Enhanced Network Intrusion Detection via Gradient Boosting Tuned by Emperor Penguin Optimization Algorithm (EPOA)

Lina Qin
Informatization Department, Wuhan Business University, Wuhan City, Hubei Province, 430056, China
E-mail: qln20250422@163.com

Keywords: cybersecurity, network intrusion detection, classification, machine learning

Received: May 25, 2025

Protecting network infrastructures from increasingly complex cyberthreats requires the use of intrusion detection systems, or IDSs. However, because of changing attack patterns and high data dimensionality, it is still difficult to differentiate between malicious and benign network activity. In order to improve IDS performance, this study critically evaluates six popular machine learning classifiers: Random Forest (RF), Gradient Boosting (GB), Decision Tree (DT), XGBoost (XGB), AdaBoost (AB), and K-Nearest Neighbors (KNN). Two sophisticated hyperparameter tuning methods, Grid Search (GS) and the Emperor Penguin Optimization Algorithm (EPOA), were used to increase predictive accuracy and model robustness. With accuracy, precision, recall, and F1-score values of 0.9997, 0.9898, 0.9999, and 0.9948, respectively, the optimized Gradient Boosting (EPOA-GB) model outperformed the others. Important contributing features were also found using SHAP-based interpretability analysis, which provided insightful information about the classification procedure. The models became more scalable for deployment when Principal Component Analysis (PCA) was used to reduce dimensionality, improving generalization and computational efficiency. These results show how well ensemble classifiers and intelligent optimization work together to reduce false alarms, a crucial requirement for real-time intrusion detection. This work provides practical guidelines for implementing high-performance IDSs and highlights the importance of future validation across diverse datasets and deployment environments to ensure robustness and adaptability in real-world cybersecurity scenarios.

Povzetek: Študija primerja šest klasifikatorjev IDS ter z EPOA-optimizacijo nastavi Gradient Boosting. EPOA-GB na CIC-IDS-2017 doseže najboljše rezultate, uporabi PCA in SHAP, zmanjša lažne alarme.

1 Introduction

In recent years, cybersecurity has gained significant attention owing to the increasing reliance on digital infrastructure and the sharp rise in sophisticated cyberattacks. These attacks, ranging from disruptions in electrical grids to advanced assaults on SCADA systems and high-profile incidents like the Stuxnet virus targeting nuclear facilities [1], pose severe threats to national security, critical infrastructure, and private enterprises. The financial and operational consequences of such attacks are often catastrophic, emphasizing the urgent need for advanced security mechanisms.

Intrusion detection systems (IDSs) serve as a vital line of defense in network security, enabling the identification of both ongoing intrusions and previously compromised systems. An IDS, implemented as either hardware or software, monitors network or system traffic and triggers alerts upon detecting suspicious patterns [2], [3]. Depending on design, IDSs can be categorized based on data collection methods (e.g., host-based, network-based), deployment strategy (e.g., centralized, hybrid), or detection technique (e.g., signature-based, anomaly-based, or hybrid) [4], [5], [6].

However, traditional IDS solutions, such as signature-based and rule-driven systems, struggle to keep pace with the dynamic landscape of cybersecurity threats, particularly zero-day exploits and advanced persistent threats (APTs). These conventional systems lack adaptability and often produce high false-positive rates when facing novel attack patterns. This growing complexity, along with the exponential increase in network traffic, highlights the need for intelligent, scalable, and real-time detection frameworks.

In response, machine learning (ML)-based IDSs have emerged as a powerful alternative, leveraging data-driven algorithms to detect anomalies, learn from evolving threats, and generate accurate predictions with minimal human intervention. ML methods not only enhance detection accuracy but also offer flexibility and automation in processing large-scale network data. As such, ML-integrated IDSs represent a promising direction in modern cybersecurity. Several recent studies have explored this paradigm, aiming to optimize intrusion detection in terms of accuracy, efficiency, and interpretability [7], [8], [9], [10], [11]. To illustrate the current landscape and identify research gaps, Table 1 summarizes a comparison of key works applying ML models for IDS development.

Table 1: Overview of the related works.

Ref.	Dataset	Best Model	Accuracy	Other Metrics	Gaps/Limitations
[12]	CIVEMSA-2020	Deep Neural Network (DNN)	0.9978	-	No hyperparameter optimization; unclear on overfitting and interpretability
[13]	Various (incl. KDD, CIC)–comparative; primary on classic IDS)	Decision Tree (DT)	0.9992	Precision, Geometric mean, F-measure	Limited to default parameters, no tuning optimization, lacks interpretability discussion
[14]	HONET 2020 (Smart Communities)	Instance-Based Learning algorithm (IBK)	0.9982	Precision, Recall, F-measure, Receiver Operating Curve (ROC)	Small feature set used; no thorough hyperparameter tuning; only basic metrics
[15]	CICIDS-2017	Random Forest (RF)	0.9986	Precision, Recall	Features selected but no optimization algorithm for RF; no interpretability or false positive rate analysis
[16]	CIC-IDS (Canadian Institute of Cybersecurity)	Extreme Gradient Boosting (XGB)	0.9954	-	Lacks detailed tuning description, interpretability, and metrics beyond accuracy
[11]	Custom 5G IoT dataset	Hybrid DNN + Feature Engineering	0.9960	Recall, F1-measure, ROC AUC	Focused on 5G-specific threats; lacks generalizability; limited optimization details
[17]	CIC-IDS-2017 and CSE-CIC-IDS-2018	Hybrid CNN-LSTM	0.9730	Recall, Precision, F1, AUC	Strong architecture but lacks interpretability analysis; tuning method not clearly described
[18]	UNSW-NB15	Bi-LSTM + Attention	0.9800	Recall, F1 Score, False Positive Rate, False Negative Rate, t-test (p-value)	Effective against specific scanning attacks, but not validated on broader benchmark datasets
This study	CIC-IDS-2017	Emperor Penguin Optimization Algorithm-Gradient Boosting (EPOA-GB)	0.9997	Precision, Recall, F1-Score, ROC AUC, PR AUC, Log loss, MCC, Cohen's Kappa	Addresses gaps in prior works, including hyperparameter optimization, interpretability (SHAP), robust validation, and comprehensive metrics

This study introduces several key innovations to address the limitations commonly found in existing classification modeling approaches. While many prior works rely on default settings or manual hyperparameter tuning—such as in DNNs and DTs—we employ the Emperor Penguin Optimization Algorithm (EPOA) to automatically and efficiently tune the GB classifier, leading to improved model performance. Overfitting and robustness, often overlooked in previous studies, are explicitly addressed in our approach through the use of cross-validation and evaluation on held-out test datasets. Furthermore, to tackle the frequent lack of interpretability in traditional models, we incorporate SHAP (SHapley Additive exPlanations) and perform feature importance analysis to provide deeper insights into model behavior and feature contributions. Unlike prior research that typically reports limited metrics, our study offers a comprehensive performance evaluation by including accuracy, precision, recall, F1-score, and ROC AUC (Receiver Operating Characteristic-Area Under Curve), PR AUC (Precision-Recall-Area Under Curve), Log loss, Matthews Correlation Coefficient (MCC), Cohen's Kappa. Finally, to enhance generalizability, we validate our approach on another state-of-the-art tuning model rather than relying on traditional benchmark models, demonstrating the robustness and wide applicability of our method. Collectively, these contributions represent a significant advancement over existing methods in terms of optimization, interpretability, evaluation, and generalizability.

Therefore, the major novelty in the current work is in its overall approach toward enhancing IDS through integrating high-performance classification models with an advanced optimization algorithm. Unlike traditional IDS frameworks with static settings or basic tuning techniques, in this work, a new algorithm, the EPOA, is proposed for application in a dynamic hyperparameter optimization method for diverse models. The suggested

method provides a secure and efficient mechanism for real-time implementation by systematically improving computational models. Moreover, the comparative analysis in the study not only compares model performance in individual terms but also identifies the transformational role played by EPOA in enhancing model efficiency, stability, and overall generalization performance. The main objectives of this work involve enhancing accuracy in detecting malicious and non-malicious activities, minimizing false positive and false negative occurrences, and lessening computational expense in processing large networks. By combining these, this work proposes a new, flexible, and efficient IDS model with an outlook toward addressing evolving network security concerns.

Hence, in order to guide the present study, the following key research questions have been posed: **1.** Can the proposed EPOA-GB significantly enhance the performance of a ML-based IDS, particularly in comparison to other conventional as well as state-of-the-art tuning approaches? **2.** Does integrating EPOA with Gradient Boosting improve model robustness and generalizability across the employed dataset?

We hypothesize that EPOA-GB, as a bio-inspired metaheuristic algorithm, will outperform other applied tuning methods in identifying optimal hyperparameters, thereby boosting detection accuracy, reducing false positives, and improving the generalization of the IDS model. We further expect that model interpretability tools will provide actionable insights into the prediction process. To address these questions, our methodology includes **(i)** applying EPOA for automated hyperparameter optimization of the selected ML models, **(ii)** evaluating performance on benchmark dataset using cross-validation and a comprehensive set of metrics, and **(iii)** employing SHAP and feature importance analyses to interpret model predictions. Each step is designed to directly respond to the research questions and validate our hypotheses through the evaluation.

This paper is organized into key sections addressing multiple aspects of the classification tasks. The methodology comes in Section 2, introducing the dataset, its sources, feature selection, model selection, an optimizer, and evaluation metrics. Section 3 covers model performance, including key observations, trends, and insights. Section 4 summarizes key results, explains their implications, and sets out avenues for future work.

2 Methodology

An efficient IDS requires methodical development, beginning with the data preparation and concluding with model evaluation. ML models serve as a powerful tool for discovering cyberattacks through a mechanism of training on network traffic information. A sequential process for network intrusion detection via ML models using CIC-IDS (Canadian Institute for Cybersecurity) data is presented in Figure 1. Initially, the objective is determined, and then feature extraction and partitioning of 70:30 training and testing sets for a relevant dataset follow. The six models, namely Random Forest (RF), Gradient Boosting (GB), Decision Tree (DT), XGBoost (XGB), AdaBoost (AB), and K-Nearest Neighbors (KNN), are deployed to the classification issue. For enhancing performance, the models' hyperparameters are tuned with EPOA, a high-performance algorithm inspired by penguins' foraging behavior in a real environment. Lastly, model evaluation is performed to choose the best-performing classifier.

