e-ISSN: 2614-8404
p-ISSN: 2776-3234

Interpretable Ensemble-Based Intrusion Detection Using Feature Selection on the ToN_IoT Dataset

Vaman Shakir Sulaiman[1*)], Firas Mahmood Mustafa[2]
[1]Department of Computer Information System, Technical College of Zakho, Duhok Polytechnic University, Zakho, Kurdistan Region of Iraq
[2]Technical College of Engineering, Duhok Polytechnic University, Duhok, Kurdistan Region of Iraq
Email: [1]vaman.sulaiman@dpu.edu.krd, [2]firas.mahmoud@dpu.edu.krd

Abstract – With With the rapid growth of IoT, securing interconnected devices against cyber threats has become critical. IoT datasets such as ToN-IoT are often high-dimensional, which poses challenges for efficient and accurate intrusion detection. Moreover, interpretable models are essential to help security analysts understand and trust automated decisions. Intrusion Detection Systems (IDS) powered by machine learning offer promising solutions, especially when trained on realistic datasets such as ToN_IoT. However, achieving a balance between high accuracy, computational efficiency, and model interpretability remains a challenge. This study proposes an efficient and interpretable IDS framework for binary classification using the ToN_IoT dataset, aiming to identify the optimal feature selection method and ensemble learning model while leveraging explainable artificial intelligence to interpret model decisions. A quantitative experimental approach was adopted, applying and comparing Principal Component Analysis (PCA) and Recursive Feature Elimination (RFE) for feature selection, and evaluating the performance of LightGBM, XGBoost, and Random Forest classifiers using Accuracy, F1-score, Precision, Recall, and training time. RFE outperformed PCA, identifying 11 key features, and LightGBM emerged as the top-performing model with an accuracy of 99.72%, demonstrating both speed and strong generalization. SHAP (SHapley Additive exPlanations) was used to generate summary plots for global feature importance, enhancing the transparency and interpretability of IDS decisions. Overall, the combination of RFE and LightGBM resulted in a high-performing and explainable IDS framework, underscoring the importance of strategic feature selection and model choice. Compared to existing IDS approaches on the ToN-IoT dataset, our proposed framework not only achieves higher accuracy but also provides a rapid and lightweight solution. Additionally, by incorporating SHAP for feature importance analysis, our approach ensures clear model interpretability, allowing security analysts to understand and trust the system's decisions. This combination of high performance, efficiency, and explainability highlights the practical advantages of our method over previous work. Future research will extend this framework to support multiclass classification and online learning for real-time threat detection.

Keywords – LightGBM, Intrusion Detection System, XAI, SHAP, Ensemble Learning, RFE

I. INTRODUCTION

The proliferation of the Internet of Things (IoT) has revolutionized modern computing environments by enabling seamless interconnectivity between smart devices. However, this digital transformation has introduced severe cybersecurity vulnerabilities, particularly in critical infrastructure systems where reliability and security are paramount. Recent reports indicate that IoT-malware attacks rose by 400% in the first half of 2023 compared to 2022 [1], and attacks targeting IoT devices further surged by 107% in early 2024 compared to the same period in 2023 [2]. Due to the heterogeneous and resource-constrained nature of IoT environments, traditional security mechanisms are insufficient, necessitating the deployment of intelligent and adaptive Intrusion Detection Systems (IDS) [3], [4]. These systems must not only detect known and novel threats but also remain interpretable and scalable across diverse IoT scenarios.

To evaluate IDS models, high-quality datasets that reflect real-world network activity are crucial. The ToN_IoT dataset, developed by the Cyber Range Lab at UNSW [5], is a recent and robust dataset that captures telemetry from IoT devices, operating systems, and network traffic. It includes multiple attack types and supports both binary and multiclass classification, making it particularly suitable for training machine learning-based IDS frameworks [6]. Its realism and multimodal nature make it one of the most reliable benchmarks for intrusion detection in modern IoT environments.

Given the complexity and high dimensionality of IDS datasets, effective Feature Selection (FS) is essential to reduce model complexity, improve classification accuracy, and shorten training time. Two widely adopted techniques in IDS research are Principal Component Analysis (PCA) and Recursive Feature Elimination (RFE). PCA reduces dimensionality by transforming the original variables into a smaller set of orthogonal components while preserving most of the variance in the data. This can mitigate overfitting and improve generalization in high-dimensional spaces [7]. In contrast, RFE works by recursively training a model and eliminating the least important features based on model performance, making it particularly effective when paired with ensemble methods [8], [9]. In the context of this study, PCA and RFE are especially relevant as they address the high dimensionality and redundancy of the ToN-IoT dataset, enabling the development of IDS models that are both computationally efficient and interpretable. Moreover, these techniques help reduce multicollinearity and redundant attributes, while simplifying the feature space for improved interpretability.

In addition to enhancing performance, ensemble learning techniques have gained popularity due to their robustness and superior generalization ability. Random Forest (RF), LightGBM, and XGBoost are particularly well-suited for intrusion detection tasks due to their ability to model complex feature interactions, handle imbalanced data, and resist overfitting [10]. When combined with optimized feature sets from PCA or RFE, these models deliver faster convergence and stronger decision boundaries, making them ideal for real-time IDS deployment.

Despite their effectiveness, ensemble models are often perceived as “black boxes” which limits their applicability in sensitive or regulated environments. To address this, eXplainable AI (XAI) tools like SHAP (SHapley Additive exPlanations) are increasingly integrated into IDS pipelines. SHAP provides both global and local interpretability by quantifying the contribution of each feature to a prediction. This interpretability is vital in cybersecurity, where human analysts must understand the rationale behind alerts to validate and respond effectively [11].

Although several IDS studies have been conducted on the ToN-IoT dataset, most focus mainly on achieving high accuracy, with limited attention to computational efficiency and model interpretability. Many existing models are complex and resource-intensive, making them unsuitable for real-time or constrained IoT environments. To address these gaps, this study proposes a lightweight and interpretable IDS framework that integrates Recursive Feature Elimination (RFE) with LightGBM for fast and accurate detection, enhanced by SHAP-based explainability to ensure transparency and trust in model decisions.

In this paper, we propose an intrusion detection framework that utilizes an ensemble of RF, LightGBM, and XGBoost classifiers for binary classification using the ToN_IoT dataset. To evaluate the influence of feature dimensionality on model performance, we apply and compare two distinct FS techniques: PCA and RFE. PCA is employed to reduce feature dimensionality through unsupervised variance-preserving transformations, while RFE is used as a supervised method that recursively eliminates less relevant features based on model feedback. The system is further enhanced with SHAP to provide a global explanation of feature importance using summary plots, helping to visualize and interpret the contribution of features across the dataset. Our contributions are fourfold: (1) a comprehensive performance evaluation of ensemble classifiers on the ToN_IoT dataset, (2) a comparative analysis of PCA and RFE feature selection methods, (3) integration of SHAP summary plots for global interpretability, and (4) visualization of key features contributing to attack classification to aid in real-time cybersecurity triage. Experimental results demonstrate that the proposed method achieves high detection accuracy while maintaining interpretability and computational efficiency.

