real-world implementation.

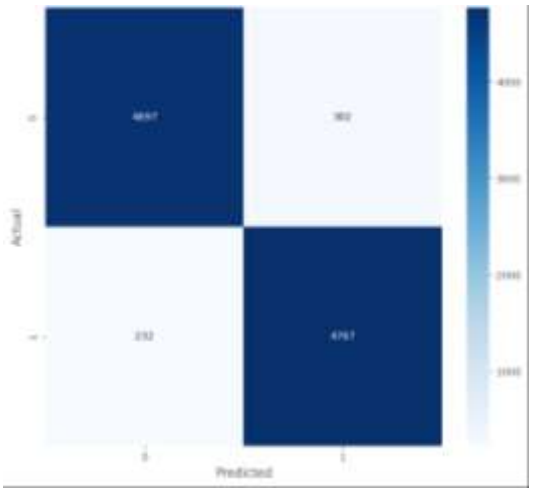

Fig. 15 XGBoost

XGBoost demonstrates performance comparable to Random Forest, with 4697 TN and 4767 TP. It has the lowest false positive count (302) and a low false negative count (232), indicating a well-balanced model with high accuracy in both classes.

The confusion matrices reveal that all models achieve high accuracy, with XGBoost and Random Forest slightly outperforming the others. The Decision Tree model shows balanced performance, while Logistic Regression exhibits a higher tendency for false positives. These results suggest that ensemble methods (Random Forest and XGBoost) are particularly effective for this classification task when using MRF features. The higher false positive rate in Logistic Regression indicates it may be less suitable for scenarios where minimizing false alarms is crucial. The high true positive rates across all models, particularly in Random Forest and XGBoost, indicate strong capability in detecting anomalous network behaviour. However, the varying false positive rates highlight the importance of model selection based on specific system requirements, such as tolerance for false alarms versus the need for high detection rates.

Table 6. Computational performance

Model	Training Time (seconds)	Detection Time (seconds)
Random Forest	140.83 → 72.29 (-48.66%)	2.52 → 1.44 (-42.86%)
Decision Tree	15.01 → 6.34 (-57.75%)	0.08 → 0.03 (-62.50%)
Logistic Regression	6.17 → 36.70 (+494.17%)	0.11 → 0.03 (-72.73%)
XGBoost	13.85 → 11.92 (-13.92%)	0.34 → 0.32 (-5.88%)

The computational performance of the models was evaluated based on their training and detection times. Among the models, Random Forest demonstrated the highest training time at 72.29 seconds, reflecting the complexity of its ensemble learning process, while Decision Tree was the fastest to train, requiring only 6.34 seconds.

Logistic Regression and XGBoost had moderate training times of 36.70 seconds and 11.92 seconds, respectively. In terms of detection time, both Decision Tree and Logistic Regression were exceptionally fast, completing detection in just 0.03 seconds, making them highly efficient for real-time applications. XGBoost followed with a detection time of 0.32 seconds, while Random Forest required the longest detection time at 1.44 seconds.

These results highlight a trade-off between model complexity and computational efficiency, with simpler models like Decision Tree excelling in speed, while more complex models like Random Forest and XGBoost offer higher performance metrics but demand greater computational resources.

5. Conclusion

This study demonstrates the effectiveness of filter based hybrid feature selection, integrating Grey Wolf Optimization (GWO) and Particle Swarm Optimization (PSO), in enhancing intrusion detection system (IDS) performance. By reducing features to a carefully selected subset, the Hybrid GWOPSO approach improved model accuracy, precision, recall, and computational efficiency across multiple machine learning models, including Random Forest, Decision Tree, Logistic Regression, and XGBoost. Random Forest and Decision Tree, particularly, showed strong performance with minimal loss in predictive power after feature selection, while Logistic Regression benefited most from reduced features, showing marked improvement in generalization and reduced false positives. XGBoost, already highly optimized, maintained stable performance even with feature reduction, reflecting its robustness.