It is worth mentioning that, the choice of ML models in this study is theoretically motivated by their complementary strengths in handling various challenges intrinsic to intrusion detection tasks, such as high-dimensional feature spaces, imbalanced data, and the need for generalization under noisy conditions. RF was selected for its ensemble architecture and robustness to overfitting, especially in large and noisy datasets. DTs provide interpretability and serve as a baseline, while GB and XGB offer superior accuracy by focusing on correcting prior errors in a sequential manner, making them suitable for complex intrusion patterns. AdaBoost was chosen for its ability to minimize bias through weight adjustment, and KNN was involved owing to its non-parametric, instance-based nature which offers a contrasting learning paradigm that can benefit small or local pattern detection. Moreover, the selection of the EPOA for hyperparameter tuning is grounded in its demonstrated success as a metaheuristic optimizer in solving nonlinear, high-dimensional problems. Its swarm-intelligence-inspired mechanism offers a balance between global exploration and local exploitation, which is critical for fine-tuning complex models. Compared to traditional grid or random search approaches, EPOA improves convergence to optimal hyperparameter configurations, thereby enhancing model stability and predictive accuracy in IDS tasks.

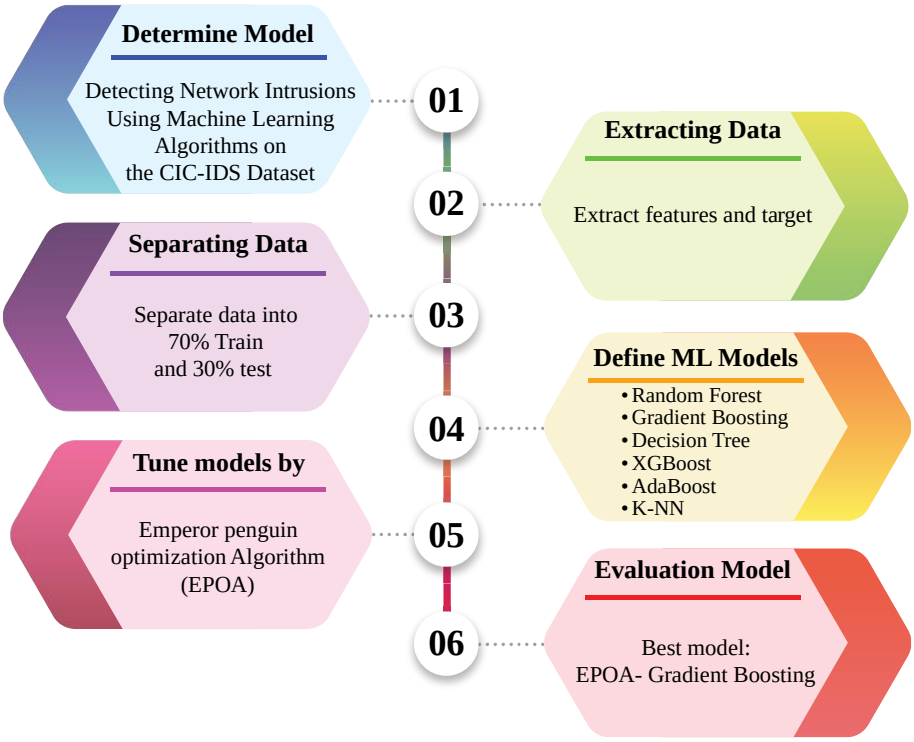


Figure 1: Visual presentation of the research framework.

2.1 Dataset

The CIC-IDS-2017 dataset used in this study was sourced from [19]. This dataset is rich in network flow data, capturing essential attributes that make it a valuable resource for cybersecurity research and intrusion detection system (IDS) development. Prior to applying ML models, the dataset undergoes a comprehensive preprocessing pipeline. This includes handling missing values to avoid bias during model training and encoding categorical features to ensure compatibility with various algorithms. Numerical features are standardized to bring them to a common scale, which is essential for the performance of most ML models. Following standardization, dimensionality reduction is performed using Principal Component Analysis (PCA), reducing the feature space to 20 principal components while preserving most of the dataset's variance. This step enhances computational efficiency, mitigates the risk of overfitting, and improves the model's generalization ability on unseen data. Figure 2 shows the PCA feature loadings heatmap generated to visualize the correlation between network flow features and principal components, facilitating dimensionality reduction for improved model performance. And, Figure 3 demonstrates the PCA variance plot employed to determine the number of components needed to retain 90% of the data's variance, guiding effective dimensionality reduction.

Additionally, a key data-cleaning step involves removing unnecessary spaces from column names to maintain consistency throughout the dataset. To ensure the reliability and robustness of model evaluation, k-fold cross-validation is employed, allowing the models to be trained and validated across multiple data splits. This approach not only prevents overfitting but also provides a more accurate estimate of model performance.

After that, the target variable and input variables are specified. The target variable is then converted into four classes, where BENIGN maps to 0 and Web Attack – Brute Force, Web Attack – XSS, and Web Attack – SQL Injection labels are respectively mapped to 1, 2, 3. This classification framework simplifies the detection problem for models by targeting either normal or abnormal behaviors. A bar chart in Figure 4 illustrates that benign traffic comprises the majority of the dataset, with significantly fewer instances of web attacks such as brute force, XSS, and SQL injection.

The cleaned data is then divided into test (30%) and training (70%) sets in preparation for model training. This way, the model will be evaluated on unseen data; hence, it can be tested for its generalization capability. Moreover, the features are standardized using Standard Scaler; this normalizes the data, making it suitable for ML models. With the completion of these preprocessing steps, this dataset is now ready for training high-performance models for network intrusion detection, furthering cybersecurity research.

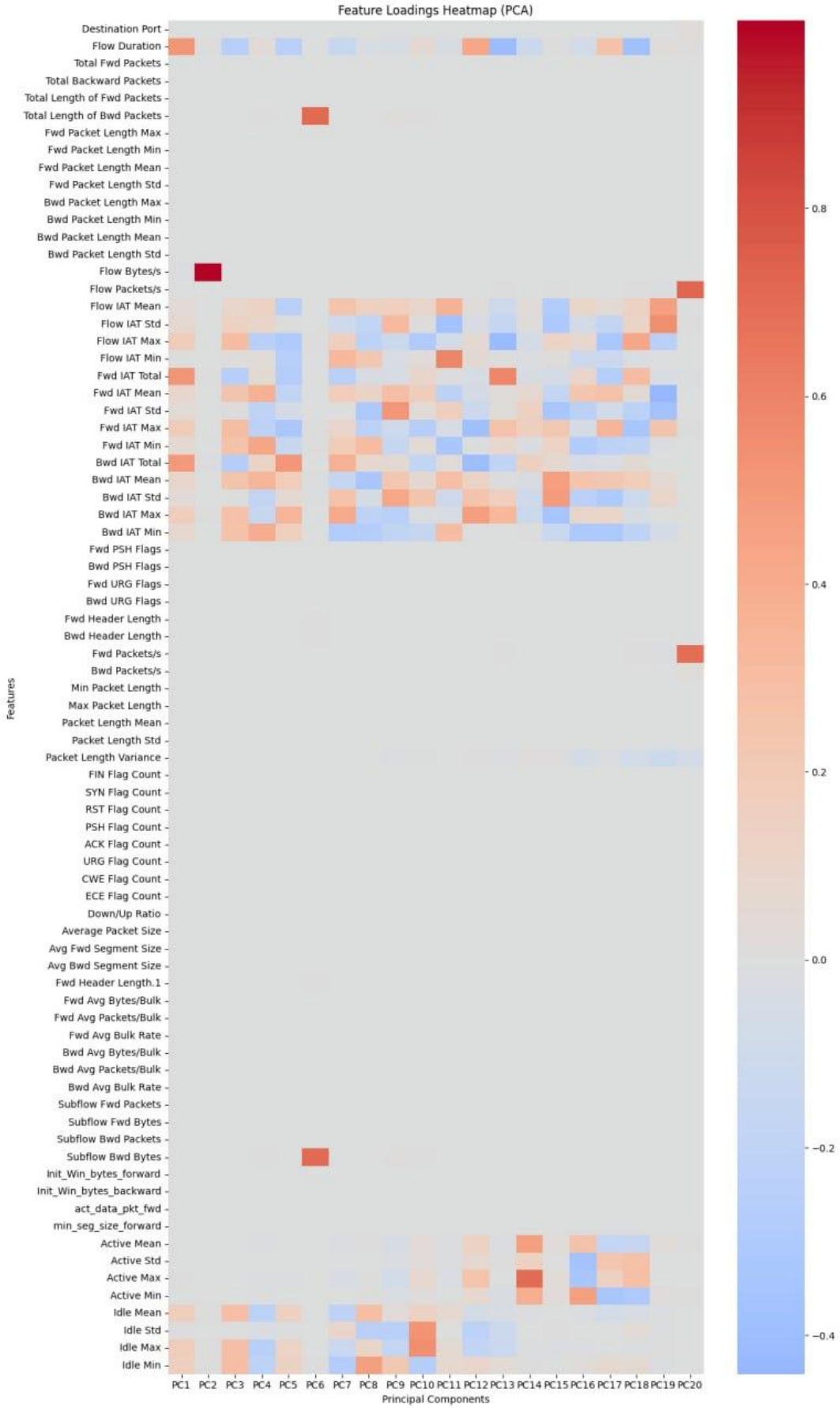


Figure 2: PCA feature loadings heatmap showing correlations between network flow features and principal components.

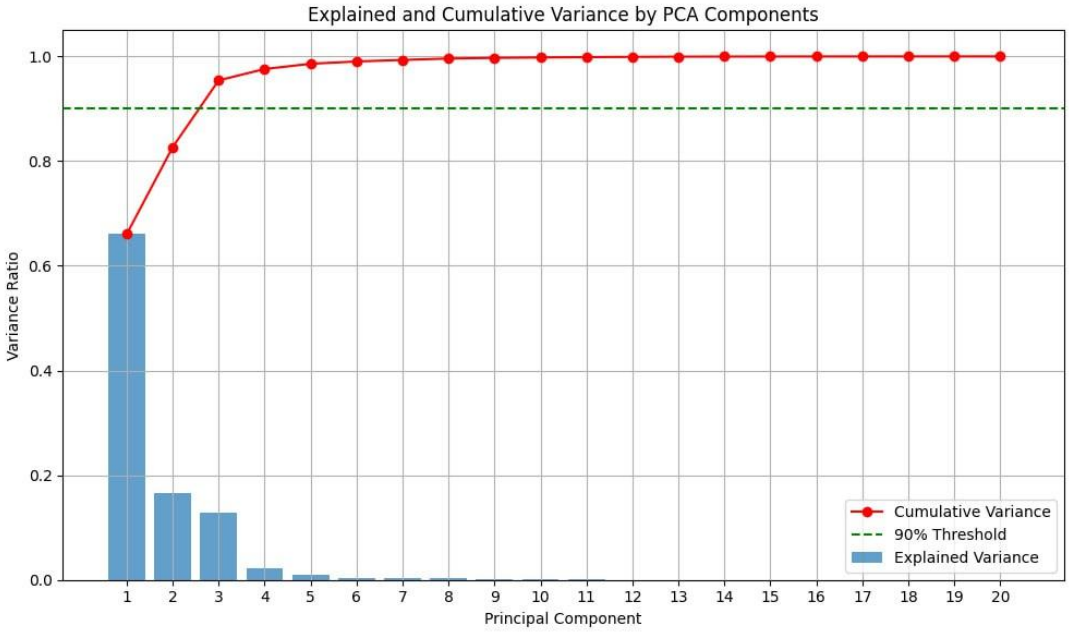


Figure 3: Explained and cumulative variance of principal components from PCA for dimensionality reduction.