Several studies have utilized the ToN-IoT dataset to develop an IDS capable of distinguishing between normal and abnormal network activities; for example, the researchers in [12] proposed an ensemble-learning framework for building an IDS using the TON-IoT dataset. Their approach began with data preprocessing, which included cleaning, label encoding, normalization, and train/test splitting. To further enhance performance, they applied FS using Mutual Information (MI), Pearson Correlation Coefficient (PCC), and K-Best methods, resulting in a final reduced subset of features. Four supervised Machine Learning (ML) classifiers, including RF, Decision Tree (DT), Logistic Regression (LR), and K-Nearest Neighbor (KNN) were trained on the dataset. These base models were then combined using two ensemble strategies, Voting and Stacking, to improve detection accuracy. Experimental results showed that while the individual classifiers achieved strong performance (e.g., LR accuracy 98.42%), the stacking ensemble with LR as meta-learner outperformed all others, achieving an Accuracy of 98.63%, Precision of 98.20%, Recall of 98.60%, and F1-score of 98.61. Similarly, the researchers in [13] proposed a hybrid IDS by integrating 1D Convolutional Neural Networks (CNN) with Long Short-Term Memory (LSTM). To enhance model performance, the authors applied FS using the PCC. The model was evaluated against ten traditional ML and DL algorithms, including RF, DT, AdaBoost, LR, KNN, CNN, Multi-Layer Perceptron (MLP), and LSTM, using Accuracy, Precision, Recall, and F1-score as evaluation metrics. Experimental results demonstrated that the hybrid CNN-LSTM approach outperformed all baselines, achieving an Accuracy of 98.75%, with correspondingly high Precision, Recall, and F1-score values for binary classification on the TON-IoT dataset. In [14] the researchers explored the optimization of IoT IDS by comparing feature selection (PCC-based correlation) and feature extraction (PCA) techniques. Using the TON-IoT dataset, the authors conducted preprocessing that included duplicate removal, categorical feature encoding, normalization, and stratified 80/20 data splitting. Five classifiers, including DT, RF, KNN, Naive Bayes (NB), and MLP were employed to evaluate the two feature reduction methods under binary classification. Findings demonstrated that while FS achieved lower training and inference times, Feature Extraction (FE) with PCA consistently produced higher detection accuracy. Specifically, the KNN classifier with PCA reached the best performance, achieving an Accuracy of 89.10%, Precision of 87.78%, Recall of 89.28%, and F1-score of 88.39% for binary classification on the TON-IoT dataset. Another study by [15] proposed a Secured Automatic Two-level IDS (SATIDS) leveraging an improved LSTM network for IoT environments. The preprocessing phase involved removing IP address and time-stamp features, followed by a 70/30 train-test split of the dataset. The model was evaluated on the TON-IoT dataset for binary classification, distinguishing between normal and attack traffic. Evaluation metrics included Accuracy, Precision, Detection Rate (Recall), and F1-score. Findings showed that the proposed SATIDS achieved strong performance, recording an Accuracy of 96.35%, Precision of 98.4%, Detection Rate of 96%, and F1-score of 97.35%, surpassing traditional IDS models tested on the same dataset. The researchers in [16] developed an IDS using DL models. Three models were implemented: a customized 1D-CNN, a Deep Neural Network (DNN), and the pre-trained TabNet model. The TON-IoT dataset was used, with preprocessing steps including handling missing values, duplicate removal, normalization, and label encoding. Evaluation metrics comprised Accuracy, Precision, Recall, and F1-score. Experimental findings revealed that the CNN model on the TON-IoT network dataset achieved the strongest performance, with an Accuracy, Precision, Recall, and F1-score of 99.24%, 98%, 98%, and 98%, respectively, while the DNN model followed closely with 99.03% accuracy. The results demonstrate that CNN outperformed both DNN and TabNet. In [17] the authors conducted a comparative study of FS and FE techniques for intrusion detection in IoT systems. The preprocessing phase on the TON-IoT dataset involved handling missing values, duplicate removal, categorical encoding, normalization, and an 80/20 stratified split. FS methods included Pearson Correlation and Chi-square, while FE methods comprised PCA and Autoencoders (AE). Multiple classifiers were tested, including DT, RF, NB, KNN, and Multi-Layer Perceptron (MLP). For binary classification, results showed that FE consistently outperformed FS, with the RF combined with Autoencoder (AE) achieving the highest performance: Accuracy = 88.66%, Precision 88%, Recall 88%, and F1-score 88%, thereby surpassing all FS-based approaches.

II. Research Methodology

The methodology adopted in this work is designed to systematically preprocess the ToN_IoT dataset, reduce data redundancy, extract meaningful representations, and train high-performing ensemble classifiers. The entire pipeline consists of six main stages: (i) dataset preprocessing, (ii) feature engineering, (iii) feature selection and dimensionality reduction, (iv) representation learning using an autoencoder, (v) model training using ensemble methods, and (vi) model evaluation with explainability. Figure 1 illustrates the workflow diagram.

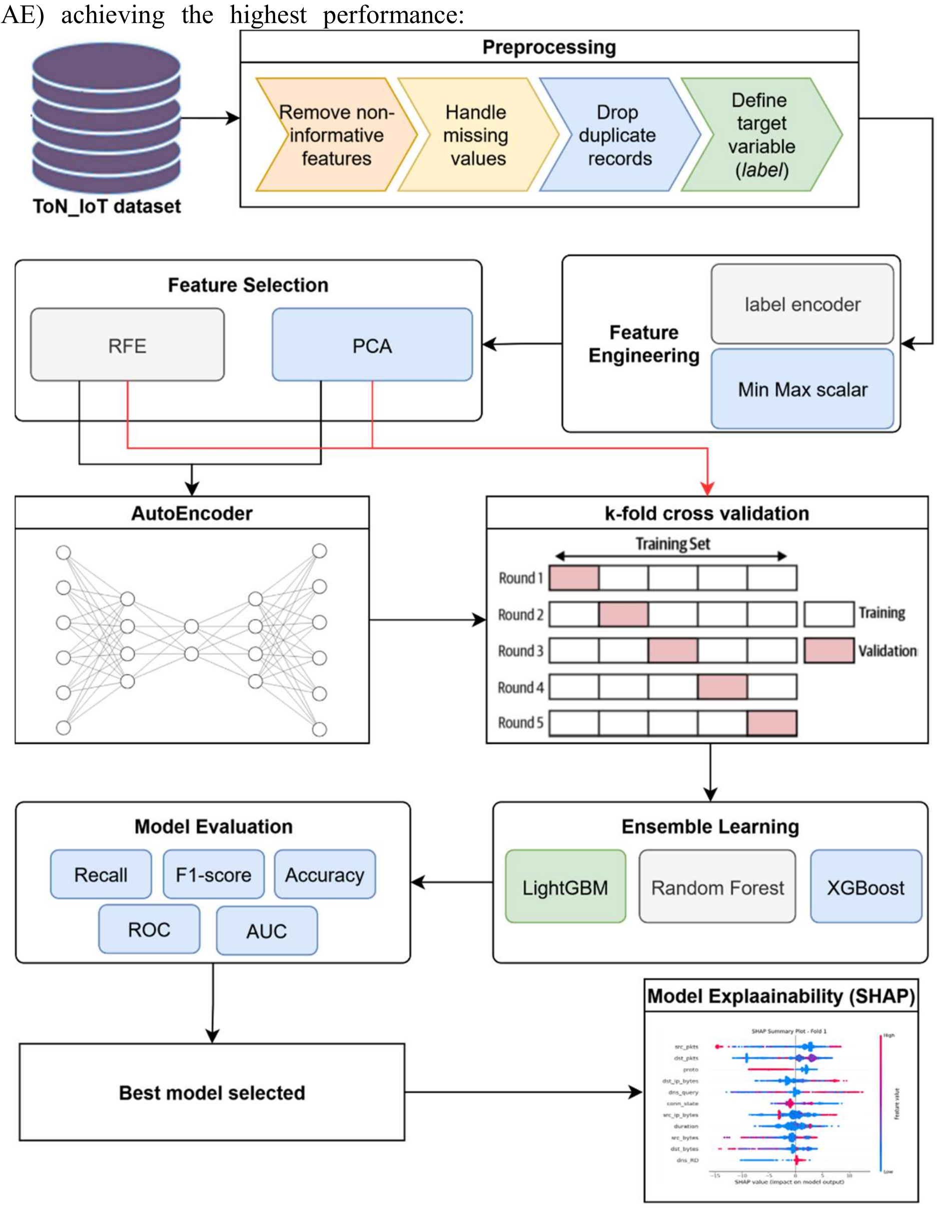

Figure. 1 Workflow diagram of the proposed IDS.

A. Dataset and Preprocessing

The experiments were conducted using the ToN-IoT dataset, a widely recognized benchmark in IoT security and intrusion detection research. This dataset contains heterogeneous IoT network traffic records with multiple categorical and numerical attributes. The dataset consists of 211,043 samples and 44 columns in total. Among these, 42 columns correspond to input features, while the remaining two are output labels: one used for binary classification (normal vs. attack) and the other representing the attack type for multiclass classification.

Preprocessing was conducted to ensure data consistency and quality before feeding it into the learning algorithms. First, four features related to network identifiers, namely src_ip, src_port, dst_ip, and dst_port, were removed. These attributes were excluded since they are primarily identifiers rather than discriminative features, and their presence may introduce bias or overfitting without contributing to generalizable classification performance.

Second, missing values were handled using mode (frequency) imputation, whereby the most frequent value of each feature was used to replace missing entries. This approach preserved the statistical distribution of the dataset while avoiding excessive loss of data. Third, duplicate records were eliminated, leaving only 93925 samples, as redundant samples could distort class distribution and inflate evaluation results. Finally, the target variable was defined as the attribute label, which encodes the binary classification task: distinguishing between normal and attack traffic.

B. Feature Engineering

The dataset contained a mixture of categorical and numerical features. Categorical attributes were encoded into numerical representations using the Label Encoder. This method assigns a unique integer to each category, thereby enabling ML models to process them as numerical variables. Unlike one-hot encoding, label encoding does not increase dimensionality, making it computationally efficient for datasets with many categorical values.

Numerical attributes, on the other hand, were normalized using the Min–Max Scaler. This transformation linearly scales features into the range [0,1], ensuring that attributes with larger magnitudes do not dominate the learning process. Feature scaling is particularly important for algorithms such as XGBoost, and LightGBM, which are sensitive to variations in feature ranges.

