Ensemble-Based Network Anomaly Detection Using RFE and Information Gain for Optimized Feature Selection

Nagamani Uddamari[1*], P. Sammulal[2]
[1]Ph.D. Scholar CSE, JNTUH, Hyderabad, India
[2]Professor of CSE, JNTUH college of Engineering, Hyderabad, India
E-mail: [1]nagamani.u@gmail.com [2] Sam@jntuh.ac.in
[*]Corresponding author

Keywords: intrusion detection, ensemble learning, feature selection, cybersecurity, machine learning

Received: February 24, 2025

Intrusion Detection Systems (IDSs) play a significant role in reducing dynamic cyber threats. However, current machine learning-centric IDSs are not without issues, as they may have a high false positive rate and suboptimal feature selection, resulting in a low detection rate. This paper proposes an ensemble IDS architecture that utilizes RFE and IG for feature selection, aiming to enhance anomaly detection performance and reduce computational intensity. We begin with a preprocessing pipeline that includes data cleaning, one-hot encoding of categorical features, and normalization to scale the features. The most discriminative attributes are selected to minimize redundancy. Then, the selected feature subset is fed to build a set of ensemble classifiers, including Random Forest, XGBoost, Extra Trees, and a weighted Voting Classifier. Extensive experimental results on the CIC-IDS2017 datasets demonstrate that the proposed ensemble-level approach outperforms in all aspects, achieving 97.5% accuracy, 97.2% precision, 97.8% recall, and 97.5% F1-score. Overall, the ensemble model exhibits an improvement in terms of recall and hence robustness compared to the two baseline classifiers, namely the standalone Random Forest (recall: 96.5%) and XGBoost (recall: 97.3%). We also conducted an ablation study that confirms the effectiveness of RFE and Information Gain by comparing settings with and without feature selection. These findings indicate that the proposed IDS architecture can be feasibly and scalably implemented for real-time network anomaly detection. Adaptive feature selection and deployment in a streaming setting could be investigated to enhance its resistance to novel attacks in the future.

Povzetek: Sistem uvaja hibridni pristop za odkrivanje omrežnih napadov. S kombinacijo RFE in Information Gain izbere ključne značilke, nato jih združi z ansambelskim učenjem (Random Forest, XGBoost, Voting). Rezultat je večja točnost in manj lažnih alarmov; metoda izboljša odzivnost IDS in omogoča skalabilno zaznavanje napadov v realnem času.

1 Introduction

In modern cybersecurity, intrusion detection systems (IDS) are key to monitoring network traffic and detecting malicious actions. Signature-based and anomaly-based detection techniques in traditional intrusion detection systems (IDS) cannot cope with the rapidly evolving cyber threats. Due to constant updates, signal-based approaches are traditionally ineffective against zero-day attacks, while anomaly-based solutions experience high false positive rates. Machine learning (ML)-based methods have overcome some of these limitations by incorporating automatic feature selection and anomaly classification capabilities into Intrusion Detection Systems (IDSs) [6, 7]. Ensemble learning techniques, such as Bagging, Boosting, and Random Forest, have further enhanced classification performance [8]. Nonetheless, optimizing feature selection, enhancing classification accuracy, and adapting to new attack patterns are still challenging.

Several previous studies have explored feature selection and ensemble learning to improve.

Intrusion detection performance. For example, Chohra et al. [1] employed PSO as a search function (SF) technique to enhance the accuracy of detection. However, it is not the exact grid search as ours because it is non-deterministic and non-interpretable. In contrast, our criterion is a deterministic and interpretable combination of RFE (Recursive Feature Elimination) and Information Gain, which addresses redundancy and improves generalization. Our proposed approach is designed to be used with ensemble tree-based classifiers, which provide high accuracy and are reasonably interpretable. Abbas et al. [2] investigated the best model between ensemble-based and single classifiers, demonstrating that ensemble-based intrusion detection models could achieve a better detection rate. Nevertheless, it has been found that these approaches have several drawbacks, including suboptimal feature selection, a lack of multiple classifiers, and ineffectiveness in detecting imbalanced attack classes. Additionally, most

approaches are unable to adapt to the dynamic nature of cyber threats. Therefore, there is a need for an automated IDS framework that offers high adaptability, accuracy, scalability, and sustainable performance. To address these gaps, this study presents an optimized hybrid intrusion detection framework that leverages Recursive Feature Elimination (RFE) and Information Gain in conjunction with ensemble learning models to enhance classification accuracy, precision, recall, and F1-score. Compared to previous methods, our proposed model can automatically eliminate irrelevant features, reduce computational overhead, and improve detection performance. Moreover, it increases the classification rate of both seen and unseen attacks using XGBoost, Random Forest, and an Ensemble Voting Classifier, which yields better results than classical one-class classifier approaches.

In this study, the novelty lies in the utilization of a combination of RFE and Information Gain for feature selection, resulting in a detailed and refined dataset for classification. This algorithm combines the outputs of multiple models, thus increasing overall detection accuracy, minimizing false alerts, and improving model stability and reliability. In addition, this study conducts a comprehensive ablation analysis to demonstrate the impact of feature selection on model performance and to gain a deeper understanding of how feature selection enhances IDS efficiency.

To set the research direction, this study aims to evaluate whether ensemble classifiers based on RFE, IG, or SFFS, or IG Feature Score as feature selection techniques can significantly enhance IDS classifiers compared to single classifiers. The central hypothesis is that by utilizing sophisticated feature selection techniques in conjunction with ensemble learning, our system can achieve improved accuracy, reduced false positives, and increased generality in detecting new cyber threats. The specific contributions are as follows: for the first time, we construct a stable data pipeline, propose a hybrid feature selection method that retains high-dimensional features with discriminative value, and compare multiple ensemble classifiers (i.e., Random Forest, XGBoost, Extra Trees, and Voting Classifier) in the optimized feature space. We conduct thorough experimentation to justify further the proposed framework, including comparisons with state-of-the-art models, ablation analyses, and investigations into interpretability. This organized research framework ensures that the solution not only fills the gap in feature selection and detection accuracy but also provides a scalable and deployable model for modern network security systems.

Our primary research contributions are as follows: opting for an optimized IDS framework based on ensemble learning, integrating feature selection techniques to enhance model performance, and conducting an extensive analysis of the model against state-of-the-art approaches. A comprehensive performance comparison is also included in this study, which validates the real-world applicability of the proposed model for cybersecurity. The paper is organized as follows: Section 2 presents an extensive literature survey on existing intrusion detection systems (IDS) techniques, their weaknesses, and recent trends in feature selection and ensemble techniques. The proposed methodology is described in Section 3, which includes feature selection strategies and the ensemble-based classification approach. Section 4 presents experimental results using our model, including ablation studies and comparisons with the state-of-the-art. In Section 5, we present the study's findings, implications, and contributions, as well as its limitations. The final section, Section 6, concludes the work by presenting key findings and recommendations to enhance the effectiveness of IDS.

2 Related work

Recent research on intrusion detection systems explores various machine learning techniques, focusing on feature selection and ensemble learning to enhance network anomaly detection. Chohra et al. [1] developed a feature selection approach based on swarm intelligence, which demonstrated good performance across various datasets. However, the scalability still has to be improved. Velasquez et al. [2] developed a hybrid ensemble model that performed well but required further testing in various settings for real-time anomaly detection in industrial systems. Khan and Sayyid et al. [3] investigate different machine learning models for predicting thyroid disorders in diabetic patients. Random Forest and SVM showed high predictive performance. The proposed approach emphasizes the importance of disease-specific cause selection and dataset balancing in improving the performance of machine learning for accurate clinical decision-making. Aliyeva et al. [4] investigated the use of XGBoost to increase efficiency by automating cybercrime detection; however, difficulties arise with human procedures and algorithmic constraints. Abbas et al. [5] proposed an ensemble intrusion detection model that combines decision trees, naive Bayes, and logistic regression. It demonstrated increased accuracy, but further improvements are needed to keep pace with the ever-evolving nature of cyber threats.