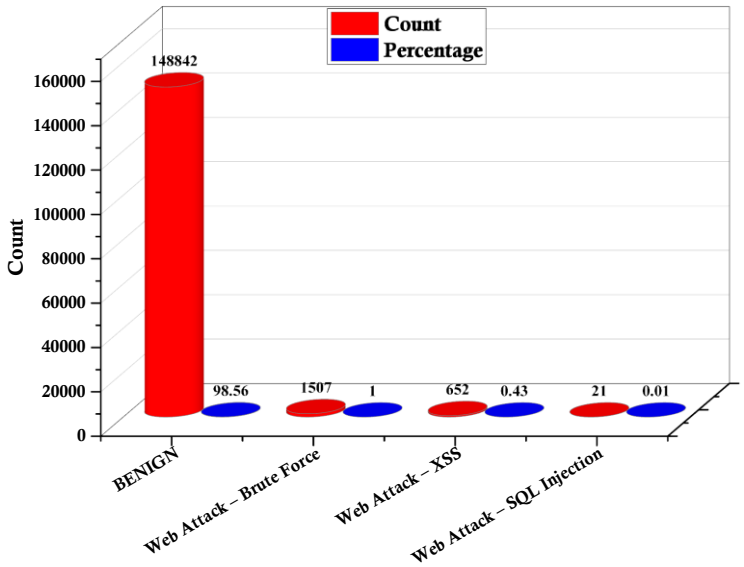


Figure 4: Bar chart showing the frequency distribution of web attack types, highlighting the dominance of benign traffic.

2.2 Model selection

2.2.1 Random Forest

Random Forest (RF) is an ensemble learning model that constructs many decision trees and aggregates them to make a prediction, enhancing predictive accuracy and minimizing overfitting. Initially, Breiman [20] developed RF by training numerous decision trees over a range of bootstrapped datasets, utilizing bagging and feature selection at each split on a random basis. Each tree classifies a new case individually, and the final classification is generated through a vote over all trees in a majority vote style. This helps in generalization and strengthening, making RF a strong tool for handling large and noisy datasets (see [20, 21]).

2.2.2 Gradient boosting

GB is another strong ensemble learner model that constructs predictive models sequentially, with each model optimized for performance through the improvement of preceding model errors. It was first proposed by Friedman [23] as an expansion of boosting algorithms. GB trains weak learners, in most cases DTs, in a sequential manner, with each successive tree attempting to minimize the residuals of its predecessor through gradient descent. In contrast to RF, whose trees are constructed individually, trees in GB are constructed

iteratively, with weights tuned to minimize bias and maximize accuracy (see [24]). This produces a very powerful model that can handle intricate patterns; however, it requires careful adjustment to avoid overfitting.

2.2.3 Decision Tree

DT is a supervised learner model that distinguishes data into hierarchical branches regarding feature values and ends in a predicted class label. By applying measures such as Gini impurity or information gain (based on entropy), the model builds a tree by choosing the most informative feature at each node. It recursively divides the data to a point when it reaches a leaf with a high level of purity, at which point classification occurs (see [25], [26]). DTs have simple interpretability and computational efficiency but suffer from overfitting, and pruning can counteract this.

2.2.4 Extreme Gradient Boosting

XGB, also called XGBoost, is a high-performance ML model that maximizes the efficiency, accuracy, and overall performance of traditional gradient boosting algorithms. XGB, developed by Chen and Guestrin [27], gained widespread acceptance in data science competitions and real-world implementation. XGB builds an ensemble model sequentially, with each new tree fitting in a direction that corrects residuals of preceding trees with gradient-boosted trees via gradient descent optimization. It offers important enhancements, including parallel computing, tree pruning, and regularization parameters for overfitting avoidance, which result in a quicker and more scalable approach. XGB is also efficient in dealing with missing values and can work with sparse data and is therefore ideal for sophisticated classification scenarios (see [27, 28]).

2.2.5 Adaptive Boosting

AB is a pioneer model in ensemble learning that takes several weak classifiers and forms a strong predictive model out of them. AB was developed by Freund and Schapire [30] as a refinement of boosting algorithms. The AB model iteratively trains weak learners on the training set, adjusting their weights according to classification errors. Misclassified cases receive increased weights in subsequent iterations, forcing the model to pay attention to challenging cases. Final classification is performed via a weighted vote of all weak classifiers. AB is flexible, simple, and can minimize bias with interpretability; however, it is not robust when dealing with outliers and noisy data (see [30, 31]).

2.2.6 K-Nearest Neighbors

KNN is a simple yet effective supervised learning approach, generating labels for classes through a similar-data point comparison. Initially, Fix [33] developed it as a non-parametric model for pattern classification. KNN operates through the computation of the distance of a query point from all training samples in a dataset. It then assigns a label to a new instance based on the most frequent label among its KNNs. Unlike most methods, KNN doesn't require explicit training; instead, it stores all training samples and compares them to new cases, making it a lazy learner algorithm. While it is efficient and simple for small datasets, it becomes computationally expensive for large datasets, as it requires distance calculations for each new prediction (see [33, 34]).

2.3 Optimization algorithm

Optimization algorithms serve as a basis for ML, and through them, model performance can be optimized. Such algorithms gain inspiration from many mathematical laws and processes in nature and work towards resolving complex optimization problems in many fields. The Emperor Penguin Optimization Algorithm (EPOA) [36] is one of the best nature-inspired algorithms for nonlinear and multidimensional problem solving; Figure 5 shows its conceptual workflow.

The emperor penguin optimizer is a metaheuristic algorithm inspired by emperor penguins' huddling behavior for survival in the extreme conditions of the Antarctic. It is proposed as a swarm intelligence approach and simulates the dynamic thermal control manner in which penguins form dense, regulated groups to maintain warmth. In optimization cases, this turns into candidate solutions (penguins) moving toward the best solution (warmest area) by modifying their positions according to temperature (fitness value). Exploration (random search for the diversity of solutions) and exploitation (refinement of a potential solution) are balanced through the simulation of penguins' movement behavior (see [37]). In this work, EPOA was successfully applied for hyperparameter tuning in all ML models, including RF, GB, DT, XGB, AB, and KNN, enhancing their accuracy and performance.

In addition to EPOA, Grid Search (GS) was applied as a baseline hyperparameter tuning method to benchmark optimization efficiency. The hyperparameters of all chosen models were optimized using both EPOA and GS, as detailed in Tables 2 and 3, respectively. The comparison shows that while EPOA consistently achieved high-performing configurations across all models, it also required the same iterations, but generally higher runtime than GS.

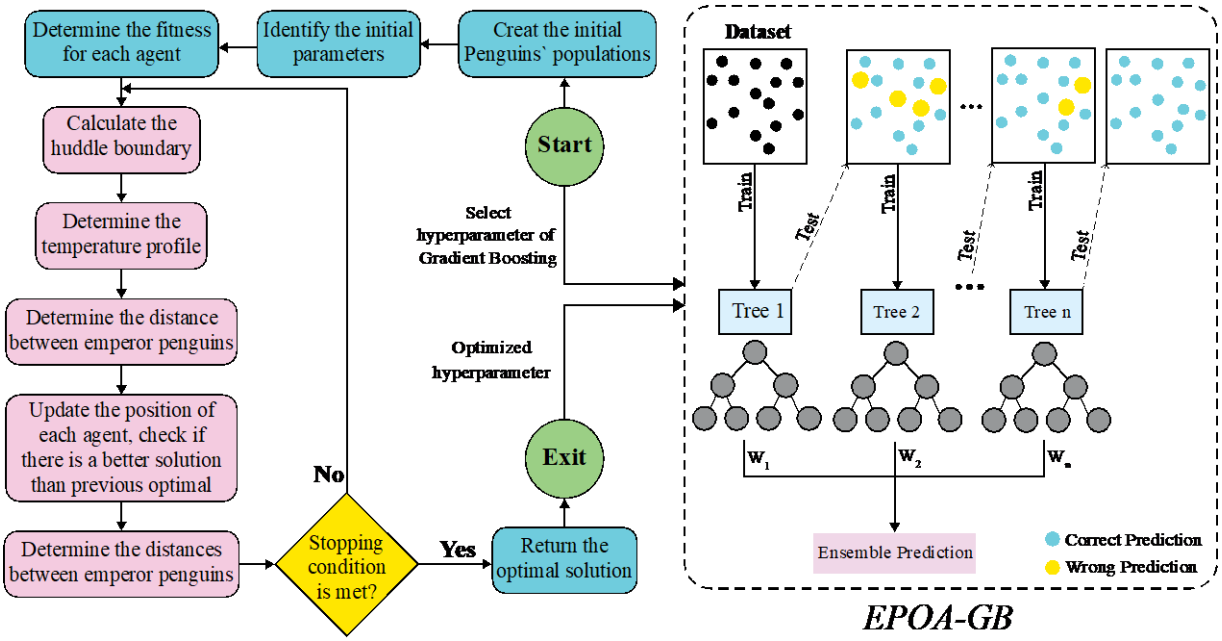


Figure 5: An example workflow of the GB model's hyperparameter tuning process using EPOA.

Table 2: The applied models' hyperparameter tuning using EPOA.