C. Feature Selection and Dimensionality Reduction

To enhance computational efficiency and mitigate the curse of dimensionality, both filtering and projection-based dimensionality reduction approaches were investigated. In particular, two methods were adopted:

- Recursive Feature Elimination (RFE): RFE is a wrapper-based feature selection method that recursively eliminates the least important features based on model weights until a predefined number of features remains. In this study, out of all 40 features in the dataset, two subsets of features were selected: one containing 11 features and another containing 9 features, in order to assess the trade-off between feature reduction and model performance.
- Principal Component Analysis (PCA): PCA is a linear transformation technique that projects correlated features into orthogonal principal components ranked by explained variance. Similar to RFE, two different configurations were tested, retaining the top 11 principal components and the top 9 principal components.

Both RFE and PCA were therefore applied under two settings each (9 and 11 features/components), and the resulting feature subsets were used independently in subsequent model training stages. This comparative design allowed the evaluation of how different dimensionality reduction strategies and feature subset sizes influence classification performance.

D. Representation Learning with Autoencoder

In addition to direct use of RFE- and PCA-based features, a deep autoencoder was implemented to capture non-linear feature interactions and generate compressed latent representations. The autoencoder is a feed-forward neural network consisting of two main components:

- Encoder: progressively reduces dimensionality by mapping the input features (generated by the feature selection phase) into a compressed latent space through successive hidden layers.
- Decoder: reconstructs the input features from the latent representation, forcing the encoder to learn compact yet informative embeddings.

The autoencoder architecture was symmetrical, with the encoder composed of layers of sizes 128, 64, and 32 neurons, and the decoder composed of layers of sizes 64, 128, and the original feature dimension. The ReLU activation function was applied in all hidden layers, while a linear activation was used in the output layer to allow reconstruction of continuous feature values. Training was performed using the Stochastic Gradient Descent (SGD) optimizer and the Mean Squared Error (MSE) as the reconstruction loss for 20 epochs, employing a validation split of 0.8.

Once trained, the encoder part of the autoencoder was extracted and applied to generate reduced-dimensionality representations for classification. The corresponding implementation is illustrated below (See Figure 2).

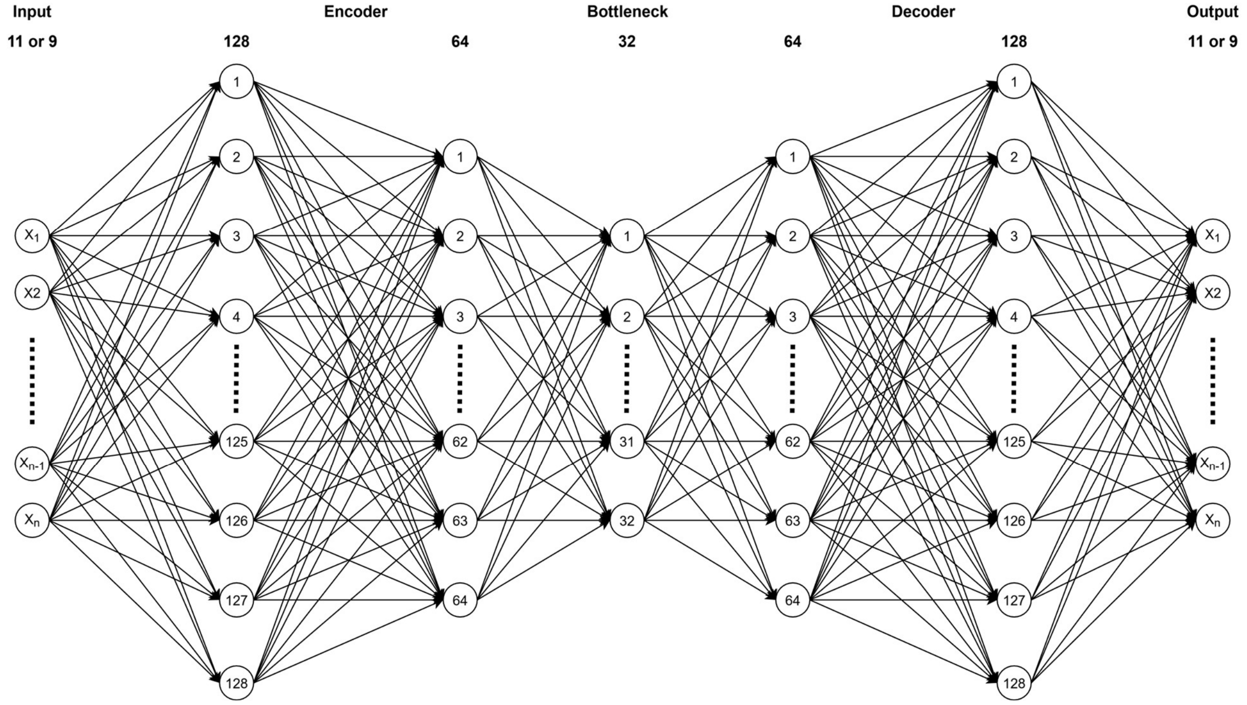

Figure. 2 Autoencoder model for feature reduction in IDS..

E. Model Training

The classification task was addressed using three widely adopted ensemble tree-based learning algorithms: XGBoost, LightGBM, and RF. These algorithms were selected due to their proven ability to handle heterogeneous data, capture complex feature interactions, and provide competitive performance in cybersecurity-related tasks.

To ensure robustness, k-fold cross-validation was employed, whereby the dataset was partitioned into 5 folds, with each fold serving once as the validation set while the remaining folds were used for training. This approach reduces overfitting and provides a more reliable estimate of model generalization.

Each classifier was trained under multiple hyperparameter configurations, including variations in tree depth, learning rate, number of estimators, and others. The configurations tested are summarized in Table 1.

Table 1. Hyperparameter configurations explored for LightGBM, XGBoost, and RF classifiers, including parameter ranges and their descriptions.

Classifier	Parameter	Values	Description
LightGBM	learning_rate	0.05, 0.1	Step size shrinkage for boosting
	max_depth	3, 5, 7, 9, 10, 11	Maximum tree depth
	n_estimators	100, 300, 400, 500, 600, 700	Number of boosting iterations
	num_leaves	15, 31, 63	Maximum leaves per tree
	Objective	binary	Binary classification task
	class_weight	balanced / None	Handle class imbalance
XGBoost	learning_rate	0.05, 0.1	Step size shrinkage
	max_depth	3, 5, 7, 9, 10, 11	Maximum tree depth
	n_estimators	100, 300, 400, 500, 600, 700	Number of boosting iterations
	Subsample	0.8, 1.0	Row sampling ratio
	colsample_bytree	0.8, 1.0	Feature sampling ratio
	min_child_weight	1, 5	Minimum sum of instance weight per child
	Gamma	0, 0.3	Minimum loss reduction for split
RF	min_samples_split	2, 5, 10	Minimum samples required to split a node
	max_depth	3, 5, 7, 9, 10, 11	Maximum tree depth
	n_estimators	100, 300, 400, 500, 600, 700	Number of trees in the forest
	min_samples_leaf	1, 2, 4	Minimum samples required at a leaf node

F. Model Evaluation and Explainability

The performance of the trained classifiers was assessed using three widely recognized metrics: accuracy, F1-score, and recall. Accuracy measures the overall correctness of predictions, recall quantifies the ability to identify attack samples, and the F1-score balances precision and recall, making it suitable for imbalanced datasets.

Additionally, the Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC) were computed for each fold, providing a graphical and quantitative assessment of the trade-off between true positive and false positive rates. The results were aggregated by averaging across folds to obtain a robust performance estimate.

Finally, to enhance interpretability, XAI was incorporated into the evaluation process using SHAP. In this study, only the SHAP summary plot was employed to provide a global view of feature importance, highlighting the overall contribution of each feature to the model's predictions. This step ensures that the proposed framework is not only accurate but also transparent and interpretable, which is a critical requirement in cybersecurity applications.

III. Results And Discussion

This section presents the results of a series of experiments conducted on the TON-IoT dataset for binary classification, distinguishing between normal and attack instances. All experiments employed k-fold (5-Fold) cross-validation with a fixed random state of 42 to ensure reproducibility. Three ensemble classification algorithms were evaluated (XGBoost, LightGBM, and RF) over two feature selection techniques (RFE, and PCA). The primary objective was to assess the performance of each algorithm and identify the most suitable model for deployment in the proposed system.

Initially, each algorithm was tested independently, and the results were analyzed to determine their classification accuracy. Hyperparameter tuning was subsequently performed to identify the optimal configurations that maximize performance. Finally, the highest accuracies obtained for each algorithm were compared to select the best-performing model for implementation in the proposed system.

A. XGBoost

Table 2 presents the hyperparameter search space considered during model optimization. The parameters varied included the number of estimators (100–700), learning rate (0.05 and 0.1), maximum tree depth (3–11), subsample ratio (0.8 and 1.0), column subsampling ratio (0.8 and 1.0), minimum child weight (1 and 5), and gamma (0 and 0.3). In total, these settings resulted in 1,344 unique configurations that were systematically evaluated. The best-performing parameter combination, as derived from this search, corresponded to the model that achieved the highest classification accuracy.