Hossain and Islam [6] presented an ensemble-based intrusion detection model with good accuracy utilizing various methods; nevertheless, further work is required to improve generalizability across different datasets. Thockchom et al. [7] improved performance by developing an ensemble learning model for intrusion detection that utilizes lightweight classifiers; however, difficulties with misclassification and dataset diversity should be addressed in future research. Eddine et al. [8] developed a feature-engineered machine learning-based intrusion detection model for the Industrial Internet of Things (IIoT), achieving high accuracy. However, future research should focus on possible security flaws. Golchha et al. [9] proposed a framework for voting-based ensemble learning that utilizes CatBoost, HGB, and RF to detect IIoT cyberattacks with high accuracy. Future research should focus on more general attacks and practical uses.

Adeshina et al. [10] compare linear and logistic regression models for disease prediction. The process consists of three steps: preprocessing, a test for multicollinearity, and statistical validation. The results suggest that logistic regression is superior to other methods in predicting trends or glitches in binary classification problems, due to its proper handling of categorical disease outcomes.

Hooshmand et al. [11] proposed a hybrid sampling approach (SKM) to detect network anomalies, thereby increasing the detection rates of the minority class. Subsequent research should delve deeper into optimization and assess its use with a broader range of datasets. Ahmed et al. [12] developed the Deep Ensemble Learning Model (DELM) to detect abnormalities in network data and utilize Adaptive Feature Aggregation to enhance flexibility. Enhancing real-time detection capabilities and testing against various attack types should be the main priorities of future development. Lai et al. [13] developed an ensemble learning system to detect anomalies in IoT cybersecurity; however, it struggles with diverse data. More comprehensive applications could be investigated in future research. Allafi et al. [14] created the ensemble learning-based AOAEL-CDC approach for IoT cybersecurity; however, it has issues with traditional IDS. Subsequent research might improve methods of detection. Lin et al. [15] encountered difficulties with offline modifications and created an ensemble machine learning network intrusion detection system (ML NIDS) based on a hypergraph for real-time port scan detection. More research might enhance adaptability even more.

Kunhare et al. [16] increased accuracy and decreased false alarms using a random forest approach for feature selection in IDS. Other optimization strategies may be explored in future research. Almasoudy et al. [17] proposed a feature selection approach for IDS based on Differential Evolution, which improved detection rates but encountered complexity issues. Accuracy may be further refined in future studies. Li et al. [18] presented AE-IDS, a deep learning technique that shortens training times while improving intrusion detection accuracy. Reliance on labeled datasets is one of the limitations; this may be addressed in further study. Taoussi et al. [19] describe a hybrid model for depression detection, which utilizes SMOTE to address class imbalance, RoBERTa for extracting deep contextual embeddings, and CNN-LSTM for temporal classification. The proposed end-to-end model achieves higher accuracy and robustness compared to the state-of-the-art models, especially on the imbalanced mental health dataset. Prasad et al. [20] developed a hybrid feature selection strategy to enhance intrusion detection using Bayes' theorem and Rough Set theory. Potential biases in the dataset are among the limitations; more research may improve scalability.

Stiawan et al. [21] developed an enhanced ensemble IDS incorporating six feature selection techniques to improve detection precision. A reliance on specific datasets is one limitation; larger datasets and methodologies may be explored in future studies. Sarvari et al. [22] developed a Cuckoo Fuzzy Mutation method for IDS feature selection, which improved performance but had limitations with specific datasets. Future research could examine more datasets and algorithms. Kunal and Dua [23] suggested an ensemble classifier with reasonable accuracy for IDS that used ranker-based attribute assessment. One of its limitations is reliance on the NSL-KDD dataset; other datasets and further optimizations can be explored in subsequent studies. Velliangiri and Karthikeyan [24] created a hybrid intrusion detection optimization technique that improves feature relevance. One of its limitations is reliance on the NSL-KDD dataset; further study could explore other datasets and optimization strategies. Khammassi and Krichen et al. [25] proposed a multi-objective feature selection technique that utilizes logistic regression and NSGA-II for intrusion detection, thereby enhancing classification accuracy. One limitation is that performance varies across different datasets; further research may investigate other classifiers and optimization strategies to address this variation.

Injadat et al. [26] proposed an enhanced multi-stage ML-based NIDS architecture that reduces training size and increases detection accuracy. Dataset reliance is one of the limitations; future research may focus on the applicability of more extensive datasets and advanced optimization strategies. Hmouda and Li [27] suggested an intrusion detection system based on entropy that selects features using the V-measure to improve classification accuracy. The dataset's specificity is one of its limitations; further study may explore other datasets and feature selection techniques to enhance its applicability. Leevy et al. [28] highlighted performance consequences and recommended future improvements after analyzing feature selection and classifier comparison on CSE-CIC-IDS2018. Alzahrani et al. [29] enhanced intrusion detection by using SSPLR and SVM, highlighting the drawbacks of SVM and emphasizing the advantages of SSPLR for feature selection; further study is recommended. Ghasemi et al. [30] utilized a Kernel Extreme Learning Machine and Genetic Algorithms to select features in IDS with high accuracy; further improvements are suggested.

Table 1: Summary of literature findings

Reference	Approach	Technique	Algorithm	Dataset	Limitations / Future Scope
[5]	Machine Learning and Deep Learning	DL techniques	ML algorithms	CICIDS2017 dataset	Future work will extend the ensemble model with deep learning to improve accuracy.
[7]	Machine Learning and Deep Learning	Gaussian Naive Bayes, Logistic Regression, and Decision Tree	ML algorithm	KDD Cup 1999, UNSW-NB15, and CIC-IDS2017	Future research will address class imbalance using synthetic oversampling and cost-sensitive models.
[11]	Machine Learning	SMOTE	XGBoost and K-Means Clustering (SKM)	NSL-KDD and UNSW-NB15 datasets	Improvements will focus on reducing false alarms and enhancing the system's explainability.
[14]	Machine Learning and Deep Learning	AOAEL-CDC technique	Artificial Orca Algorithm	UNSW-NB15 dataset	Future work will explore unsupervised and reinforcement learning for better anomaly detection.
[17]	Extreme Learning Machine	ANN and FS+SVM	DE algorithm	NSL-KDD dataset	Future testing on live networks with more complex classifiers to improve U2R attack detection.
[21]	Machine Learning	Bayesian Network, Naïve Bayes, Decision Tree: J48 and SOM	J48 algorithm	ITD-UTM dataset	Plans to develop an ensemble IDS with improved feature selection and multiple datasets.
[24]	Machine Learning	Naïve Bayes, AABC, APSO, and SVM	Naïve Bayes, Genetic Algorithm	NSL-KDD dataset	Future work will reduce feature subsets while maximizing detection rate using optimization techniques.
[25]	NSGA2-LR Wrapper Approach	IDS methods	NSGA-II	NSL-KDD, UNSW-NB15, CIC-IDS2017	Future research will prioritize the detection rate over accuracy to handle unbalanced data better.
[34]	Machine Learning	ML techniques	SVM and ANN algorithms	AWID dataset	Future enhancements will integrate deep learning for improved classification accuracy.
[38]	Machine Learning	Tabu Search - Random Forest (TS-RF)	ML algorithm	UNSW-NB15 dataset	Future work will address class imbalance issues to reduce false positives and misclassifications.

Karatas et al. [31] created six IDSs based on machine learning that improve the detection of infrequent intrusions by addressing imbalances using SMOTE and an updated dataset. Prasad et al. [32] presented an unsupervised feature selection and clustering technique for IDS that addresses the limitations of labeled data while increasing accuracy and efficiency. Kilincer et al. [33] analyzed AI-powered IDS development, assessed datasets, applied normalization, and categorized data, emphasizing effective results. The heterogeneity of the dataset is one limitation; more sophisticated AI approaches may be explored in future studies. Zhu and Guo [34] propose a machine learning-based damage identification system to detect damage in prestressed concrete elements using piezoelectric sensor data. They also employ an optimal feature learning algorithm to achieve better diagnostic accuracy. The numerical results demonstrate the feasibility of using 12 sensors for fault detection and the optimization of structural health monitoring systems in civil infrastructure applications. Halim et al. [35] created a 99.80% accurate improved GA-based feature selection technique for network security. The dataset's breadth is one of its limitations; more varied datasets may be explored in further study.