	EPOA-RF	EPOA-GB	EPOA-DT	EPOA-XGB	EPOA-AB	EPOA-KNN
N estimators	100	100	-	100	100	-
Min samples split	2	-	2	-	-	-
Min samples leaf	1	-	1	-	-	-
Max depth	None	3	None	3		-
Learning rate	-	0.1	-	0.1	0.1	-
Subsample	-	-	-	0.7		-
Colsample bytree	-	-	-	0.7		-
Weights	-	-	-	-		Uniform
P	-	-	-	-		2
N neighbors	-	-	-	-		5
Runtime (sec)	10206.15	100361.13	1002.1	2010.57	15009.47	10209.86
Iteration	30	30	30	30	30	30

Table 3: The applied models' hyperparameter tuning using Grid Search (GS).

	GS-RF	GS-GB	GS-DT	GS-XGB	GS-AB	GS-KNN
Max depth	7	5	3	12	-	-
Min samples leaf	1	-	8	-	-	-
Min samples split	3	8	8	-	-	-
N estimators	65	300	-	229	271	-
Learning rate	-	0.0653	-	0.0443	0.4729	-
Colsample bytree	-	-	-	0.8997	-	-
Subsample	-	-	-	0.9519	-	-
N neighbors	-	-	-	-	25	-
P	-	-	-	-	1	-
Weights	-	-	-	-	Distance	-
Runtime (sec)	90.8311	980.4113	8.7119	15.5305	114.8171	93.5015
Iteration	30	30	30	30	30	30

2.4 Evaluation Indicators

The effectiveness of ML models in identifying network intrusions is evaluated using a variety of statistical testing indicators. Comparing and evaluating a model through several factors helps in planning improvements and comparisons. Figure 6 depicts a selection of key evaluation statistics for application in classification scenarios. They involve the F1-score, which sets a balance between precision and recall; Cohen's Kappa, which is an

agreement between two raters that considers chance; recall, which measures how well the scheme detects positive cases; and Log loss, which evaluates the accuracy of probabilistic predictions. In addition, accuracy measures overall model correctness; the Matthews Correlation Coefficient (MCC) provides a balanced performance even in unbalanced datasets, and the proportion of correctly positive instances detected is measured through precision.

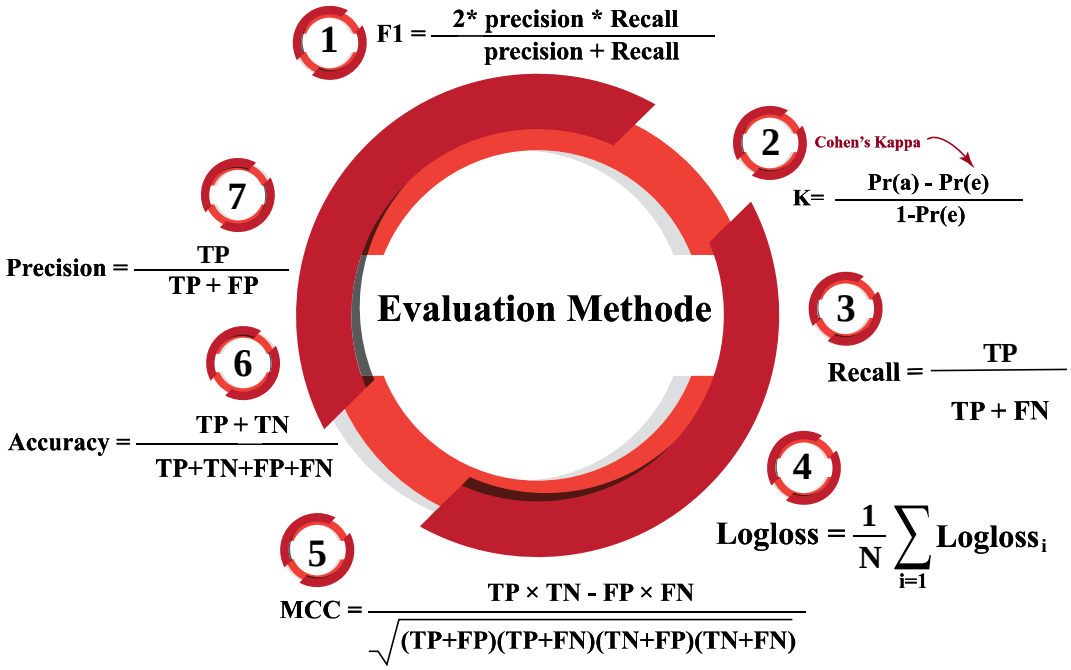


Figure 6: The formal description of several key indicators.

Other important evaluation indicators, apart from metrics in Figure 6, include PR AUC (Precision-Recall-Area Under Curve) and ROC AUC (Receiver Operating Characteristic-Area Under Curve). The ROC curves are plots of the true positive (TP) rate against the false positive (FP) rate at diverse threshold levels, which is an indicator of how well a model can differentiate between classes. Better differentiation between benign and attack traffic is shown by a higher ROC AUC value, which lowers the possibility of false negatives (FNs). On the other hand, the PR AUC gives the usability of a model under conditions where the detection of TP cases is highly desirable, while that of the FP is kept at a minimal value. This indicator is particularly applicable for unbalanced datasets, as it addresses the trade-off between precision and recall rather than considering true negatives (TNs). The predictive power, resilience, and general efficacy of each model in differentiating between benign and malicious network traffic are all verified through these indicators.

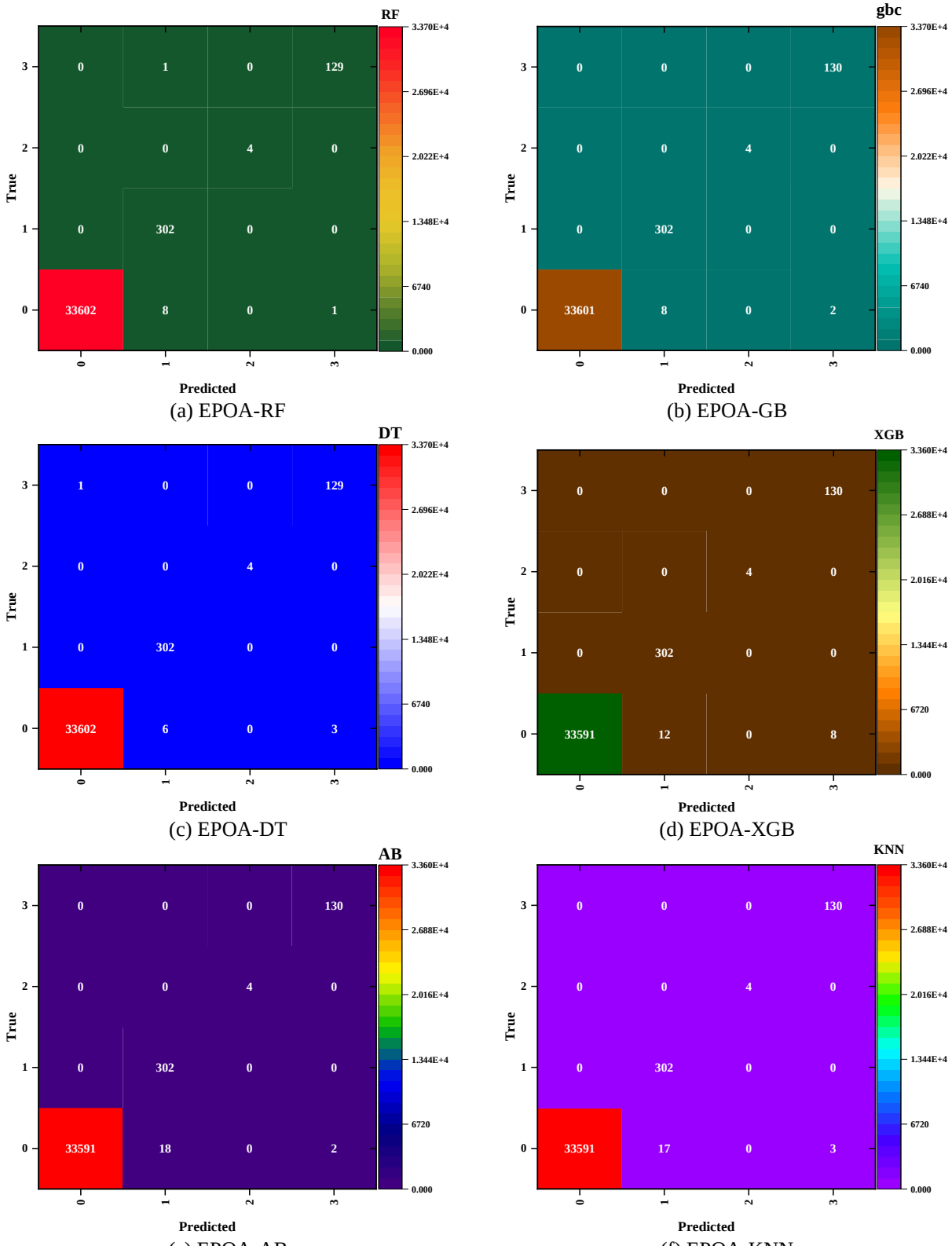
3 Results and discussion

3.1 Model comparison

In comparing and contrasting the employed models for network intrusion detection, a careful analysis was performed with several performance factors considered. Confusion matrices in Figure 7 present each model's classification output, indicating its capability to differentiate between benign and three types of attack traffic. In this figure, the target variable is categorized into four classes, with Benign labeled as 0, and Web Attack – Brute Force, Web Attack – XSS, and Web Attack – SQL Injection labeled as 1, 2, and 3, respectively.

The EPOA-RF model shows excellent performance, correctly classifying 33602 benign samples with only 9 misclassifications. All Brute Force and XSS attacks are correctly identified, while only one SQL Injection sample is misclassified. This reflects high precision and recall across all classes, especially the minority attack types. EPOA-GB performs similarly well, with 33601 benign samples correctly predicted and only 10 errors. It obtains satisfactory classification for all attack categories—Brute Force, XSS, and SQL Injection—indicating strength in handling both majority and minority classes. The EPOA-DT model correctly identifies all Brute Force and XSS samples and 129 out of 130 SQL Injection cases. It classifies 33602 benign samples accurately, with 9 minor misclassifications. The results highlight reliable performance across all classes, similar to RF and GB. While EPOA-XGB accurately classifies all attack categories, it shows slightly higher misclassification in the benign class, with 20 errors out of 33611 samples. Despite this, it maintains adequate detection for Brute Force, XSS, and SQL Injection attacks, confirming its strength in minority class recognition. The EPOA-AB model shows more errors in predicting class 0, with 20 misclassified instances. While class 1 is predicted without error, there are two instances of class 0 predicted as class 3. Although

generally strong, the model is slightly more prone to false positives compared to RF and DT. The KNN model mirrors AB's pattern, with 17 instances of class 0 misclassified and 3 as class 3. It performs properly for class 1 and class 2. Like AB, it achieves good overall accuracy but is less robust for the majority class than RF and DT. These observations provide insight into both data limitations (e.g., class imbalance and feature similarity) and model sensitivity.