Table 2. Hyperparameter search space for XGBoost classifier, including all values evaluated.

Parameter	Values Tested
n_estimators	100, 200, 300, 400, 500, 600, 700
learning_rate	0.05, 0.1
max_depth	3, 5, 7, 9, 10, 11
subsample	0.8, 1.0
colsample_bytree	0.8, 1.0
min_child_weight	1, 5
gamma	0, 0.3

Table 3 summarizes the hyperparameter configurations and corresponding accuracies of the XGBoost model under different experimental settings, including scenarios with and without autoencoder-based feature reduction, and with features selected using either RFE or PCA (11 or 9 features).

The best-performing model used RFE without the autoencoder, achieving an accuracy of 99.53% with 11 features and the following hyperparameters: n_estimators = 500, learning_rate = 0.05, max_depth = 11, subsample = 0.8, colsample_bytree = 1.0, min_child_weight = 1, gamma = 0. Applying the autoencoder with 11 RFE-selected features yielded a slightly lower but comparable accuracy of 99.38%. Reducing the RFE feature set to 9 features led to a noticeable drop in performance, with accuracies of 95.95% (with autoencoder) and 96.46% (without autoencoder).

For PCA, results were more stable but slightly lower overall. Using 11 PCA components, the accuracies were 99.26% (with autoencoder) and 99.10% (without autoencoder). Reducing the PCA feature set to 9 components resulted in marginally lower accuracies of 99.27% (with autoencoder) and 99.06% (without autoencoder).

Table 3. The optimal values for each hyperparameter representing XGBoost classification model

Feature Selector	**RFE**				**PCA**			
AutoEncoder used?	**Yes**		**No**		**Yes**		**No**	
Number of features	**11**	**9**	**11**	**9**	**11**	**9**	**11**	**9**
n_estimators	600	600	500	300	700	500	600	400
learning_rate	0.1	0.1	0.05	0.05	0.1	0.1	0.1	0.1
max_depth	9	5	11	9	9	10	11	11
Subsample	1.0	0.8	0.8	1.0	1.0	0.8	0.8	0.8
colsample_bytree	0.8	1.0	1.0	0.8	1.0	0.8	1.0	1.0
min_child_weight	1	1	1	1	1	1	1	1
Gamma	0	0	0	0	0	0	0	0
Accuracy	99.38%	95.95%	**99.53%**	96.46%	99.26%	99.27%	99.10%	99.06%

Figure 3 illustrates the cross-validated ROC curve of the best-performing model. The curve shows an outstanding classification performance, with each fold achieving an AUC greater than 0.999. The mean ROC AUC is 0.9949, indicating excellent consistency across all validation folds. The ROC curve stays very close to the top-left corner of the plot, which reflects a very high true positive rate with an almost negligible false positive rate. This demonstrates that the model achieves near-perfect discrimination between normal and attack instances compared to the random chance line (red dashed line).

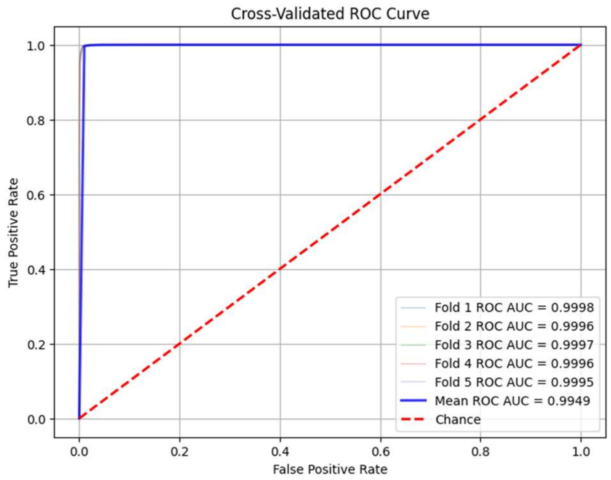

Figure. 3 ROC curve for the best-performing XGBoost model

To further evaluate classification performance, confusion matrices were generated for each of the five folds of the cross-validation experiment (See Figure 4). Across all folds, the classifier demonstrated strong discriminative capability, with very few misclassifications.

- True Positives (attack correctly identified) and True Negatives (normal traffic correctly identified) dominate each matrix, indicating that the model is highly reliable for both classes.
- The number of False Positives (normal traffic incorrectly classified as attack) per fold ranges between 46 and 66, while the number of False Negatives (attack traffic incorrectly classified as normal) is consistently low, between 32 and 37.
- Given that each fold contained approximately 18,000 samples, these misclassification counts represent a very small fraction of the dataset (<0.5%).

The consistency of the confusion matrices across all folds suggests that the classifier generalizes well and is not overfitting to specific partitions of the data. This indicates that the chosen hyperparameters for XGBoost yield a stable and robust model.

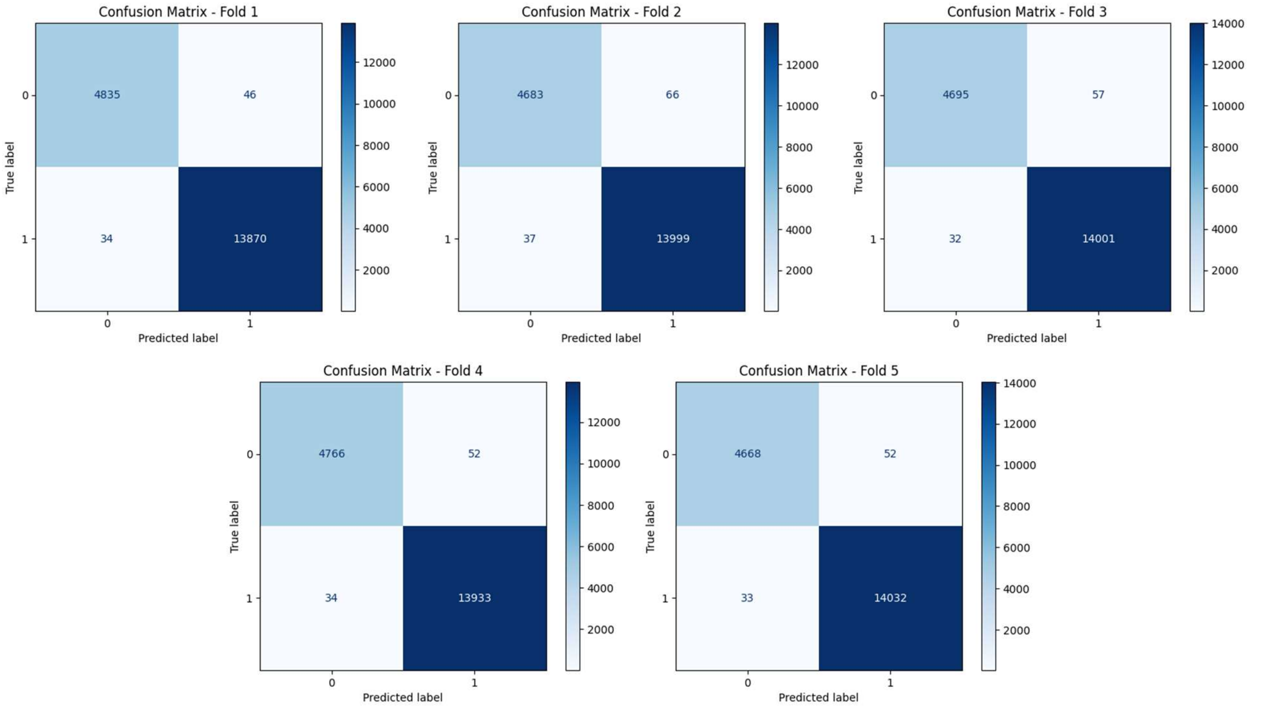

Figure. 4 Confusion Matrix curve for the best-performing XGBoost model.

B. LightGBM

Table 4 presents the hyperparameter search space explored during the optimization of the LightGBM model. The parameters varied included the number of estimators (100–700, in steps of 100), learning rate (0.05 and 0.1), maximum tree depth (3–11), and the number of leaves (15, 31, and 63). In total, these choices produced 252 unique parameter configurations, each systematically evaluated through cross-validation. Throughout the experiments, the class_weight parameter was fixed to "balanced" to compensate for class imbalance. The best-performing configuration was selected based on the highest classification accuracy achieved.

Table 4. Hyperparameter search space for LightGBM classifier, including all values evaluated.