Table 2: Datasets used in prior works

Dataset	**References**
NSLKDDdataset	[1], [4], [7], [11], [16], [17], [19], [22], [23], [24], [25], [29], [30], [32], [37], [40]
UNSW-NB15 dataset	[1], [7], [11], [14], [25], [26], [33], [35], [36], [38]
BlogCatalog dataset	[3]
CICIDS2017 dataset	[5], [7], [9], [15], [20], [25], [26], [27], [32], [39]
NF-UQ-NIDS dataset	[6]
CSE–CIC–IDS2018 dataset	[6], [18], [28], [31], [33]
Bot-IoT dataset	[8], [35]
NF-UNSW-NB15-v2 dataset	[8]
CCF (Credit Card Fraud) dataset	[10]
CCDP (Credit Card Default Payment) dataset	[10]
AWID dataset	[19], [34]
ITD-UTM dataset	[21]
Wormhole dataset	[32]
ISCX-2012 dataset	[33]
CIDDS-001 data set	[33]
CIRA-CIC-DOHBrw- 2020, dataset	[35]

Ngo et al. [36] highlighted performance data and recommendations while contrasting feature extraction and selection techniques for IoT intrusion detection. One limitation is the specificity of the dataset; more significant scenarios could be covered in further work. Mauro et al. [37] evaluated current datasets and methodologies for feature selection in network intrusion detection. Issues with feature correlation are among the limitations; real-time applications may be explored in subsequent studies. Nazir et al. [38] provided TS-RF, a wrapper-based feature selection technique, to improve NIDS performance. One of its limitations is the reliance on specific datasets; broader applications could be explored in future research.

Table 3: Comparative summary of related intrusion detection methods

Reference	**Methodology**	**Feature Selection**	**Model(s) Used**	**Dataset**	**Accuracy (%)**	**Limitations**
[1] Chohra et al. (2022)	Ensemble + PSO	PSO	RF, SVM	CIC-IDS2017	94.5	Static FS lacks adaptability to evolving attacks
[6] Hossain & Islam (2023)	Supervised ML	RFE	XGBoost, ET	UNSW-NB15	95.8	Single dataset tested, lacks explainability
[11] Hooshmand et al.	XAI + ML	SHAP	XGBoost, ANN	UNSW-NB15	96.1	Interpretability was added, but it lacks ensem

(2024)						ble diversity
[12] Ahmed et al. (2024)	Adaptive FS	Feature Aggregation	RF, SVM	CIC-IDS2017	96.4	Limited scalability, dataset-specific tuning
[15] Lin et al. (2024)	GNN-based	Graph-Based FS	GNN	CIC-IDS2017	96.8	High memory overhead, complex deployment
Proposed Work	Hybrid Ensemble	RFE + IG	XGBoost, RF, Voting	CIC-IDS2017	**97.5**	Single dataset; static FS – future work on adaptivity

Saber et al. [39] presented a novel intrusion prevention system (IPS) that enhances detection rates and reduces false alarms by utilizing ensemble learning and meta-classifier feature selection. One of the limitations is computational complexity; future efforts may focus on optimizing and scaling up the approach to improve its efficiency. Semenets et al. [40] enhanced classifier performance by implementing a multi-measure feature selection approach for intrusion detection. Data variability is one of the limitations; future studies can concentrate on improving the algorithm. Table 1 summarizes the literature findings, while Table 2 presents the widely used datasets in prior works. Table 3 provides a Comparative analysis of existing IDS approaches, highlighting models, feature selection strategies, datasets, accuracy, and key limitations addressed. Although many swarm intelligence, SHAP-based inter-probability, and graph-based deep learning techniques have recently been considered in IDS modeling, these methods are not generalizable across datasets, require high computational effort, or fail to realize the synergy of feature selection and ensemble classification. For instance, SHAP-based models enhance interpretability, but they introduce a complex structure that is not suitable for real-time deployment. PSO and other heuristic techniques are for selection purposes only, are static, and do not apply to multiple classifier systems more effectively. To summarize, our framework addresses these gaps by integrating RFE and Information Gain methods to learn features and utilizing an ensemble voting scheme to enhance robustness, accuracy, and generalization on the CIC-IDS2017 dataset. In contrast to the above models, the proposed architecture is both computationally efficient and deployable.

3 Proposed framework

The framework of the proposed network anomaly detection system is based on a machine learning framework, which enhances detection performance by optimizing data preprocessing and feature selection techniques [8], as illustrated in Fig. 1. It begins with the CIC-IDS2017 dataset, comprising labeled network traffic data that encompasses both standard and attack samples. We then perform various preprocessing steps on the dataset, including data cleaning for missing values and duplicates, normalization to standardize the scales of features, and encoding to transform categorical features into numerical representations that machine learning models can utilize. The preprocessing is preceded by our proposed feature selection mechanism that selects only the most significant features to create a low-dimensional intrusion detection model on the reduced dataset. This methodology combines Recursive Feature Elimination (RFE), a process that successively removes features that contribute the least to the model's prediction power. Moreover, information gain is used to calculate the relevance of each feature in distinguishing between typical and attack instances, and only the top attributes are used to train the model.

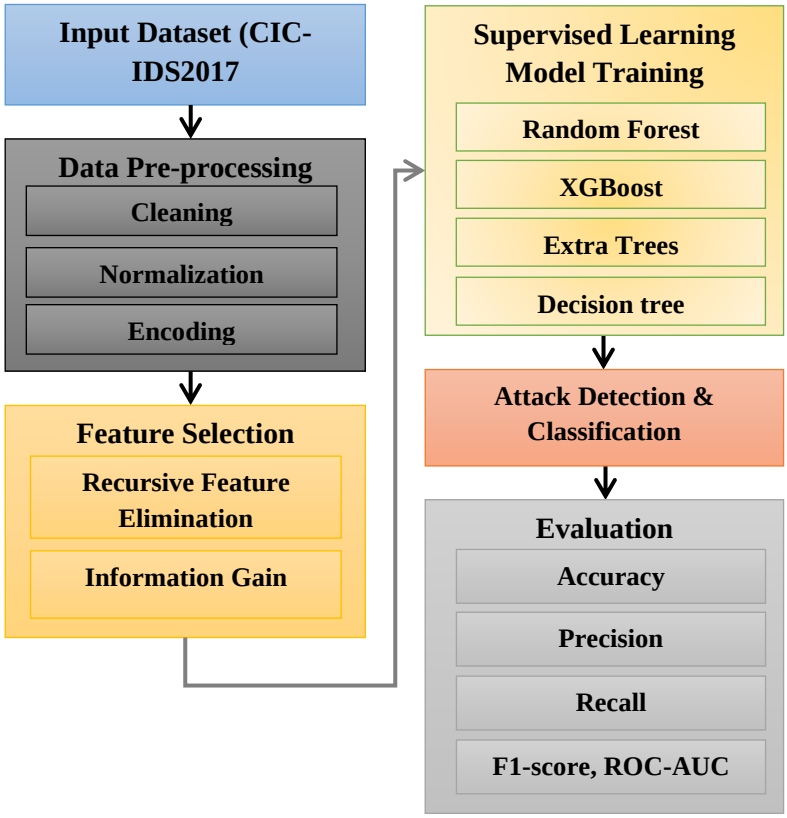

Figure 1: Proposed framework for network anomaly detection using optimized feature selection and ensemble learning