(a) EPOA-RF (b) EPOA-GB

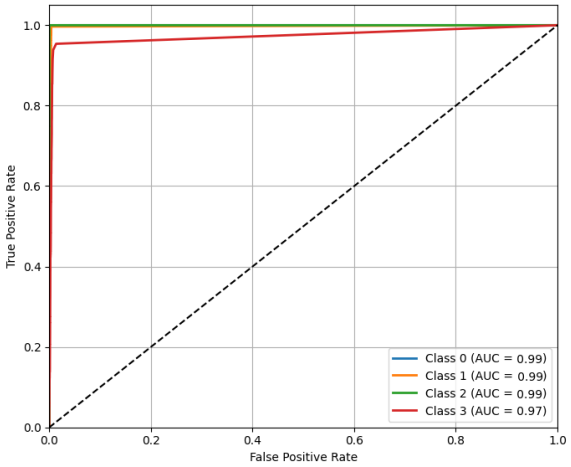
(c) EPOA-DT (d) EPOA-XGB

(e) EPOA-AB (f) EPOA-KNN

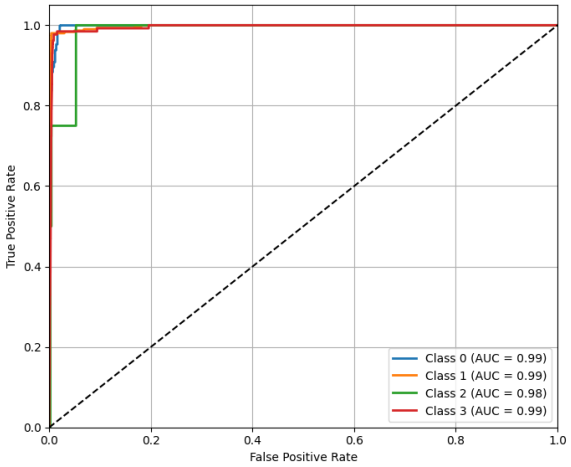
Figure 7: Confusion matrices for the applied hybrid models' prediction, including (a) EPOA-RF, (b) EPOA-GB, (c) EPOA-DT, (d) EPOA-XGB, (e) EPOA-AB, and (f) EPOA-KNN.

Moreover, the following analysis explicitly differentiates between ROC AUC and PR AUC metrics, highlighting the critical importance of PR AUC in intrusion detection scenarios due to its sensitivity to class imbalance and its direct reflection of the models' ability to maintain precision in identifying attack instances.

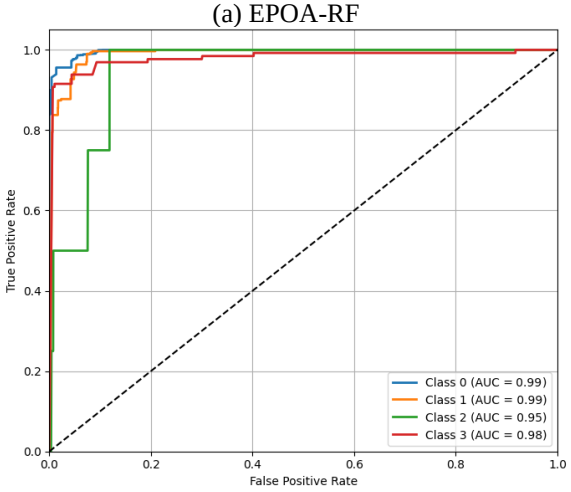
Figure 8 shows the ROC curves of various models applied to network intrusion detection with their performance on the test dataset. According to this figure, the EPOA-RF and EPOA-GB hybrid models exhibit an optimal performance with an AUC of above 0.97, considering all classes. This suitable score reflects that the benign and attack network traffic can be appropriately differentiated without misclassification during the test, representing the high predictive performance of such models. Also, EPOA-DT, EPOA-XGB, and EPOA-AB are performing optimally as well, as shown by their high AUC values above 0.87 for class 2, and AUC values above 0.94 for the other classes. The corresponding ROC curves of these models ascend nearly vertically to True Positive Rate (TPR) = 0.999 at False Positive Rate (FPR) close to 0, further establishing their dependability on cybersecurity applications. On the other hand, while the AUC values are very high in these models, showing their great discrimination power, the respective values of EPOA-KNN stand a little bit behind the others. Its ROC curve is lower with 0.75 for class 2, indicating some minor chance of misclassifications at particular thresholds.



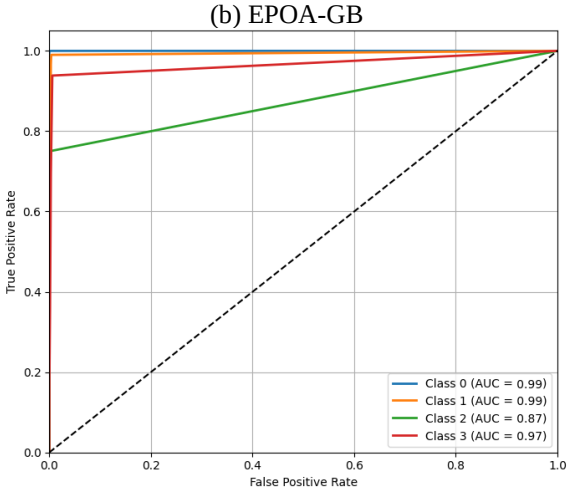
(a) EPOA-RF



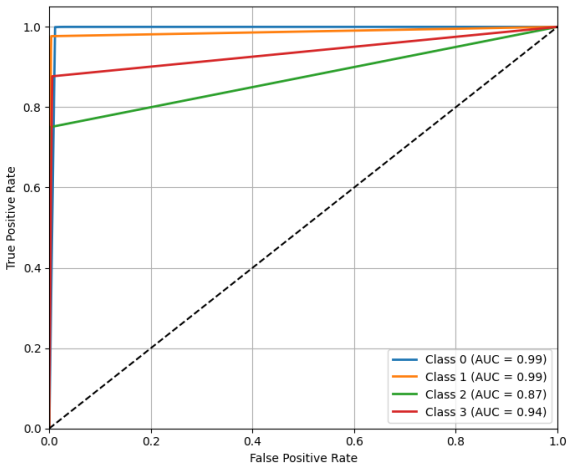
(b) EPOA-GB



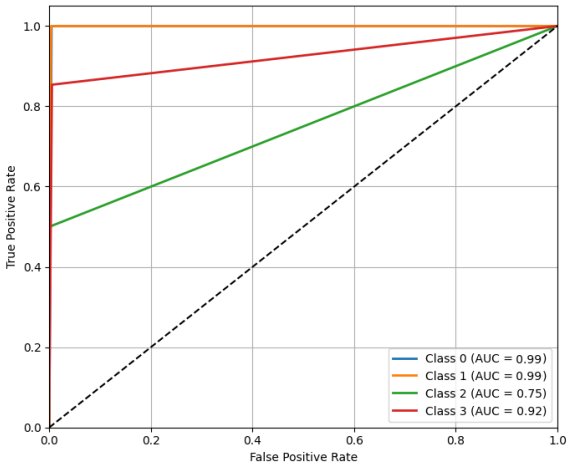
(c) EPOA-DT



(d) EPOA-XGB



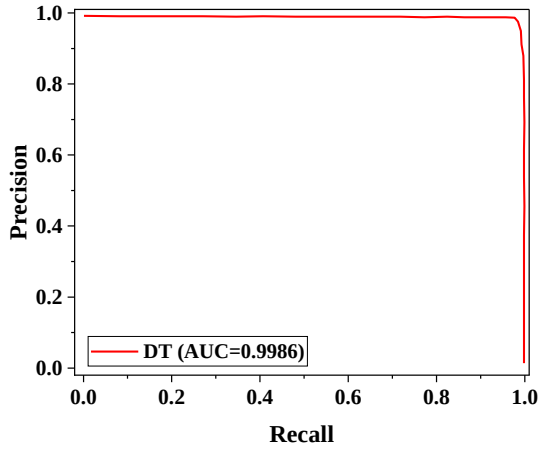
(e) EPOA-AB



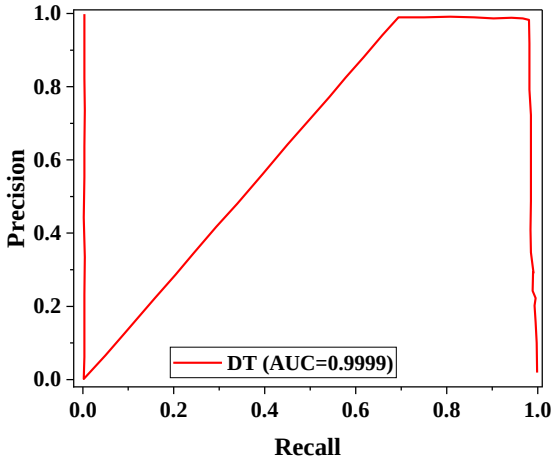
(f) EPOA-KNN

Figure 8: ROC curves for the applied hybrid models, including (a) EPOA-RF, (b) EPOA-GB, (c) EPOA-DT, (d) EPOA-XGB, (e) EPOA-AB, and (f) EPOA-KNN

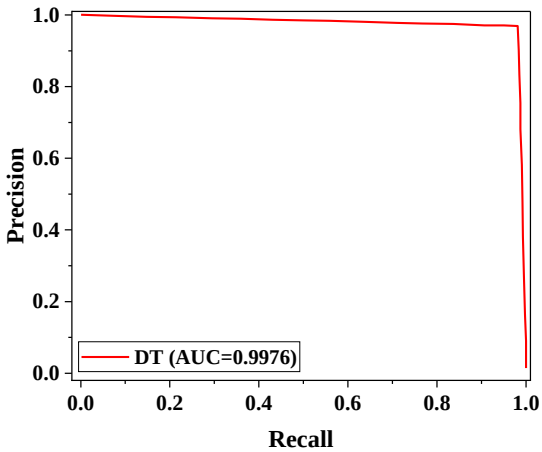
The Precision-Recall (PR) curves for all the models over the testing dataset have been displayed in Figure 9, with a critical examination of each model's performance in having high precision and high recall for network intrusion detection. Based on this figure, the AUC values of the PR curve of EPOA-GB, EPOA-RF, and EPOA-DT are all close to adequate performance above 0.99. They maintain consistently high precision at any recall value, proving that such models can effectively detect intrusion with negligible additional FPs. These experiments confirm not only high accuracy but also reliable and strong performance in real-world cybersecurity scenarios. The EPOA-KNN and EPOA-XGB perform moderately, with a value for PR AUC close to 0.98, marginally less but still high in effectiveness. However, EPOA-AB reaches a value for a PR AUC of 0.97, signifying a minor drop in accuracy at certain recall values.