Parameter	Values Tested
n_estimators	100, 200, 300, 400, 500, 600, 700
learning_rate	0.05, 0.1
max_depth	3, 5, 7, 9, 10, 11
num_leaves	15, 31, 63
class_wieght	Balanced

Table 5 presents the hyperparameter configurations and corresponding accuracies of the LightGBM model under different experimental settings, encompassing scenarios with and without autoencoder-based feature reduction, and with features selected using either RFE or PCA (11 or 9 features). The highest classification performance was achieved using RFE without the autoencoder, yielding an accuracy of 99.72% with 11 selected features. The associated hyperparameters for this optimal configuration were: n_estimators = 700, learning_rate = 0.1, max_depth = 11, num_leaves = 63. Incorporating the autoencoder with

11 RFE-selected features resulted in a slightly lower but comparable accuracy of 99.34%. Reducing the RFE feature set to 9 features produced accuracies of 99.14% (with autoencoder) and 99.68% (without autoencoder), indicating a minor decline in performance relative to the full feature set.

For PCA-based feature selection, the model exhibited slightly lower and more stable accuracies. Using 11 PCA components, the model achieved accuracies of 99.30% (with autoencoder) and 98.80% (without autoencoder). When the PCA feature set was reduced to 9 components, the accuracies decreased marginally to 99.28% (with autoencoder) and 98.71% (without autoencoder). Overall, these results indicate that RFE without autoencoder preprocessing provides the most effective feature representation for LightGBM in this experimental setup.

Table 5. The optimal values for each hyperparameter representing LightGBM classification model

Feature Selector	**RFE**				**PCA**			
AutoEncoder used?	**Yes**		**No**		**Yes**		**No**	
Number of features	**11**	**9**	**11**	**9**	**11**	**9**	**11**	**9**
n_estimators	700	700	700	600	700	700	500	300
learning_rate	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.05
max_depth	11	10	11	7	11	10	11	7
num_leaves	63	31	63	31	63	31	15	63
Accuracy	99.34%	99.14%	**99.72%**	99.68%	99.30%	99.28%	98.80%	98.71%

Figure 5 depicts the cross-validated ROC curve for the best-performing model. Each individual fold demonstrates near-perfect classification performance, with ROC AUC values ranging from 0.9998 to 0.9999, highlighting the model's robustness across folds. The mean ROC AUC of 0.9949 indicates excellent generalization capability and consistent predictive performance. The ROC curve closely follows the top-left corner of the plot, representing a very high true positive rate while maintaining an almost negligible false positive rate. In comparison to the chance line (red dashed), the model clearly exhibits superior discrimination between normal and attack instances, confirming its efficacy in correctly identifying both positive and negative classes.

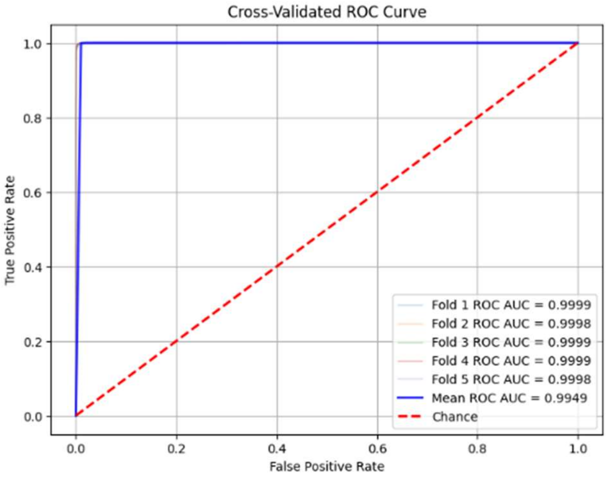

Figure. 5 ROC curve for the best-performing LightGBM model.

Figure 6 presents the confusion matrices obtained for each of the five folds in the cross-validation experiment. The results demonstrate that the classifier maintains consistently high performance across different data partitions.

- For the normal traffic class (True Negatives), the model consistently identified between 4,679 and 4,854 samples correctly in each fold.
- For the attack traffic class (True Positives), the classifier correctly predicted between 13,888 and 14,051 samples per fold, reflecting excellent reliability in detecting attacks.
- The number of False Positives (normal traffic misclassified as attack) remains very small, ranging between 27 and 41 across folds.
- Similarly, False Negatives (attacks misclassified as normal) are rare, with counts between 16 and 26.
- Considering that each fold contained approximately 18,000 test samples, these misclassifications represent less than 0.3% of the total predictions, underscoring the robustness of the model.

The dominance of correct predictions (both True Positives and True Negatives) across all folds, along with the consistently low error counts, indicates that the classifier generalizes well and is not overfitting to specific partitions of the data. This confirms that the chosen hyperparameters yield a stable and reliable model for distinguishing between normal and attack traffic.

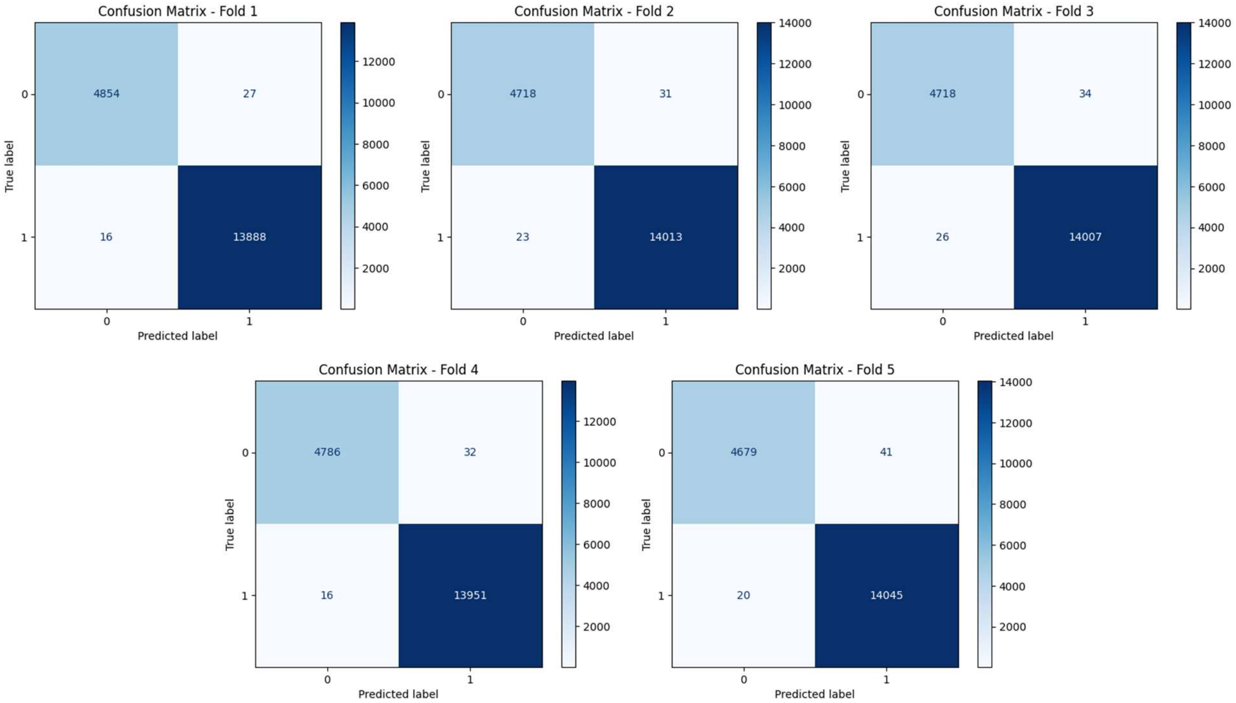

Figure. 6 Confusion Matrix curve for the best-performing LightGBM model.

C. Random Forest

Table 6 reports the hyperparameter search space considered during the optimization of the RF classifier. The search strategy incorporated variations in the number of estimators, maximum tree depth, and node-level splitting constraints, thereby covering a broad spectrum of model complexities. By evaluating this space through cross-validation, the study ensured that both underfitting-prone shallow trees and potentially overfitting deep trees were systematically assessed. This comprehensive approach provided a balanced exploration of the bias–variance trade-off, which is critical for achieving robust generalization.

The best-performing configurations, summarized in Table 7, reveal consistent patterns across experimental scenarios. In all cases, the classifier favored deeper decision trees, with the maximum depth converging at 11, min_samples_split fixed at 2, and min_samples_leaf at 1. These settings reflect the model's reliance on fine-grained partitions to capture the underlying data structure. While the number of estimators varied between 200 and 600 depending on feature selection, the classification performance remained consistently strong. Notably, RFE with 11 features emerged as the most effective representation, producing the highest accuracy of 99.20%, irrespective of the presence of an autoencoder. In contrast, PCA yielded slightly lower accuracies, ranging from 98.62% to 98.71%, indicating that PCA-based dimensionality reduction may discard some discriminative information relevant to intrusion detection. This finding suggests that RFE is more aligned with the feature distribution in this dataset, providing a richer and more task-specific representation.