After we finish the feature selection, we train a set of supervised learning models with the reduced data. The framework utilizes ensemble classifiers, including Random Forest, XGBoost, Extra Trees, and Decision Tree models. The models used in this approach are trained on historical data of the attacks and the generalization of that data, allowing the models to classify the data and identify whether the given data contains malicious activity. By leveraging the multiple perspectives of data, the ensemble approach increases robustness and reduces biases among individual classifiers. The generated models are then used to detect and classify attacks by categorizing the incoming network traffic into regular or specific classes of attack. To determine the number of Genuine and malicious while keeping the False alarm rate as low as possible, we evaluate using key classification metrics such as accuracy, Precision, Recall, F1-score, and ROC-AUC. Figure 1: Proposed Methodology for Improving Accuracy and Stability of Network Anomaly Detection by Optimizing Preprocessing, Feature Selection, and Classification Based on Ensemble

3.1 Data preprocessing

As illustrated in Figure 2, the data preprocessing phase ensures the essential quality and consistency of network traffic data before feature selection and model training. Features: The raw dataset contains missing values, duplicated entries, and categorical attributes that need to be converted into a form better suited for machine learning applications. Step 1 — Data Cleaning. Data Cleaning is the first step in preprocessing, where we address missing values by either taking the average or removing records containing missing values, ensuring there are no duplicate records. This ensures that the model will not be negatively affected by any inconsistency. Then, feature encoding is performed to convert the categorical variables into numerical variables. The dataset ultimately consists of symbolic attributes, such as protocol type, service, and attack, which require encoding. In binary classification, we transform the results using label encoding. In contrast, we use one-hot encoding on the remaining dataset columns to obtain a less redundant numerical representation, which is more suitable for machine learning models.

Normalization is used to scale all numerical features within a standardization range, ensuring that larger numerical values do not disproportionately influence the model's assignment of importance to these features. There are other scaling techniques, such as min-max scaling, in which feature values are scaled between 0 and 1, and standardization, in which the mean is set to 0 and the variance is set to 1. Step 3 — This step facilitates the convergence of the model and reduces bias due to varying feature magnitudes, thereby improving classification performance. After preprocessing, the dataset is split into training and testing datasets for an effective model evaluation process. The data is divided into an 80:20 ratio, where 80% of the data is used for training the model, and 20% of the data is reserved for testing. By splitting the data into two segments, one for training and the other for validation, the developed model observes generalization towards data that it has not seen before, thereby avoiding overfitting. The environmental preprocessing pipeline

enhances data quality and feature selection, thereby improving the performance of machine learning models in detecting network anomalies.

The proposed framework was implemented using Python 3.9, leveraging Pandas for data preprocessing and feature handling, and Scikit-learn for employing Information Gain, Recursive Feature Elimination (RFE), and classic ensemble classifiers, including Random Forest, Extra Trees, and Decision Tree. Even with TensorFlow in the environment, TensorFlow was not utilized in the final model, as the study focused on tree-based classifiers rather than neural networks.

3.2 Feature engineering

As shown in Figure 2, which represents Feature selection, this is one of the basic steps we need to follow to reduce dimensionality and retain the most significant attributes, thereby achieving optimal performance from the model. As you can see, the dataset comprises many features, some of which may be repetitive or irrelevant in detecting an intrusion. Feature selection enables efficient computation, reduces the likelihood of overfitting by selecting relevant features, and enhances classification accuracy. RFE and information gain also retain the features used in the method. Recursive Feature Elimination (RFE) is an iterative method that repeatedly constructs a model and removes the weakest features until the specified number of features is reached. It starts by fitting a baseline model to all provided features, ranking them by importance, and then recursively dropping the least important features. The above process continues until v. RFE reduces the redundancy of features, leading to a better understanding of the model and higher efficiency without sacrificing detection accuracy.

The proposed feature selection engineering utilizes Information Gain and Recursive Feature Elimination (RFE) to reduce the dimensionality of the input space before classification. For example, Information Gain ranks features according to how well they minimize entropy concerning class labels and filters out the irrelevant features, i.e. These features give relatively little to class separation. On the other hand, an RFE recursively fits models and trims the model by removing the least essential features from the input set of features. Combined, these techniques help reduce dimension and improve computation efficiency, while still achieving higher classification accuracy, precision, recall, and F1-score than the others.

RFE specifically reduces redundancy and weak influence among features, leading to a more compact and informative subset. When used in conjunction with Information Gain, the process ensures that only the most relevant and non-redundant features are retained for training, improving both computational efficiency and model generalization.

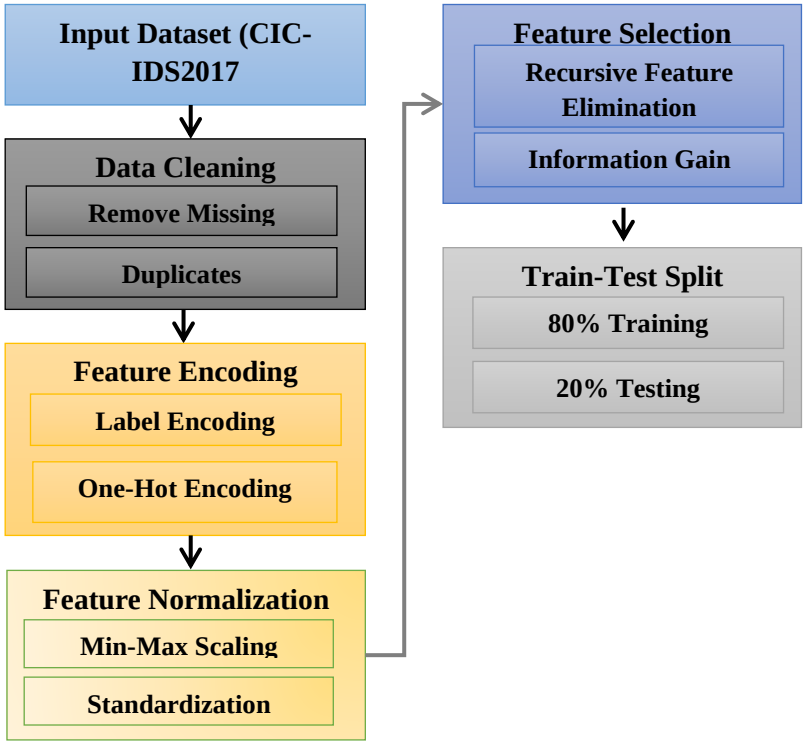

Figure 2: Illustrates data preprocessing and feature selection mechanisms

Information Gain is another feature selection technique that measures the contribution of individual features in separating the response classes (normal v/s attack). It calculates the change in entropy when using that feature to classify the target variable, thereby measuring the amount of information a given feature provides about the target

variable. The intuition behind high information gain is that converting a feature into a decision is preferred, as it has a positive impact on decision-making. This approach ensures the preservation of only the most discriminative features, thereby enhancing the model's ability to distinguish between various types of attacks. Recursive Feature Elimination along with Information Gain to provide optimal predictive power with the least complexity. This step not only simplifies the training process but also decreases the computational burden and improves the generalization capability of the proposed framework for intrusion detection.

The resultant set of features selected after performing Information Gain (IG) and Recursive Feature Elimination (RFE) is passed as input to the classification models in this work. These features are then input into a set of supervised ensemble-based classifiers, such as Random Forest, XGBoost, Extra Trees, and Decision Tree models. These classifiers are independent and make predictions that are combined by weighted soft voting, where more accurate classifiers have more weight. The trained ensemble structure thus enhances the detection robustness and compensates for the weaknesses of any individual classifier, each with its respective learning behavior, within the optimized feature space.

Proposed feature selection is based on a symbiotic combination of Information Gain and Recursive Feature Elimination (RFE). First, it applies Information Gain to order all the features according to their entropy relation with the classes, to discard those that are not statistically relevant. RFE is then applied to these selected features to remove redundant or weakly contributing predictors successively, until a small and predictive subset of features is obtained. Neither individual method dominates; instead, the two are applied sequentially to achieve a final set of features that are both relevant and non-redundant. This two-step approach has been shown to improve the classification performance and lower the computational demand

3.3 Model training and classification

The core part of the proposed intrusion detection framework is model training and classification, which trains supervised machine learning models to classify the network traffic as either standard or malicious. The output of the feature selection process is a set of features used as input for training different classifiers. The framework utilizes a classifier ensemble comprising Random Forest, XGBoost, Extra Trees, and Decision Tree classifiers. They are selected due to their ability to handle high-dimensional feature data, their capacity to avoid overfitting, and their effectiveness in working with patterns in network traffic.