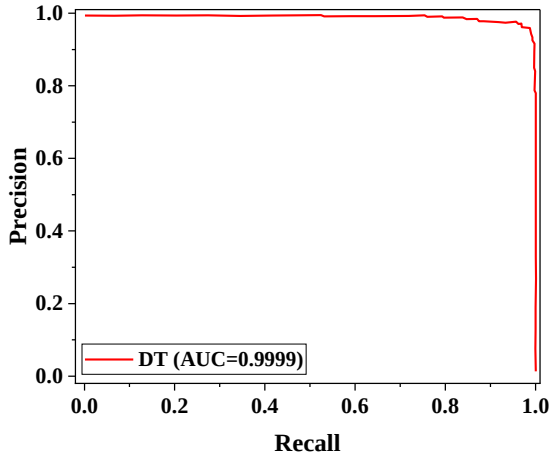
(a) EPOA-RF



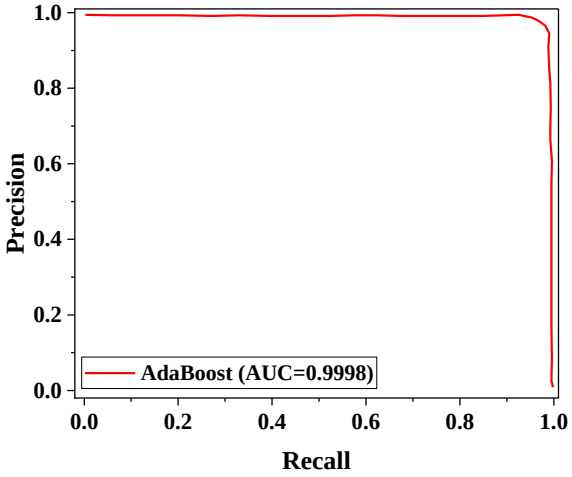
(b) EPOA-GB



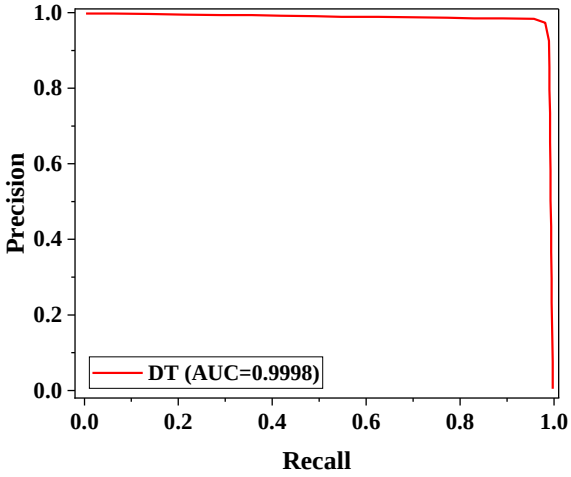
(c) EPOA-DT



(d) EPOA-XGB



(e) EPOA-AB



(f) EPOA-KNN

Figure 9: PR curves for the applied hybrid models, including (a) EPOA-RF, (b) EPOA-GB, (c) EPOA-DT, (d) EPOA-XGB, (e) EPOA-AB, and (f) EPOA-KNN

Table 4 presents the performance metrics of the applied hybrid models tuned using EPOA on both train and test. Evaluated on the test dataset, all models exhibited outstanding classification performance, with test accuracies exceeding 0.9994. Among them, the EPOA-GB model demonstrated the best overall performance, achieving a test accuracy of 0.9997, F1-score of 0.9948, ROC AUC of 0.9999, and the highest Cohen's Kappa value of 0.9885. Additionally, EPOA-GB had the lower Log Loss (0.0050) among most of the models, indicating both high confidence and low prediction error. These results highlight the effectiveness of EPOA in achieving near-perfect classification performance, with EPOA-GB standing out as the most robust model under this tuning approach.

Table 5 illustrates the performance of the hybrid models tuned using Grid Search (GS) on both train and test. Based on the test dataset, although the models achieved generally high accuracies above 0.994, their performance metrics—particularly precision, recall, and F1-score—were significantly lower than their EPOA-tuned counterparts. The GS-GB model emerged as the best among the GS-tuned models, with the highest test accuracy of 0.9961 and a relatively balanced F1-score of 0.5432, ROC AUC of 0.9904, and Cohen's Kappa of 0.8481. However, compared to EPOA-GB, the GS-GB model had higher Log loss (0.0138) and considerably lower precision and recall, indicating reduced reliability and robustness. These findings reinforce the superiority of EPOA in both predictive performance and optimization efficiency.

Table 4: Statistical results of the hybrid models tuned by EPOA.

	Accuracy	**Precision**	**Recall**	**F1 Score**	**ROC AUC**	**PR AUC**	**Log loss**	**MCC**	**Cohen's Kappa**	**Runtime (sec)**
Train										
EPOA-RF	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.0000	0.9999	0.9999	10206.15
EPOA-GB	0.9999	0.9999	0.9965	0.9982	0.9968	0.9965	0.0010	0.9962	0.9962	100361.13
EPOA-DT	0.9999	0.9963	0.9999	0.9981	0.9999	0.9968	0.0001	0.9971	0.9971	1002.1
EPOA-XGB	0.9999	0.9990	0.9972	0.9981	0.9976	0.9967	0.0030	0.9962	0.9962	2010.57
EPOA-AB	0.9998	0.9915	0.9999	0.9957	0.9999	0.9923	0.0006	0.9914	0.9914	15009.47
EPOA-KNN	0.9998	0.9907	0.9999	0.9953	0.9999	0.9906	0.0003	0.9914	0.9914	10209.86
Test										
EPOA-RF	0.9997	0.9908	0.9980	0.9944	0.9986	0.9919	0.0020	0.9886	0.9885	10206.15
EPOA-GB	**0.9997**	**0.9898**	**0.9999**	**0.9948**	**0.9999**	**0.9924**	**0.0050**	**0.9886**	**0.9885**	**100361.13**

EPOA-DT	0.9997	0.9894	0.9980	0.9937	0.9976	0.9910	0.0110	0.9886	0.9885	1002.1
EPOA-XGB	0.9994	0.9760	0.9999	0.9876	0.9999	0.9799	0.0020	0.9776	0.9773	2010.57
EPOA-AB	0.9994	0.9821	0.9999	0.9908	0.9998	0.9746	0.4190	0.9776	0.9773	15009.47
EPOA-KNN	0.9994	0.9810	0.9999	0.9902	0.9998	0.9836	0.4190	0.9776	0.9773	10209.86

Table 5: Statistical results of the hybrid models tuned by Grid Search (GS).

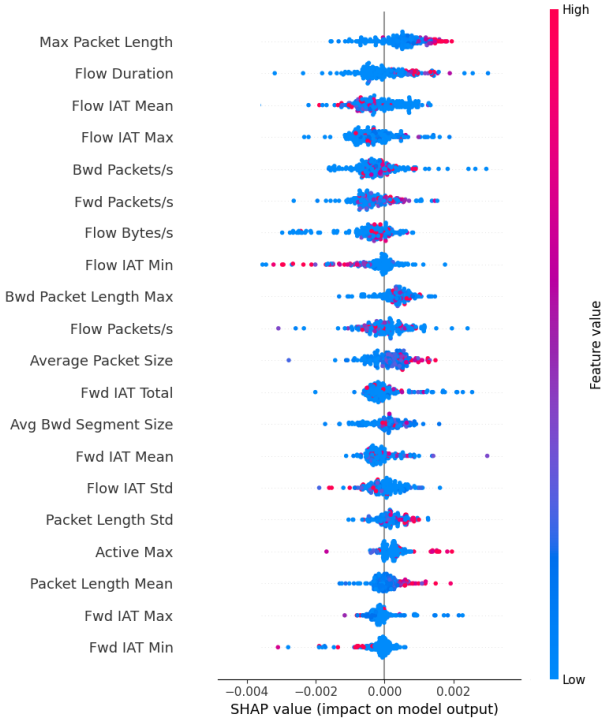
	Accuracy	**Precision**	**Recall**	**F1 Score**	**ROC AUC**	**PR AUC**	**Log loss**	**MCC**	**Cohen's Kappa**	**Runtime (sec)**
Train										
GS-RF	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.0021	0.9999	0.9999	90.8311
GS-GB	0.9973	0.9079	0.8433	0.8532	0.9996	0.9993	0.0059	0.8952	0.8950	980.4113
GS-DT	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	2.2E-6	0.9999	0.9999	8.7119
GS-XGB	0.9955	0.6705	0.5005	0.4670	0.9989	0.9636	0.0107	0.8191	0.8185	15.5305
GS-AB	0.9942	0.4133	0.4503	0.4300	0.9927	0.9178	1.1639	0.7535	0.7500	114.8171
GS-KNN	0.9971	0.8764	0.6551	0.6946	0.9994	0.9863	0.0059	0.8833	0.8833	93.5015
Test										
GS-RF	0.9954	0.5427	0.5420	0.5418	0.9757	0.9828	0.0125	0.8202	0.8201	90.8311
GS-GB	**0.9961**	**0.5631**	**0.5410**	**0.5432**	**0.9904**	0.9834	**0.0138**	**0.8482**	**0.8481**	**980.4113**
GS-DT	0.9952	0.5424	0.5578	0.5478	0.7763	0.9638	0.1747	0.8140	0.8139	8.7119
GS-XGB	0.9960	0.6860	0.5127	0.4973	0.9956	0.9384	0.0116	0.8446	0.8442	15.5305
GS-AB	0.9945	0.4295	0.4491	0.4388	0.9823	0.9044	1.1642	0.7705	0.7657	114.8171
GS-KNN	0.9957	0.5469	0.5428	0.5444	0.9014	0.9530	0.0268	0.8339	0.8339	93.5015