The results underscore the significance of feature selection in IDS, highlighting that optimized feature sets can effectively balance model accuracy with computational efficiency. This makes them valuable for real-world applications, where minimizing processing time and false alarms is essential for robust and reliable intrusion detection.

References

[1] Harbi, Y., Merat, S., Aliouat, Z., & Harous, S. (2024, May). Bio-inspired Intrusion Detection System for Internet of Things Networks Security. In Proceedings of the Cognitive Models and Artificial Intelligence Conference (pp. 14-19).*)*

[2] Saheed, Y. K., Kehinde, T. O., Ayobami Raji, M., & Baba, U. A. (2024). Feature selection in intrusion detection systems: a new hybrid fusion of Bat algorithm and Residue Number System. *Journal of Information and Telecommunication*, *8*(2), 189-207.

[3] Bakro, M., Kumar, R. R., Husain, M., Ashraf, Z., Ali, A., Yaqoob, S. I., ... & Parveen, N. (2024). Building a cloud-IDS by hybrid bio-inspired feature selection algorithms along with random forest model. *IEEE Access*.

[4] Mohammad, A. H. (2021). Intrusion Detection Using a New Hybrid Feature Selection Model. *Intelligent Automation & Soft Computing*, *30*(1).

[5] Otair, M., Ibrahim, O. T., Abualigah, L., Altalhi, M., & Sumari, P. (2022). An enhanced grey wolf optimizer based particle swarm optimizer for intrusion detection system in wireless sensor networks. *Wireless Networks*, *28*(2), 721-744.

[6] Ali, M. H., Al Mohammed, B. A. D., Ismail, A., & Zolkipli, M. F. (2018). A new intrusion detection system based on fast learning network and particle swarm optimization. *IEEE Access*, *6*, 20255-20261.

[7] Vijayanand, R., & Devaraj, D. (2020). A novel feature selection method using whale optimization algorithm and genetic operators for intrusion detection system in wireless mesh network. *IEEE Access*, *8*, 56847-56854.

[8] Moghanian, S., Saravi, F. B., Javidi, G., & Sheybani, E. O. (2020). GOAMLP: Network intrusion detection with multilayer perceptron and grasshopper optimization algorithm. *IEEE Access*, *8*, 215202-215213.

[9] Hajimirzaei, B., & Navimipour, N. J. (2019). Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm. Ict Express, 5(1), 56-59.

[10] Eesa, A. S., Orman, Z., & Brifcani, A. M. A. (2015). A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems. Expert systems with applications, 42(5), 2670-2679.

[11] Manokaran, J., & Vairavel, G. (2024). DL-ADS: Improved Grey Wolf Optimization Enabled AE-LSTM Technique for Efficient Network Anomaly Detection in Internet of Thing Edge computing. *IEEE Access*.

[12] Keserwani, P. K., Govil, M. C., Pilli, E. S., & Govil, P. (2021). A smart anomaly-based intrusion detection system for the Internet of Things (IoT) network using GWO–PSO–RF model. Journal of Reliable Intelligent Environments, 7(1), 3-21.

[13] Almomani, O. (2020). A feature selection model for network intrusion detection system based on PSO, GWO, FFA and GA algorithms. Symmetry, 12(6), 1046.

[14] Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. *IEEE transactions on emerging topics in computational intelligence*, *2*(1), 41-50.

[15] Safaldin, M., Otair, M., & Abualigah, L. (2021). Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks. *Journal of ambient intelligence and humanized computing*, *12*, 1559-1576.

[16] Safaldin, M., Otair, M., & Abualigah, L. (2021). Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks. *Journal of ambient intelligence and humanized computing*, *12*, 1559-1576.

[17] Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications*, *50*, 10241

[18] Ambusaidi, M. A., He, X., Nanda, P., & Tan, Z. (2016). Building an intrusion detection system using a filter-based feature selection algorithm. *IEEE transactions on computers*, *65*(10), 2986-2998.

[19] Siddiqi, M. A., & Pak, W. (2020). Optimizing filter-based feature selection method flow for intrusion detection system. *Electronics*, *9*(12), 2114.