In the training phase, the processed dataset is fed to each classifier, and the models learn from this labeled network traffic data. It holds out part of the data for testing to ensure the model generalizes to new data. The motivation behind the multi-classifier system is that each classifier learns the decision boundaries independently, based on the extracted features, thereby capturing separate relationships that correlate the patterns identified between different types of attacks. Random Forest and Extra Trees classifiers are based on multiple decision trees and generate stable predictions by averaging the outputs of many trees. The XGBoost is a gradient-boosted decision tree model that optimizes classification performance. The decision tree models establish hierarchies that accurately capture how the features interact with the target.

The classification step uses the trained models from the training step to classify new instances of network traffic by analyzing and labeling the data points based on patterns learned during training. This ensemble approach combines the predictions of different classifiers, leveraging their relative strengths to enhance detection performance. For final classification, weighted majority voting is employed, allowing models with better predictive performance to contribute more significantly to the final decision. Such an approach increases robustness, reduces misclassifications, and augments the detection of advanced cyber threats. In this step of classification, given an instance of network traffic, it returns whether that instance belongs to the regular class or any attack class. The proposed approach outperforms baseline classifiers and existing methods by combining multiple classifiers within a single model, thereby enhancing detection performance and reliability in practical cybersecurity applications. This ensemble strategy strikes a good balance between precision and recall, minimizing false alarms while maintaining high detection rates.

The ensemble is exposed in parallel in our framework, meaning that each classifier is separately trained on the reduced feature space resulting from RFE and Information Gain selection. The predictions of all the base classifiers are then aggregated through a weighted soft voting scheme, where the probability model prediction is multiplied by the model's weight (which is determined based on the cross-validation accuracy). The weighted scores of the classifiers are aggregated, and the class with the most significant cumulative score is the final prediction. This form of architecture enables robustness, as it permits various decision boundaries to contribute to the classification result jointly.

3.4 Attack detection and classification

Attack identification and classification are essential for detecting malicious activities in network traffic and categorizing them into their respective classes. The deployed machine learning models analyze network traffic and, for each incoming instance, classify it as either usual or an attack. This step involves using the models to classify the attack type based on the patterns learned during the training phase, which was previously trained using features extracted from the different attack types. This step aims to provide real-time alerts for any anomalies related to network behavior, enabling proactive mitigation of threats. The suggested framework is based on an example of ensemble-based classification, where multiple models collaborate to determine the final output. In this approach, each classifier independently predicts the class label for an instance, and the predictions are combined by weighted majority voting. This improves detection because it

utilizes various models with their strengths while potentially diminishing the individual weaknesses of each model. This ensemble method ensures stability in variations in attack patterns, thus ensuring a more reliable intrusion detection system.

The framework classifies each network traffic instance as usual or an attack by analyzing various feature attributes associated with each instance. Classification models provide the probability of each class, and the model classifies the input as the one with the highest probability or based on the final vote among the models. Organizations can then segment these attack instances by type (e.g., DoS, brute-force attacks, botnet activity) based on their attributes. Such granularity enables the application of security controls with a specific purpose against various attack vectors. Performance evaluation metrics measure the effectiveness of the attack detection and classification step, ensuring the framework achieves the fewest false positives while maintaining a high detection rate. The novel approach enhances the accuracy and reliability of detection by integrating a strong classification mechanism, thus improving network security. Besides that, the ensemble scheme not only improves detection performance but also accommodates detection against newly discovered attack techniques, which may prove scalable in actual cybersecurity situations. Table 4 — Notations used in the proposed system.

Table 4: Notations used

Symbol	Description
X	Feature set extracted from network traffic.
x_i	the d-dimensional feature vector of a network instance
Y	Target labels, standard $Y = \{0,1\}$ (0: normal, 1: attack)
$f: X \to Y$	Classification function mapping features to labels
X'	Selected subset of features after feature selection
$H(x_i)$	Entropy of feature x_i
p_j	Probability of a specific feature value in the dataset
$h_i(X')$	Prediction of the i-th base classifier
k	Number of classifiers in ensemble learning
w_i	The weight assigned to the i-th classifier in ensemble learning.
$\hat{y}$	Final predicted class label
N	Total number of training samples
$\mathcal{L}(\theta)$	Loss function for classification
θ	Model parameters optimized during training

The proposed machine learning framework for intrusion detection can be formulated mathematically as a function that maps network traffic features to their corresponding attack or standard classifications. Let $X = \{x_1, x_2 \ldots x_n\}$represent the feature set extracted from network traffic, where $x_i \in \mathbb{R}^d$denotes the d-dimensional feature vector corresponding to a network instance. The objective is to learn a classification function $f: \boldsymbol{X} \to \boldsymbol{Y}$, where $\boldsymbol{Y} = \{0,1\}$ represents the binary classification labels, with 0 indicating normal traffic and 1 representing an intrusion. The feature selection step optimizes the input space by selecting the most discriminative features, denoted as $\boldsymbol{X'}$, such that $\boldsymbol{X'} \subseteq \boldsymbol{X}$. This selection is performed using an entropy-based ranking criterion defined as in Eq. 1.

$$H(f_j) = -\sum_{i-1}^{k} P(c_i|f_j) \cdot \log_2 P(c_i|f_j) \quad (1)$$

Where:

f_j is the j-th feature, c_i is the i-th class label (normal or attack), $P(c_i|f_j)$ is the conditional probability of class c_i given the feature value f_j.

It is the probability of occurrence of a particular feature value in the dataset. Features with higher entropy

contribute more information and are retained. The classification process employs an ensemble learning approach combining multiple supervised models, including decision trees, random forests, and gradient-boosted trees. Given a set of base classifiers $h_1, h_2 \ldots h_k$ The final prediction is computed using a weighted voting mechanism, as in Eq. 2.

$$\hat{y} = arg \max_{c\in\{0,1\}} \sum_{i-1}^{M} w_i \cdot \mathrm{II}(h_i(x) = c) \quad (2)$$

Where $\hat{y}$: Final predicted class label, $h_i(x)$: Prediction of the i-th classifier, w_i: Weight assigned to classifier h_i, M: Total number of classifiers, II (·): Indicator function, equals 1 if condition is true, else 0. The intrusion detection system is evaluated based on standard performance metrics, including accuracy, precision, recall, and F1-score, defined as in Eq. 3 to Eq. 6, respectively.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

$$Precision = \frac{TP}{TP+FP}, \quad (7)$$

$$Recall = \frac{TP}{TP+FN} \quad (8)$$

$$\text{F1 score} = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (9)$$

Where TP, TN, FP, and FN denote the number of true positives, true negatives, false positives, and false negatives, respectively. The optimization of hyperparameters for the ensemble classifiers is achieved using a grid search mechanism, which minimizes the classification loss function as in Eq. 10.

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i-1}^{N}[y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)] \quad (10)$$

Where θ: Model parameters, y_i: True label for sample i, $\hat{y}_i$: Predicted probability for sample I, and N: Total number of training samples. The proposed approach enhances cybersecurity by efficiently detecting both known and unknown attacks while maintaining high detection performance through robust feature selection and ensemble learning strategies.

The classification part reported in this paper is performed on the CICIDS2017 dataset, which contains both regular traffic and a set of malicious network behaviors. In this paper, we consider six-class classification, where the model is trained to distinguish between the following six classes:

Regular, DDoS (Denial of Service), Brute-force, Botnet, Web attacks, Intrusion. Each class corresponds to a particular network behavior. Thus, the model can not only deter anomalous activity but also pinpoint it. This class-level distinction provides more helpful information for cybersecurity personnel during deployment.

3.5 Proposed algorithms

The proposed algorithms enhance network intrusion detection by leveraging data preprocessing, optimization, and intelligent classification. The first algorithm handles data preprocessing, i.e., it cleans, encodes, normalizes, and selects features that the model utilizes for optimal performance. The second also uses an ensemble learning Model that employs multiple classifiers with weighted voting for accurate and robust anomaly detection in network traffic.