3.2 Sensitivity analysis

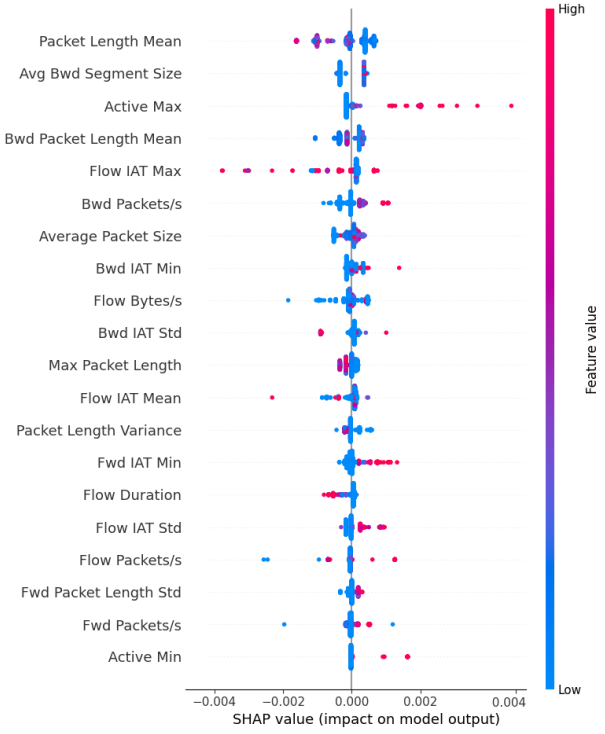
Figure 10 demonstrates SHAP summary of input features' impact on the hybrid models' output. According to this figure, the hybrid models' SHAP values range from -0.004 to +0.003. The EPOA-RF and EPOA-AB models show the lowest variability and features' impact among the other models, explaining lower interpretability or weaker feature separation compared to other models. The SHAP values for these models are relatively small, indicating that the individual feature contributions to the output are subtle.

The EPOA-GB model, on the other hand, demonstrates the highest variability and feature impact among the other models. Flow IAT Max, Active Max, and Packet Length Mean are the most significant features. Their wide SHAP value ranges show that EPOA-GB is highly sensitive to changes in these key features and it utilizes these characteristics, making it well-suited for complex, nonlinear data patterns. Therefore, the strong influence of a few dominant features supports GB's reputation for high accuracy and interpretability through clear feature attributions.

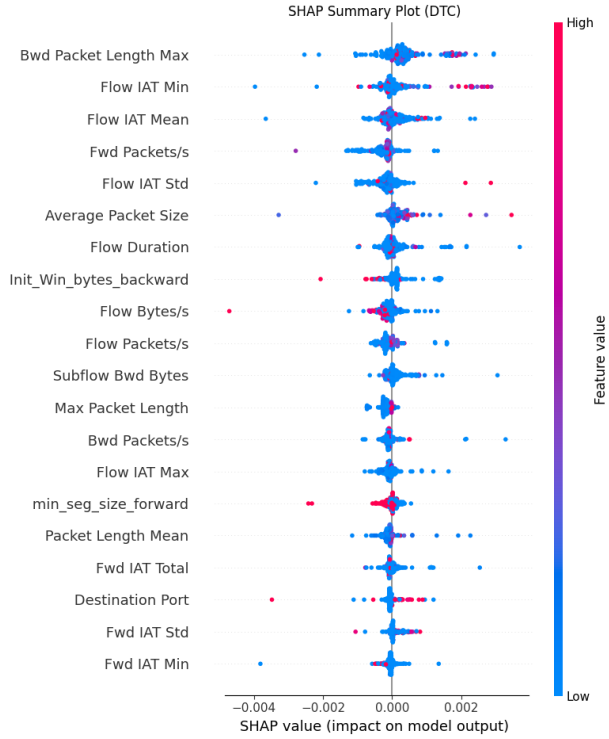
EPOA-XGB shows the next higher variability and feature impact. The most impactful features include Average Packet Size and Destination Port. The SHAP value spread here is broader than in the EPOA-RF and EPOA-AB plots, showing clearer separation of feature contributions. The EPOA-KNN and EPOA-DT models also appear relatively high in their SHAP values' variability, suggesting that these models' decisions are moderately sensitive to changes in a few dominant features and indicating more decisive rule-based splits.



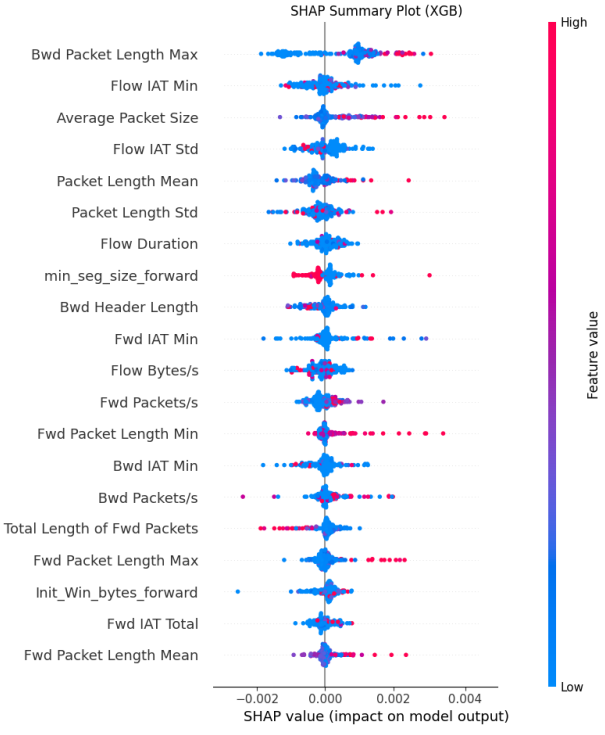
EPOA-RF



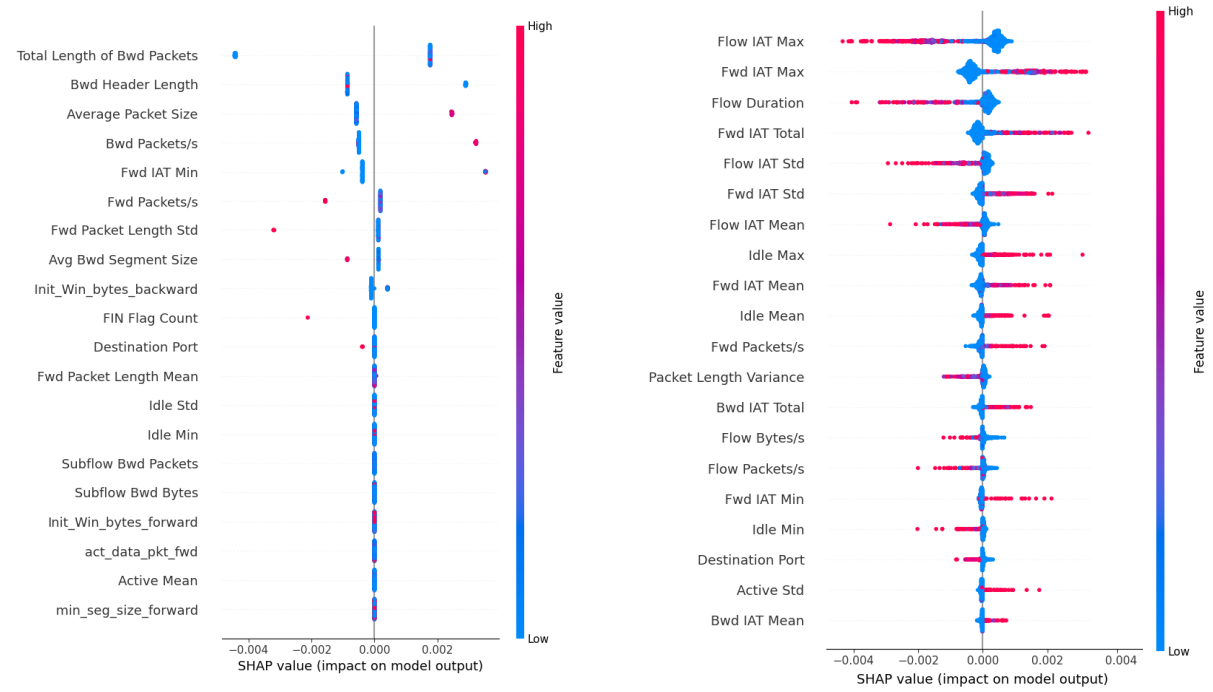
EPOA-GB



EPOA-DT



EPOA-XGB



EPOA-AB EPOA-KNN

Figure 10: SHAP summary of input features' impact on the employed hybrid models' output.

4 Conclusion

In order to improve the efficiency of Intrusion Detection Systems (IDSs) in differentiating between malicious and benign network activity, this study provided a thorough evaluation of a number of ML classifiers. One of the main contributions is the use of sophisticated hyperparameter optimization methods, specifically the Emperor Penguin Optimization Algorithm (EPOA) and Grid Search (GS), to improve the predictive performance of six selected classifiers: Random Forest (RF), Gradient Boosting (GB), Decision Tree (DT), XGBoost (XGB), AdaBoost (AB), and K-Nearest Neighbors (KNN). With the highest testing accuracy, precision, recall, and F1-score values of 0.9997, 0.9898, 0.9999, and 0.9948, respectively, the optimized models—EPOA-GB in particular—showed exceptional classification abilities. SHAP-based feature importance analysis was carried out to promote model transparency by identifying crucial network attributes affecting classification and providing interpretability into the decision-making process. These insights can help cybersecurity experts improve monitoring rules and prioritization strategies by revealing which traffic features are most suggestive of threats.

Practically speaking, the results highlight how effective it is to combine intelligent optimization with powerful ensemble classifiers to enhance detection performance while reducing false alarms, which is an essential prerequisite for real-time intrusion response systems. Additionally, the application of PCA for dimensionality reduction enhanced the approach's scalability by improving generalization and lowering computational load.

It is necessary to recognize a few limitations, though. First, the evaluation was limited to a single dataset, which might not fully represent the variety of contemporary or developing cyberattack techniques, even though there was strong within-dataset generalization. Second, real-world traffic frequently contains previously unseen anomalies that could test the robustness of the model, even though retraining and testing on unseen splits were done to evaluate generalizability. Third, even though EPOA works well, it has a significant computational overhead during training, which might restrict its direct use in environments with limited resources or real-time deployment.

Future studies should validate the suggested models on more benchmark datasets, like UNSW-NB15 and TON_IoT, to evaluate their resilience across a range of network conditions and attack types in order to overcome these drawbacks and facilitate real-world implementation. To lessen latency and computational load during live deployment, integration with edge computing environments, real-time streaming frameworks, and lightweight optimization techniques is also advised. Moreover, hybrid models that combine evolutionary optimization and deep learning can be investigated to capture intricate attack behaviors while preserving flexibility.