The robustness of the optimized RF model is further corroborated by the ROC analysis. Figure 7 presents the cross-validated ROC curves, which consistently demonstrate near-perfect discrimination across all five folds. The ROC AUC values range between 0.9992 and 0.9995, with a mean ROC AUC of 0.9948, confirming excellent generalization ability. The ROC trajectory adheres closely to the upper-left boundary of the plot, reflecting a high true positive rate while maintaining an extremely low false positive rate. Compared to the chance line, the separation between normal and attack traffic is pronounced, highlighting the model's reliability in real-world detection scenarios.

The confusion matrices (Figure 8) provide additional insights into classification reliability. Across the five folds, the number of correctly classified normal samples (True Negatives) ranged from 4,592 to 4,759, while attack instances (True Positives) ranged from 13,874 to 14,039. Misclassification rates were minimal, with False Positives (normal traffic misclassified as attack) ranging from 103 to 157 and False Negatives (attacks misclassified as normal) ranging from 26 to 34. Considering that each fold contained approximately 18,000 test samples, these errors represent less than 0.5% of total predictions, underscoring the stability and robustness of the model.

Overall, the results demonstrate that the RF classifier, when optimized with appropriate hyperparameters and RFE-based feature selection, achieves highly reliable performance in distinguishing between normal and attack traffic. The consistency of results across folds indicates that the model generalizes well to unseen data and is not sensitive to data partitioning. Furthermore, the superior performance of RFE relative to PCA suggests that task-driven feature selection techniques may be more effective than unsupervised dimensionality reduction methods in intrusion detection contexts. These findings confirm that RF, when appropriately tuned, can serve as a strong and interpretable baseline for intrusion detection tasks.

Table 6. Hyperparameter search space for RF classifier, including all values evaluated

Parameter	Values Tested
n_estimators	100, 200, 300, 400, 500, 600, 700

max_depth	3, 5, 7, 9, 10, 11
min_samples_split	2, 5, 10
min_samples_leaf	1, 2, 4

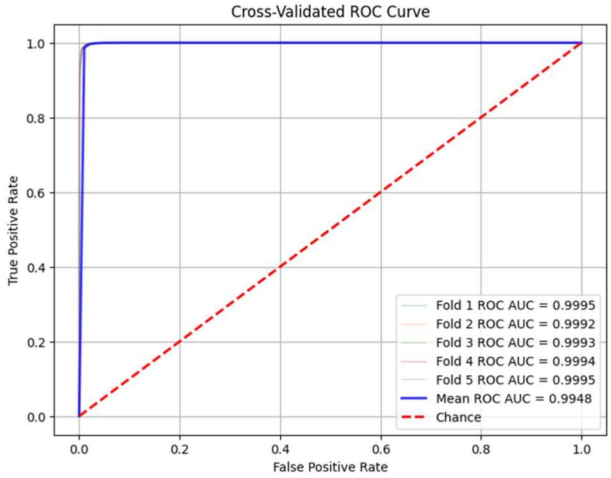

Figure. 7 ROC curve for the best-performing RF model.

Table 7. The optimal values for each hyperparameter representing RF classification model

Feature Selector	**RFE**				**PCA**			
AutoEncoder used?	**Yes**		**No**		**Yes**		**No**	
Number of features	11	9	11	9	11	9	11	9
n_estimators	400	200	600	600	500	400	300	200
max_depth	11	11	11	11	11	11	11	11
min_samples_split	2	2	2	2	2	2	2	2
min_samples_leaf	1	1	1	1	1	1	1	1
Accuracy	99.20%	99.06%	**99.20%**	99.19%	98.66%	98.62%	98.71%	98.71%

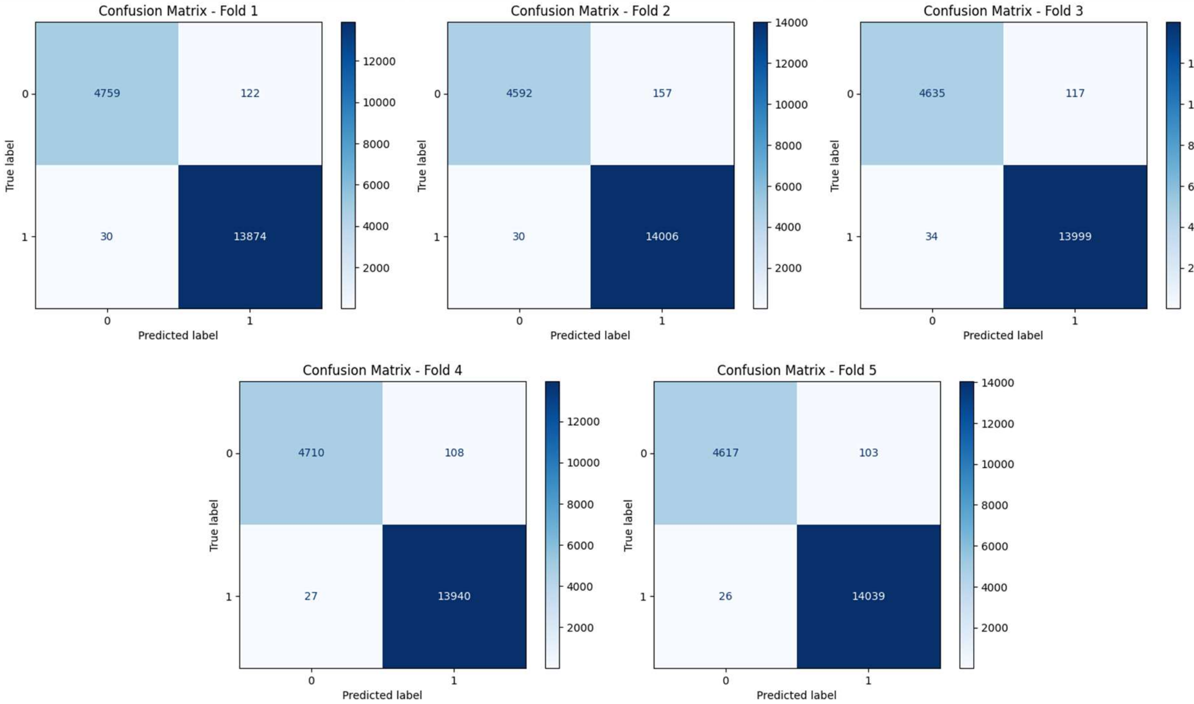

Figure. 8 Confusion Matrix curve for the best-performing RF model.

D. Best model selection

The comparative performance of XGBoost, LightGBM, and RF classifiers is summarized in Table 8. Overall, all three algorithms demonstrated excellent classification ability, achieving accuracy scores above 99%. LightGBM obtained the highest accuracy (99.72%), followed closely by XGBoost (99.53%) and RF (99.20%).

In terms of F1-score, LightGBM (99.81%) marginally outperformed XGBoost (99.68%) and RF (99.46%). A similar trend was observed for precision, where LightGBM achieved the highest score (99.76%), compared to XGBoost (99.61%) and RF (99.14%). Notably, all models achieved very high recall, with XGBoost and RF exceeding 99.7%, while LightGBM reached the maximum (99.86%).

Regarding computational efficiency, XGBoost exhibited the fastest training time (2.81 s), followed by LightGBM (4.45 s), whereas RF required substantially more time (34.95 s). For prediction time, XGBoost was also the most efficient (0.23 s), while LightGBM (0.89 s) and RF (1.00 s) incurred longer prediction latencies.

The highest classification performance was achieved by LightGBM, with an accuracy of 99.72% using the RFE-selected feature set. Notably, the model demonstrated very low variability across multiple runs, with a standard deviation of only ±0.04%. Such a small variation indicates that the model's performance is highly stable, reflecting a strong ability to generalize beyond the training data rather than overfitting to specific samples. Overall, these results highlight that the classifier is both highly accurate and reliably consistent.

Table 8. Summary of the best predictive results achieved by the three classifiers.

Algorithm	Accuracy (std)	F1-score	Precision	Recall	Average Training time	Average prediction time
XGboost	99.53% (± 0.04%)	99.68%	99.61%	99.76%	**2.8069**	**0.228033**
LightGBM	**99.72%** (± 0.04%)	**99.81%**	**99.76%**	**99.86%**	4.4455	0.891519
RF	99.20% (± 0.06%)	99.46%	99.14%	99.79%	34.9529	1.000112

E. Model explanability

The interpretability analysis was conducted using SHAP values derived from the LightGBM model trained on the TON_IoT dataset. Since this is a binary classification task, we focus on shap_values[1], which corresponds to the contribution of each feature toward predicting the positive class (i.e., attack traffic). To ensure that only the most relevant predictors were considered, RFE was applied prior to model training. The SHAP summary plots across five cross-validation folds are presented in Figure 9, where each subplot corresponds to one fold.

Across all folds, traffic volume–related features consistently emerged as the most influential predictors. In particular, src_pkts and dst_pkts demonstrated the largest spread of SHAP values, indicating their strong discriminative power. High values of these features (shown in red) were associated with positive SHAP contributions, thereby increasing the likelihood of predicting attack traffic. Conversely, lower values (blue) pushed predictions toward the benign class. This finding highlights the centrality of packet-level statistics in distinguishing IoT-related attacks, particularly those involving volumetric anomalies such as flooding and scanning.