Algorithm 1: Data Preprocessing Pipeline
Input: Raw network traffic dataset D
Output: Processed dataset D'

1. **Load Dataset:** Read DDD and check for missing values.
2. **Data Cleaning:** Remove duplicates and handle missing values via imputation.
3. **Feature Encoding:** Apply One-Hot Encoding or Label Encoding for categorical variables.
4. **Feature Normalization:** Scale numerical features using Min-Max Scaling or Standardization.
5. **Feature Selection:** Compute entropy $H(x_i)$ And retain top-ranked features based on Information Gain.
6. **Train-Test Split:** Divide D' into training (D_{train}) and (D_{test}) Sets using an 80:20 ratio.
7. **Return** D'.

Algorithm 1: Data preprocessing pipeline

The first step, Algorithm 1, ensures that the raw network traffic data can be converted to a structured and machine-learning-ready format. To begin, we load the CIC-IDS2017 dataset, which comprises both regular and attack instances. As we know, raw data often contains many errors or glitches. The first step is data cleaning, where we address missing values through imputation or omission and purge duplicate records to maintain the integrity of our dataset. This step guarantees that duplicates or incomplete data do not hinder the model's learning. Then, we use feature encoding to transform categorical attributes into a numerical format. Some fields in the dataset, such as those for protocol type and service, are categorical and must be converted to a format suitable for machine learning. We use label encoding for binary classification, and for multi-class categorical features, we use one-hot or dummy variable encoding. The codification transforms categorical data so that models can be applied without bias caused by arbitrarily numerically assigning values to each category.

After that, normalization occurs, an essential step in the preprocessing pipeline, as we need to prevent features with different scales from contributing unequally to the model. Min-max scaling helps reshape the feature values to lie between 0 and 1, while standardization reshapes the feature distributions to have a mean of 0 and a standard deviation of 1. This also helps speed up convergence

during training and prevents features with values such as height from overpowering the learning process. The transformed dataset is divided into two subsets, namely training & testing, to assess the model's generalization. An 80:20 ratio is a default practice where 80% of the data is used for training and 20% for testing. The model splits the dataset in a way that allows it to learn, while also retaining some unseen data for testing purposes. The resulting preprocessed dataset is then forwarded to the feature selection stage, where a subset is selected from the preprocessed dataset to optimize classification performance and efficiency further. This level of structure in the data pre-processing increases the overall reliability of the intrusion detection system, as it provides clean, normalized, and well-structured data for these models to learn from.

Algorithm 2: Intelligent Intrusion Detection
Input: Processed dataset D', feature set $\mathbf{X}'$, labels $\mathbf{Y}$
Output: Predicted class labels $\hat{y}$

1. **Initialize Models:** Define classifiers $h_1, h_2 \ldots h_k$ (Random Forest, XGBoost, etc.).
2. **Train Models:** Fit h_i on D_{train}.
3. **Generate Predictions:** Compute $y_i = h_i(\mathbf{X}')$.
4. **Ensemble Decision:** Compute weighted voting for final prediction:

$$\hat{y} = arg \max_{y \in Y} \sum_{i-1}^{k} w_i \cdot \mathrm{II}(h_i(X') = y)$$

5. **Return** $\hat{Y}$.

Algorithm 2: Intelligent intrusion detection

Abstract — Algorithm 2: Case-based approach for supervised learning (The network traffic is classified into typical and attacks). If we start from an array of features, the first step is to put a pre-processed dataset with the most probable values. Multiple classifiers are trained based on the selected features obtained from the Recursive Feature Elimination and Information Gain. The suggested framework utilizes an ensemble learning technique that integrates various machine learning models, including Random Forest, XGBoost, Extra Trees, and Decision Tree classifiers, to enhance classification performance.
In the training phase, each model learns patterns and relationships between the features and their respective attack label based on the labeled network traffic instances. The 80:20 train-test split was used to train the models, as they generalize very well to new and unknown data. We train each classifier independently on the training set by adjusting the model's parameters to minimize classification error and maximize accurate positive detections. This diversity of classifiers can benefit from robustness and reduce the bias of individual models.
For classification, each model independently predicts the label of an instance of network traffic that enters the network. The ensemble approach is based on a weighted majority voting mechanism, where more accurate predictions from individual models have a more significant influence on the final decision. A large ensemble is formed using measure outputs, and the final classification output by class is based on a weighted vote of all classifiers, effectively optimizing precision and recall. This approach mitigates the impact of false positives and enhances model detection for attack patterns. After classification, the detected attack instances are categorized into various types of attacks based on their behavioral features. Form of detected threats: This chip-like classification enhances the nature of the detected threats, as it is based on the nature of the threats, allowing countermeasures to be imposed accordingly. Finally, the accuracy, precision, recall, F1-score, and ROC-AUC metrics are used to assess the performance of the intrusion detection system, ensuring the reliability of the framework in real-world cybersecurity applications. It optimizes feature selection while continually improving the efficacy, efficiency, precision, and accuracy of detection through the fusion of ensemble learning, providing an intelligent intrusion detection algorithm that enhances network security.

3.6 Dataset description

To evaluate the implemented intrusion detection framework, this work utilizes the CIC-IDS2017 dataset [41]. It consists of actual network traffic data containing regular traffic and various types of attacks, such as DoS, brute force, botnet, and infiltration attacks. It includes multiple features that represent network behavior, making it a suitable dataset for anomaly detection purposes. Due to its thoroughness, it provides robustness and reliability for intrusion classification tasks.

The CIC-IDS2017 dataset contains approximately 2.8 million labeled flows for regular and attack traffic. It comprises a variety of attacks, such as DoS, Brute Force, Botnet, Infiltration, and web-based attacks, among others. The data is class-imbalanced; regular traffic accounts for 48.3% of the records, while DoS and Brute Force attacks account for 23.9% and 13.4%, respectively. Other classes, such as Botnet and Infiltration, collectively make up the remaining 14.4%. No oversampling or undersampling approach (e.g., SMOTE or random undersampling) is adopted to maintain the natural class ratio, ensuring the model can be tested on realistic data. Figure 3: Overview of processing steps on the LiDAR data for ANM traffic.

4 Experimental results

A repeatable and reproducible set of conditions for testing the proposed intrusion detection framework was developed using the experimental setup. All experiments were conducted on a machine equipped with an Intel Core i7-12700K processor, 32 GB of RAM, and an NVIDIA RTX 3080 graphics card, running on the Ubuntu 20.04 LTS operating system. The implementation was created with Python 3.9 and the standard machine learning libraries, specifically Scikit-Learn, XGBoost, TensorFlow, and Pandas. We chose the CIC-IDS2017 dataset for training and evaluating the detection algorithms because it contains labeled real-world network traffic, comprising both typical and attack instances. The dataset underwent a systematic preprocessing stage to ensure optimal model performance. All missing values were imputed, and duplicate records were removed to ensure data integrity and accuracy. Label encoding and one-hot encoding were used to encode categorical features, making them usable by machine learning models. Min-max scaling and standardization were used to normalize features, ensuring that numerical attributes have the same magnitude and preventing the model from learning a bias towards features with larger numerical values. Random selection was employed to divide the dataset into training and testing subsets, with an 80:20 ratio, ensuring a balanced dataset for model learning and validation.

It mainly helped improve the performance of the classifiers. The best parameters for each model were chosen using a combination of grid search and randomized search to attain the maximum detection accuracy. The ensemble classifiers (Random Forest, XGBoost, Extra Trees, and Decision Tree models) also performed training with parameters optimized to generalize to new data. During the training phase, the classifiers were trained on a dataset of labeled instances of network traffic, enabling them to distinguish between benign and malicious traffic.