In conclusion, this study demonstrates that combining strong ML models with sophisticated optimization can greatly improve IDS performance. Continuous evaluation in dynamic and heterogeneous environments is crucial for practical adoption, as are attempts to strike a balance between model accuracy, interpretability, and efficiency for cybersecurity applications in the real world.

Acknowledgement

This Work was supported by the Hubei Provincial Education and Science Planning Project. (2018GB081)

References

[1] T. H. Chua and I. Salam, "Evaluation of Machine Learning Algorithms in Network-Based Intrusion Detection Using Progressive Dataset," *Symmetry (Basel)*, MDPI, vol. 15, no. 6, Jun. 2023, https://doi.org/10.3390/sym15061251.

[2] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of network and computer applications*, Elsevier, vol. 36, no. 1, pp. 16–24, 2013, https://doi.org/10.1016/j.jnca.2012.09.004.

[3] R. A. Ramadan and K. Yadav, "A novel hybrid intrusion detection system (IDS) for the detection of internet of things (IoT) network attacks," *Annals of Emerging Technologies in Computing (AETiC)*, AETIC, vol. 4, no. 5, pp. 61–74, 2020, DOI: 10.33166/AETiC.2020.05.004.

[4] A. H. Azizan *et al.*, "A Machine Learning Approach for Improving the Performance of Network Intrusion Detection Systems," *Annals of Emerging Technologies in Computing*, AETIC, vol. 5, no. 5, pp. 201–208, Mar. 2021, doi: 10.33166/AETiC.2021.05.025.

[5] J. Jose and D. V. Jose, "Deep learning algorithms for intrusion detection systems in internet of things using CIC-IDS 2017 dataset," *International Journal of Electrical and Computer Engineering*, Academia, vol. 13, no. 1, pp. 1134–1141, Feb. 2023, DOI: 10.11591/ijece.v13i1.pp1134-1141.

[6] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecurity*, vol. 4, pp. 1–27, 2021, DOI: 10.11591/ijece.v13i1.pp1134-1141.

[7] M. A. Khan, M. R. Karim, and Y. Kim, "A scalable and hybrid intrusion detection system based on the convolutional-LSTM network," *Symmetry (Basel)*, MDPI, vol. 11, no. 4, p. 583, 2019, https://doi.org/10.3390/sym11040583.

[8] M. Sarnovsky and J. Paralic, "Hierarchical intrusion detection using machine learning and knowledge model," *Symmetry (Basel)*, MDPI, vol. 12, no. 2, p. 203, 2020, https://doi.org/10.3390/sym12020203.

[9] C. Wang, Y. Sun, W. Wang, H. Liu, and B. Wang, "Hybrid intrusion detection system based on combination of random forest and autoencoder," *Symmetry (Basel)*, MDPI, vol. 15, no. 3, p. 568, 2023, https://doi.org/10.3390/sym15030568.

[10] F. Hossain, M. Akter, and M. N. Uddin, "Cyber attack detection model (CADM) based on machine learning approach," in *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, DHAKA, Bangladesh, IEEE, 2021, pp. 567–572, https://doi.org/10.1109/ICREST51555.2021.9331094.

[11] O. Malkawi, N. Obaid, and W. Almobaideen, "Intrusion Detection System for 5G Device-to-Device Communication Technology in Internet of Things," *Informatica*, Slovenian Society Informatika, vol. 48, no. 15, 2024, https://doi.org/10.31449/inf.v48i15.4646.

[12] S. Ahmad, F. Arif, Z. Zabeehullah, and N. Iltaf, "Novel approach using deep learning for intrusion detection and classification of the network traffic," in *2020 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, Tunis, Tunisia, IEEE, 2020, pp. 1–6, https://doi.org/10.1109/CIVEMSA48639.2020.9132744.

[13] I. F. Kilincer, F. Ertam, and A. Sengur, "Machine learning methods for cyber security intrusion detection: Datasets and comparative study," *Computer Networks*, Elsevier, vol. 188, p. 107840, 2021, https://doi.org/10.1016/j.comnet.2021.107840.

[14] A. Ali *et al.*, "Network intrusion detection leveraging machine learning and feature selection," in *HONET 2020 - IEEE 17th International Conference on Smart Communities: Improving Quality of Life using ICT, IoT and AI*, Institute of Electrical and Electronics Engineers Inc., Charlotte, NC, USA, IEEE, Dec. 2020, pp. 49–53, https://doi.org/10.1109/HONET50430.2020.9322813.

[15] D. Stiawan, M. Y. Bin Idris, A. M. Bamhdi, and R. Budiarto, "CICIDS-2017 dataset feature analysis with information gain for anomaly detection," *IEEE Access*, IEEE, vol. 8, pp. 132911–132921, 2020, https://doi.org/10.1109/ACCESS.2020.3009843.

[16] A. Bansal and S. Kaur, "Extreme gradient boosting based tuning for classification in intrusion detection systems," in *Advances in Computing and Data Sciences: Second International Conference, ICACDS 2018, Dehradun, India, April 20-21, 2018, Revised Selected Papers, Part I 2*, Singapore, Springer, 2018, pp. 372–380, https://doi.org/10.1007/978-981-13-1810-8_37.

[17] I. A. Abdulmajeed and I. M. Husien, "MLIDS22-IDS design by applying hybrid CNN-lstm model on mixed-datasets," *Informatica*, Slovenian Society Informatika, vol. 46, no. 8, 2022, https://doi.org/10.31449/inf.v46i8.4348.

[18] M. Guo, D. Ma, F. Jing, X. Zhang, and H. Liu, "Dynamic Anti-Mapping Network Security Using Hidden Markov Models and LSTM Networks Against Illegal Scanning," *Informatica*, Slovenian

Society Informatika, vol. 49, no. 12, 2025, https://doi.org/10.31449/inf.v49i12.6903.

[19] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization.," *ICISSp*, Scite Press, vol. 1, pp. 108–116, 2018, DOI: 10.5220/0006639801080116.

[20] L. Breiman, "Random forests," *Mach Learn*, Springer, vol. 45, pp. 5–32, 2001, https://doi.org/10.1023/A:1010933404324.

[21] A. Cutler, D. R. Cutler, and J. R. Stevens, "Random Forests," in *Ensemble Machine Learning*, New York, NY: Springer New York, 2012, pp. 157–175, https://doi.org/10.1007/978-1-4419-9326-7_5.

[22] Y. Shi, V. Charles, and J. Zhu, "Bank financial sustainability evaluation: Data envelopment analysis with random forest and Shapley additive explanations," *Eur J Oper Res*, Elsevier, vol. 321, no. 2, pp. 614–630, 2025, https://doi.org/10.1016/j.ejor.2024.09.030.

[23] J. H. Friedman, "999 REITZ LECTURE GREEDY FUNCTION APPROXIMATION: A GRADIENT BOOSTING MACHINE 1," 2001, https://www.jstor.org/stable/2699986.

[24] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Front Neurorobot*, Frontiers, vol. 7, no. DEC, 2013, https://doi.org/10.3389/fnbot.2013.00021.

[25] B. Charbuty and A. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning," *Journal of Applied Science and Technology Trends*, jastt, vol. 2, no. 01, pp. 20–28, Mar. 2021, https://doi.org/10.38094/jastt20165.

[26] R. Oktafiani, A. Hermawan, and D. Avianto, "Max Depth Impact on Heart Disease Classification: Decision Tree and Random Forest," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, Journal IAII, vol. 8, no. 1, pp. 160–168, 2024, https://doi.org/10.29207/resti.v8i1.5574.

[27] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, ACM Digital Library, 2016, pp. 785–794, https://doi.org/10.1145/2939672.2939785.

[28] A. Sharma and R. Tiwari, "Anomaly detection in smart grid using optimized extreme gradient boosting with SCADA system," *Electric Power Systems Research*, Elsevier, vol. 235, p. 110876, 2024, https://doi.org/10.1016/j.epsr.2024.110876.

[29] C. Lee and E. D. C. Maceren, "Wind energy system fault classification and detection using deep convolutional neural network and particle swarm optimization-extreme gradient boosting," *IET Energy Systems Integration*, Wiley Online Library, vol. 6, no. 4, pp. 479–497, 2024, https://doi.org/10.1049/esi2.12144.

[30] Y. Freund and R. E. Schapire, "Journal of Computer and System Sciences s SS1504 journal of computer and system sciences," 1997.

[31] L. Wen, Y. Li, W. Zhao, W. Cao, and H. Zhang, "Predicting the deformation behaviour of concrete face rockfill dams by combining support vector machine and AdaBoost ensemble algorithm," *Comput Geotech*, Elsevier, vol. 161, Sep. 2023, https://doi.org/10.1016/j.compgeo.2023.105611.

[32] L. Wang, Y. Guo, M. Fan, and X. Li, "Wind speed prediction using measurements from neighboring locations and combining the extreme learning machine and the AdaBoost algorithm," *Energy Reports*, Elsevier, vol. 8, pp. 1508–1518, Nov. 2022, https://doi.org/10.1016/j.egyr.2021.12.062.

[33] E. Fix, *Discriminatory analysis: nonparametric discrimination, consistency properties*, vol. 1. USAF school of Aviation Medicine, 1985.

[34] P. Cunningham and S. J. Delany, "K-Nearest Neighbour Classifiers-A Tutorial," Jul. 31, 2021, *Association for Computing Machinery*, Cornell University, https://doi.org/10.48550/arXiv.2004.04523.

[35] R. K. Halder, M. N. Uddin, Md. A. Uddin, S. Aryal, and A. Khraisat, "Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications," *J Big Data*, Springer, vol. 11, no. 1, p. 113, 2024, https://doi.org/10.1186/s40537-024-00973-y.

[36] G. Dhiman and V. Kumar, "Emperor penguin optimizer: A bio-inspired algorithm for engineering problems," *Knowl Based Syst*, Elsevier, vol. 159, pp. 20–50, Nov. 2018, https://doi.org/10.1016/j.knosys.2018.06.001.

[37] O. W. Khalid, N. A. M. Isa, and H. A. Mat Sakim, "Emperor penguin optimizer: A comprehensive review based on state-of-the-art meta-heuristic algorithms," Feb. 01, 2023, *Elsevier B.V.* https://doi.org/10.1016/j.aej.2022.08.013.