Beyond packet counts, protocol-specific features (proto and conn_state) also contributed significantly across folds. Their influence suggests that the structural characteristics of traffic flows, including protocol type and connection state transitions, are critical indicators of malicious behavior. These features likely capture the exploitation of specific protocols or abnormal session states that are common in IoT intrusions.

Byte-level attributes (dst_ip_bytes, src_ip_bytes, and src_bytes) were ranked consistently in the mid-range of feature importance. While their impact was less pronounced than packet counts, their consistent contribution indicates that payload size distributions and directional traffic flow remain important signals for classification.

In contrast, DNS-related features (dns_query and dns_RD) and flow duration exhibited relatively lower contributions. Although they provide supplementary cues in certain scenarios, their limited influence suggests that DNS behavior is less indicative of attack activity in the TON_IoT dataset compared to traditional network intrusion datasets. In particular, dns_RD (recursion desired flag) ranked consistently at the bottom across all folds, confirming its negligible role in the LightGBM decision process.

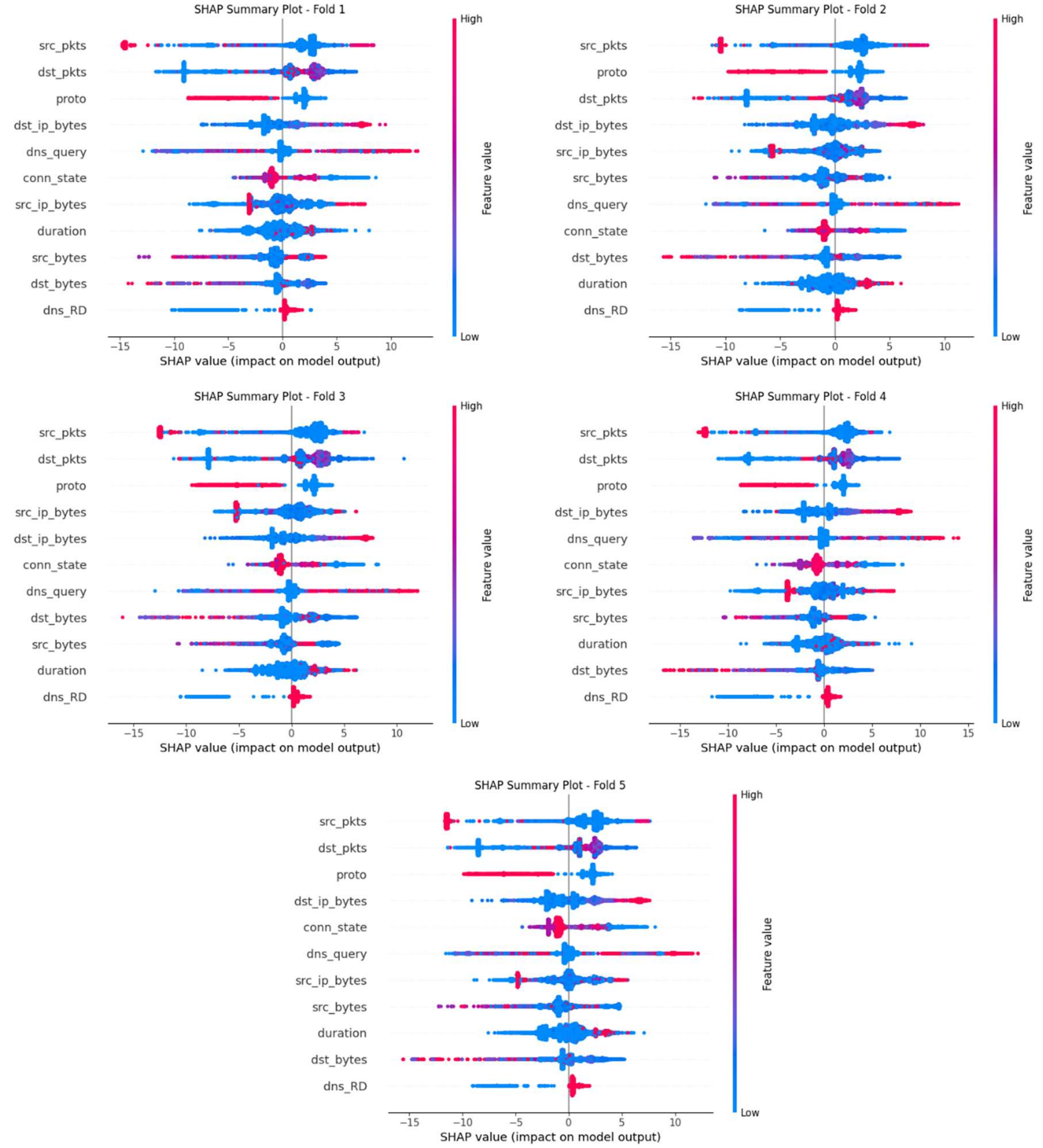

Figure. 9 SHAP summary plots for five cross-validation folds.

From a broader perspective, these findings underscore the complementary role of behavioral and structural features in IoT intrusion detection. While packet and byte-level statistics provide strong and direct indicators of malicious behavior, protocol-specific characteristics enhance classification in more ambiguous cases. Importantly, the consistency of feature rankings across five cross-validation folds demonstrates the stability and robustness of the LightGBM model, increasing confidence in its generalization to unseen traffic.

This interpretability analysis has two practical implications. First, it provides evidence-based justification for prioritizing volume-based and protocol-specific features in IoT IDSs, which can improve efficiency by focusing on the most impactful signals. Second, it shows that some attributes, such as DNS flags, may be deprioritized in future feature engineering without significantly compromising performance. Overall, the integration of RFE with SHAP analysis not only enhances model interpretability but also offers actionable insights for the development of lightweight and effective intrusion detection solutions in IoT environments.

F. Discussion

A comparison of the two feature selection techniques highlights key differences. RFE tends to achieve higher peak performance when more features are retained (11 features), but its accuracy drops more sharply when reduced to 9 features, suggesting sensitivity to feature removal. In contrast, PCA provides more consistent performance across different feature counts, though its maximum accuracy remains slightly below RFE's best results. This indicates that RFE can better preserve highly discriminative features, while PCA, being a transformation-based approach, offers robustness at the cost of a small loss in peak accuracy.

The differences in performance can be attributed to feature dimensionality and information content, and the nature of the feature selection method. RFE directly selects the most informative attributes, whereas PCA transforms the features into principal components, which may reduce redundancy but can also obscure feature interpretability. Using 11 features preserves more relevant information for classification, allowing the model to better distinguish between normal and attack instances. Applying the autoencoder introduces a feature transformation step, which can slightly reduce redundancy but may also lead to minimal information loss, explaining the small decrease in accuracy. Reducing the number of features to 9 likely removed some informative attributes, leading to a more pronounced drop in performance. Hyperparameter choices, such as the number of estimators, maximum depth, and subsampling ratios, further influence how well the model adapts to each feature space.

The comparative evaluation of the three ensemble classifiers demonstrates that, while all achieved excellent predictive performance, LightGBM consistently outperformed the alternatives across most metrics. Its superior accuracy, F1-score, and recall can be attributed to its leaf-wise tree growth strategy, which splits the leaf with the highest loss reduction. This approach generates deeper, more specialized trees capable of capturing complex decision boundaries, thereby improving overall predictive ability. In contrast, XGBoost employs a level-wise tree growth strategy, which produces more balanced trees but may sacrifice some granularity, explaining its slightly lower predictive performance compared to LightGBM. RF, despite achieving strong results, lacks the sequential error-correction mechanism inherent in boosting methods. Since its trees are constructed independently through bootstrap aggregation, the algorithm does not iteratively refine misclassified samples, leading to comparatively lower accuracy and precision.

The observed differences in computational efficiency also align with the design principles of these algorithms. XGBoost exhibited the fastest training and prediction times, largely due to its efficient parallelization, histogram-based gradient boosting, and level-wise construction, which limit tree depth and reduce computational cost. LightGBM, although generally considered efficient, was moderately slower in this study. Its leaf-wise growth strategy, while advantageous for accuracy, often produces deeper trees with more splits, which increases training and inference times on smaller datasets. By contrast, RF was the slowest algorithm, as training requires building a large number of fully grown trees independently, and predictions involve evaluating every tree in the ensemble. This brute-force approach explains its significant time overhead compared to boosting methods.

Overall, the results suggest that LightGBM is the most suitable choice when predictive accuracy is the primary objective, particularly in contexts where minimizing false positives and false negatives is critical. However, XGBoost offers the best trade-off between performance and efficiency, making it a more practical option for time-sensitive or resource-constrained applications. RF, though competitive in predictive capability, appears less optimal for large-scale or real-time tasks due to its high computational demands.