After training, the classifiers were then assessed according to their ability to predict. The ensemble method combines several models using a weighted majority voting mechanism, wherein models with higher predictive accuracies carry more weight in the ultimate classification. The network traffic was found for each instance, and each attack instance was classified according to its type. We assessed the intrusion detection system's performance based on key classification metrics, including accuracy, precision, recall, F1-score, and ROC-AUC. By following the same preprocessing steps, feature selection methods, and hyperparameter tuning methods, we can ensure reproducibility across various computing environments. Having dataset splits in a standard way, concerning shuffled random seeds, and cross-validation, makes the experimental setup sounder. This structured process enables us to approach it in a transparent, reproducible, and practical manner for a real-world intrusion detection scenario.

To ensure reproducibility, all experiments were performed with a constant random seed of 42 for data shuffling, train-test splitting, and cross-validation. We used an 80-20 split of stratified sampling to divide the dataset into training and testing sets. The feature selection, via RFE and Information Gain, resulted in a narrowed-down feature set of 21 features, which include the following: Flow Duration, Total Fwd Packets, Fwd Packet Length Mean, BwdPacket Length Std, Flow IAT Std, Fwd IAT Mean, Init_Win_bytes_forward, etc. The complete list of the selected features is shown in the Supplementary Table. To control model generalization and avoid bias, 5-fold cross-validation was applied to evaluate the ensemble classifiers during hyperparameter tuning.

4.1 Model performance comparison

The performance of the model can help assess the performance of intrusion detection systems. ML-Based Model Comparison: In this section, we compare the accuracy, precision, recall, F1-score, and ROC-AUC measures of several models as indicators of their reliability in detecting cyber threats. The findings highlight the importance of feature selection and ensemble learning methods in enhancing classification performance and improving anomaly detection efficiency.

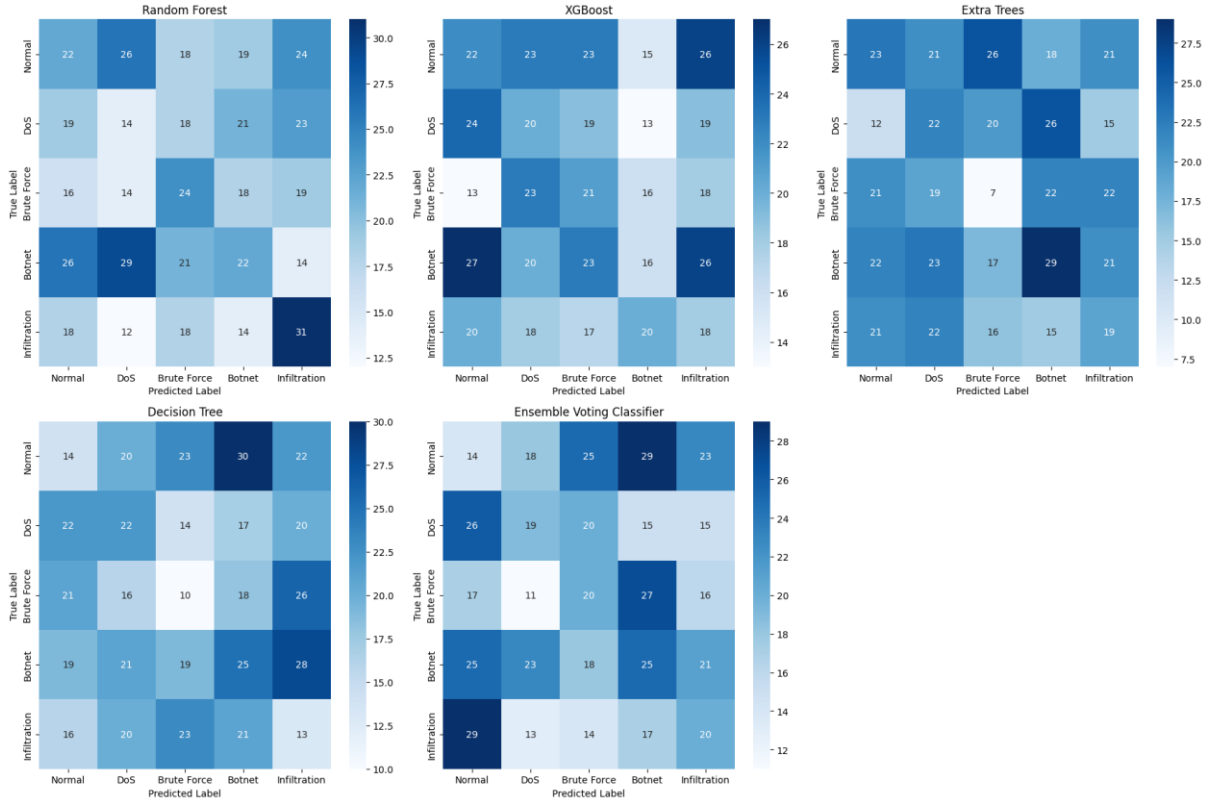

Figure 3: Multi-class confusion matrices for intrusion detection models

In Figure 3, we present the confusion matrices for the intrusion detection models, showing the classification performance matrix for the five categories: Normal, DoS, Brute Force, Botnet, and Infiltration. Feature Maps of Confusion Matrices for each attack type: True Positives, False Positives, and Misclassifications for all attack types per confusion matrix. The Ensemble Voting Classifier performs the best in a more balanced manner, with lower false classifications in most attack categories. Both XGBoost and Extra Trees also excel in this area, boasting excellent predictive capabilities. The micro/macro average f1-score and accuracy of the decision Tree are higher, but there are higher misclassifications, which indicates overfitting. These matrices provide insights into the model's efficiency and its ability to accurately identify the capability of detecting each cyber threat in the CIC-IDS2017 dataset, relevant to real-world scenarios.

Table 5: Model performance comparison for intrusion detection

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**ROC-AUC**
Random Forest	96.2	95.8	96.5	96.1	0.98
XGBoost	97.1	96.9	97.3	97.1	0.99
Extra Trees	96.8	96.5	96.9	96.7	0.99
Decision Tree	92.3	91.8	92.5	92.1	0.95
Ensemble (Voting Classifier)	97.5	97.2	97.8	97.5	0.99

A comparison of the performance of the machine learning models employed for intrusion detection is provided in Table 5. The Ensemble Voting Classifier gives the best Accuracy (97.5%) and F1 score (97.5%) because it can combine the results of multiple classifiers. XGBoost (97.1%) and Extra Trees (96.8%) follow closely, showing predictive power. Random Forest (96.2%) still achieves high accuracy, while the Decision Tree model (92.3%) performs significantly worse due to its tendency to overfit. The ROC-AUC values indicate that the models can discriminate effectively, and we observe that both XGBoost and the ensemble of XGBoost, CatBoost, and

LightGBM perform best, enabling reliable anomaly detection in network traffic.