As shown in Table 9, among all compared approaches, our proposed system demonstrated superior performance, with the RFE–LightGBM model achieving the highest accuracy (99.72%) and most balanced evaluation metrics (precision 99.81%, recall 99.76%, and F1-score 99.86%). Compared to earlier methods, which either relied on deep learning without feature selection or on FE techniques with limited discriminative capability, our system effectively combines RFE with advanced ensemble classifiers. This integration ensures that only the most relevant features are retained, thereby reducing redundancy and improving the classifier's ability to capture complex attack patterns. As a result, the proposed models not only outperform existing IDS solutions but also deliver greater consistency across all metrics, highlighting their robustness for real-world intrusion detection.

In addition to achieving superior predictive performance, the proposed system incorporates XAI through SHAP summary plots. While previous works primarily reported classification results without addressing model interpretability, our approach provides transparent insights into how individual features contribute to the final predictions. This not only enhances trust in the system but also supports cybersecurity analysts in understanding the rationale behind detection outcomes. The integration of SHAP distinguishes our work from existing IDS studies by combining state-of-the-art accuracy with interpretability, thereby addressing one of the major limitations of earlier IDS solutions that often functioned as "black boxes." Consequently, the proposed framework not only outperforms prior models in terms of accuracy, precision, recall, and F1-score, but also delivers actionable explanations that improve its applicability in real-world intrusion detection scenarios.

Table 9. Comparison of Our Proposed System with Related IDS Models.

Ref	FS Algorithm(s)	Classification algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
[12]	All of these together (MI, PCC, K-Best)	Stacking (meta-LR)	98.63	98.20	98.60	98.61
[13]	PCC	Hybrid CNN–LSTM	98.75	98.70	98.72	98.71
[14]	PCA	KNN	89.10	87.78	89.28	88.39
[15]	None	LSTM	96.35	98.40	96.00	97.35
[16]	None	1D-CNN	99.24	98.00	98.0	98.00
[17]	Autoencoder	RF	88.66	88.00	88.00	88.00
Our proposed system	RFE	LightGBM	**99.72**	**99.81**	**99.76**	**99.86**
		XGboost	99.53	99.68	99.61	99.76
		RF	99.20	99.46	99.14	99.79

IV. CONCLUSION

This study investigated the development of an efficient and interpretable Intrusion Detection System (IDS) using the ToN_IoT dataset for binary classification of network traffic. The main research questions focused on identifying (1) which feature selection technique (PCA or RFE) yields the best performance, and (2) which ensemble learning model among LightGBM, XGBoost, and Random Forest provides the highest classification accuracy and computational efficiency. The results clearly demonstrate that Recursive Feature Elimination (RFE) outperformed Principal Component Analysis (PCA), providing the best results when selecting 11 features, which led to improved detection accuracy and reduced computational overhead. Among the ensemble classifiers evaluated, LightGBM achieved the highest accuracy of 99.72%, with the fastest training and testing times. The selected model was further interpreted using the SHAP summary plot, which revealed the most influential features contributing to the classification decisions. This not only added transparency to the detection process but also enabled a deeper understanding of the underlying data patterns. The main contributions of this work include a comparative evaluation of PCA and RFE for feature selection in IDS, a benchmark comparison of three powerful ensemble learning algorithms, as well as integration of SHAP XAI for global interpretability, and a complete, fast, and accurate IDS pipeline suitable for IoT environments. Despite the strong results, one limitation of the study is that it focused solely on binary classification. Future research should explore multiclass classification to differentiate between specific attack types, and investigate the use of online or incremental learning for real-time deployment. Additionally, testing the model in a live IoT environment would help validate its robustness and generalizability. This work contributes to the growing body of intelligent cybersecurity solutions by offering a practical, high-performing, and interpretable IDS framework that can be deployed in smart industrial and IoT infrastructures.

REFERENCES

[1] "Zscaler ThreatLabz 2023 Enterprise IoT and OT Threat Report | Zscaler." Accessed: Oct. 31, 2025. [Online]. Available: https://info.zscaler.com/resources-industry-reports-threatlabz-2023-enterprise-iot-ot-threat-report?utm_source=chatgpt.com

[2] "SonicWall 2024 Mid-Year Cyber Threat Report: IoT Madness, PowerShell Problems and More." Accessed: Oct. 31, 2025. [Online]. Available: https://www.sonicwall.com/blog/sonicwall-2024-mid-year-cyber-threat-report-iot-madness-powershell-problems-and-more?utm_source=chatgpt.com

[3] M. Nitti *et al.*, "Citation: Trustworthy Adaptive AI for Real-Time Intrusion Detection in Industrial IoT Security," *IoT*, 2025, doi: 10.3390/iot6030053.

[4] N. Sharma and B. Arora, "Machine Learning and Deep Learning Models for Anomaly Intrusion Detection in Networks: A Systematic Review," *SN Computer Science 2025 6:7*, vol. 6, no. 7, pp. 1–38, Sep. 2025, doi: 10.1007/S42979-025-04352-Z.

[5] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and Adna N Anwar, "TON-IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," *IEEE Access*, vol. 8, pp. 165130–165150, 2020, doi: 10.1109/ACCESS.2020.3022862.

[6] H. Dhirar and A. Hamad, "Comparative evaluation of a novel IDS dataset for SDN-IoT using deep learning models against InSDN, BoT-IoT, and ToN-IoT," *Measurement: Digitalization*, vol. 4, p. 100015, Dec. 2025, doi: 10.1016/J.MEADIG.2025.100015.

[7] S. Rajarajeswari, M. Grover, L. Yashoda, P. Mathurkar, D. Bhanu, and M. Singh, "An Effective Design of Intrusion Detection System With Classification Algorithms And Feature Reduction In Machine Learning," *2nd IEEE International Conference on Innovations in High-Speed Communication and Signal Processing, IHCSP 2024*, 2024, doi: 10.1109/IHCSP63227.2024.10960043.

[8] M. Prasad, S. Tripathi, and K. Dahal, "A Feature Probability Estimation-based Feature Selection Approach for Intrusion Detection," *2025 6th International Conference on Recent Advances in Information Technology (RAIT)*, pp. 1–6, Jul. 2025, doi: 10.1109/RAIT65068.2025.11089293.

[9] M. Rajkumar, L. Vs, R. Karthik, and S. Pavithra, "Optimized Deep Learning Mechanism for Intrusion Detection: Leveraging RFE-Based Feature Selection and PCA for Improved Accuracy," *5th International Conference on Sustainable Communication Networks and Application, ICSCNA 2024 - Proceedings*, pp. 1517–1522, 2024, doi: 10.1109/ICSCNA63714.2024.10863936.

[10] K. Wu, Y. Li, J. Sun, Q. Qin, and J. Li, "An ensemble framework with improved grey wolf optimization algorithm and multi-level feature selection for IoT intrusion detection," *Cluster Computing 2025 28:12*, vol. 28, no. 12, pp. 1–34, Sep. 2025, doi: 10.1007/S10586-025-05374-1.

[11] M. S. Farooq *et al.*, "Interpretable Federated Learning Model for Cyber Intrusion Detection in Smart Cities with Privacy-Preserving Feature Selection," *Computers, Materials & Continua*, vol. 0, no. 0, pp. 1–10, 2025, doi: 10.32604/CMC.2025.069641.

[12] Y. Alotaibi and M. Ilyas, "Ensemble-Learning Framework for Intrusion Detection to Enhance Internet of Things' Devices Security," *Sensors*, vol. 23, no. 12, Jun. 2023, doi: 10.3390/s23125568.

[13] S. Yaras and M. Dener, "IoT-Based Intrusion Detection System Using New Hybrid Deep Learning Algorithm," *Electronics (Basel)*, 2024, doi: 10.3390/electronics.

[14] J. Li, M. S. Othman, H. Chen, and L. M. Yusuf, "Optimizing IoT intrusion detection system: feature selection versus feature extraction in machine learning," *J Big Data*, vol. 11, no. 1, Dec. 2024, doi: 10.1186/s40537-024-00892-y.

[15] R. A. Elsayed, R. A. Hamada, M. I. Abdalla, and S. A. Elsaid, “Securing IoT and SDN systems using deep-learning based automatic intrusion detection,” *Ain Shams Engineering Journal*, vol. 14, no. 10, Oct. 2023, doi: 10.1016/j.asej.2023.102211.

[16] A. Alabbadi and F. Bajaber, “An Intrusion Detection System over the IoT Data Streams Using eXplainable Artificial Intelligence (XAI),” *Sensors*, vol. 25, no. 3, Feb. 2025, doi: 10.3390/s25030847.

[17] J. Li, H. Chen, M. O. Shahizan, and L. M. Yusuf, “Enhancing IoT security: A comparative study of feature reduction techniques for intrusion detection system,” *Intelligent Systems with Applications*, vol. 23, Sep. 2024, doi: 10.1016/j.iswa.2024.200407.