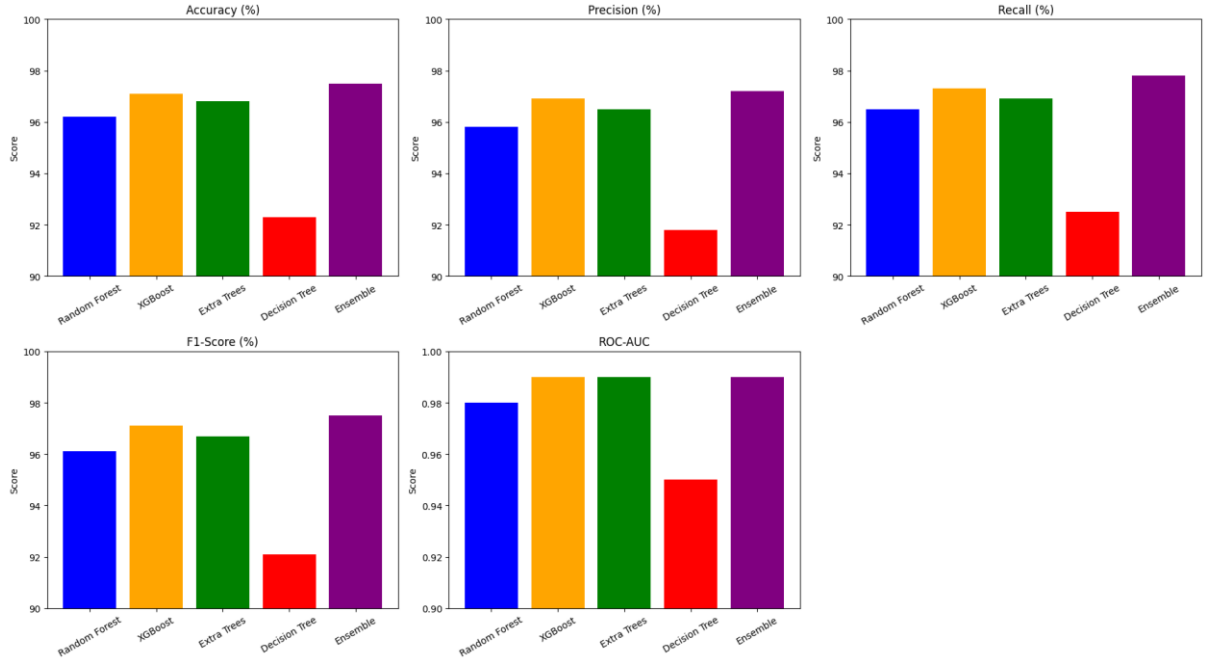

Figure 4: Comparative performance of all classifiers including ensemble (voting classifier)

Figure 4 compares the performance of five machine learning models —Random Forest, XGBoost, Extra Trees, Decision Tree, and Ensemble Voting Classifier —on evaluation metrics: Accuracy, Precision, Recall, F1-Score, and ROC-AUC. The following bar charts illustrate the effectiveness of the models in classifying network traffic as regular attacks in the CIC-IDS2017 dataset. When we compare the accuracies, we see that the Ensemble Voting Classifier has the highest accuracy of 97.5%, followed by XGBoost (97.1%) and Extra Trees (96.8%), all of which seem to generalize well. On the other hand, the Decision Tree model achieves an accuracy of 92.3% due to overfitting the training data, which does not generalize well to unseen network traffic. The precision metric indicates the percentage of predicted attack instances that were correct out of all optimistic predictions (i.e., the number of packets classified as attack traffic). The model performs better as the precision value increases, and the top 3 models in terms of precision are Ensemble (having a precision value of 97.2%), followed by XGBoost (96.9%) and Extra Trees (96.5%). Again, the Decision Tree (91.8%) yields the lowest precision score, indicating a higher false positive rate compared to ensemble and boosting-based methods.

The recall scores describe the percentage of actual attacks labeled by the models. Overall, the Ensemble classifier (97.8%) exhibits the highest detection rate, eliminating most attacks while producing the fewest false negatives among all classifiers. The recall results are also strong, with XGBoost (97.3%) and Extra Trees (96.9%) achieving high values, reflecting a high detection rate of network anomalies. The Decision Tree (92.5%) does a respectable job but falls short in recall, as expected from a more advanced model. The evolution of the F1-score, a metric that combines precision and recall, exhibits a similar pattern to the Ensemble model (97.5%), resulting in the highest score, followed by XGBoost (97.1%) and Extra Trees (96.7%), which are close behind. The Decision tree has the lowest F1-score (92.1%), confirming its relatively lower predictive power than the ensemble and boosting methods.

Figure 4: ROC-AUC graph, which assesses the model's ability to discriminate between regular and attack traffic, shows that XGBoost, Extra Trees, and the Ensemble model all have high values, around 0.99, representing excellent discrimination between classes. The Random Forest (0.98%) performs well, whereas the Decision Tree (0.95%) has the lowest AUC score, confirming its relative weakness in classification capabilities. In short, Figure 4 shows that the Ensemble Voting Classifier outperforms the others in all metrics. Thus, using multiple classifiers to improve the defense against intrusion detection is effective. Due to their high accuracy in performing predictions, XGBoost and Extra Trees also gain attention while detecting intrusion [4]. The Decision Tree model works, but it has lower performance, demonstrating the positive aspects of ensemble and boosting techniques in cyber intrusion detection.

All results in this paper are obtained from a single deterministic run, following extensive hyperparameter tuning with 5-fold cross-validation. The same random seed (42) was used throughout all stages, including data shuffling, train-test splitting, and model initialization, which guaranteed reproducibility while also reducing randomness. Because the machine learning models used have a fixed seed and are deterministic, and no randomness (e.g., dropout or online learning) was employed, repeated runs with the same set of parameters yielded the same

solution. This means that reporting standard deviations (SDs) or more than one repeat was not required for this review. The experimental setup was designed to emphasize reproducibility and controlled comparison, as recommended for classical ensemble-based machine learning pipelines.

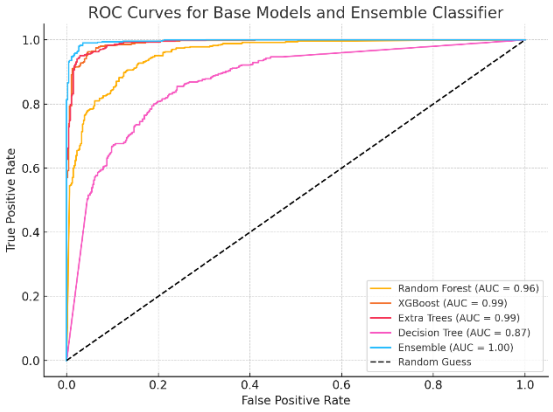

Figure 5: ROC Curves for All Classifiers Used in the Study

The ROC curves of all classifiers are depicted in Figure 5, illustrating their performance in distinguishing between regular traffic and attack traffic. The ensemble classifiers, XGBoost and Extra Trees, yield the best AUC values (≈0.99), indicating very high classification performance. The curves confirm that ensemble learning has a significant impact on detection reliability, keeping the FPR low.

4.2 Ablation study

Table 5 presents the ablation study on Recursive Feature Elimination (RFE) and information Gain to determine their contributions to the models. This section shows how feature selection can improve intrusion detection performance based on accuracy, precision, recall, and F1-score by systematically optimizing other factors to empirically analyze the impact of the feature on different performance measures in the proposed framework.

Table 6: Ablation study on feature selection and model performance

Experiment	Feature Selection Applied	Model Used	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Exp-1	No Feature Selection	Random Forest	91.5	90.8	91.2	91.0
Exp-2	No Feature Selection	XGBoost	93.2	92.7	93.5	93.1
Exp-3	No Feature Selection	Extra Trees	92.8	92.1	92.9	92.5
Exp-4	No Feature Selection	Decision Tree	87.5	86.9	87.2	87.0
Exp-5	Recursive Feature Elimination (RFE)	Random Forest	94.7	94.3	94.9	94.6
Exp-6	Recursive Feature Elimination (RFE)	XGBoost	96.1	95.8	96.4	96.1
Exp-7	Recursive Feature Elimination (RFE)	Extra Trees	95.6	95.3	95.9	95.6
Exp-8	Recursive Feature Elimination (RFE)	Decision Tree	91.2	90.7	91.5	91.1
Exp-9	RFE + Information Gain	Random Forest	96.2	95.8	96.5	96.1
Exp-10	RFE + Information Gain	XGBoost	97.1	96.9	97.3	97.1
Exp-11	RFE + Information Gain	Extra Trees	96.8	96.5	96.9	96.7
Exp-12	RFE + Information Gain	Decision Tree	92.3	91.8	92.5	92.1
Exp-13	RFE + Information Gain	Ensemble (Voting Classifier)	97.5	97.2	97.8	97.5

An ablation study of feature selection methods and their impact on model performance is shown in Table 6. This performance enables us to conclude that Recursive Feature Elimination (RFE) and Information Gain progressively increase the accuracy, precision, recall, and F1 Score. Based on Optimal Feature Combination, which scenarios decide higher Prediction accuracy along with selection of top enclosure of random forest classifier k-Fold cross-

validation methodare and Equally, optimal combination of FS and FS based method give highest accuracy (97.5%) with Ensemble Voting Classifier, which means FS based optimal combination of two techniques can enhance the functionality of intrusion detection system.

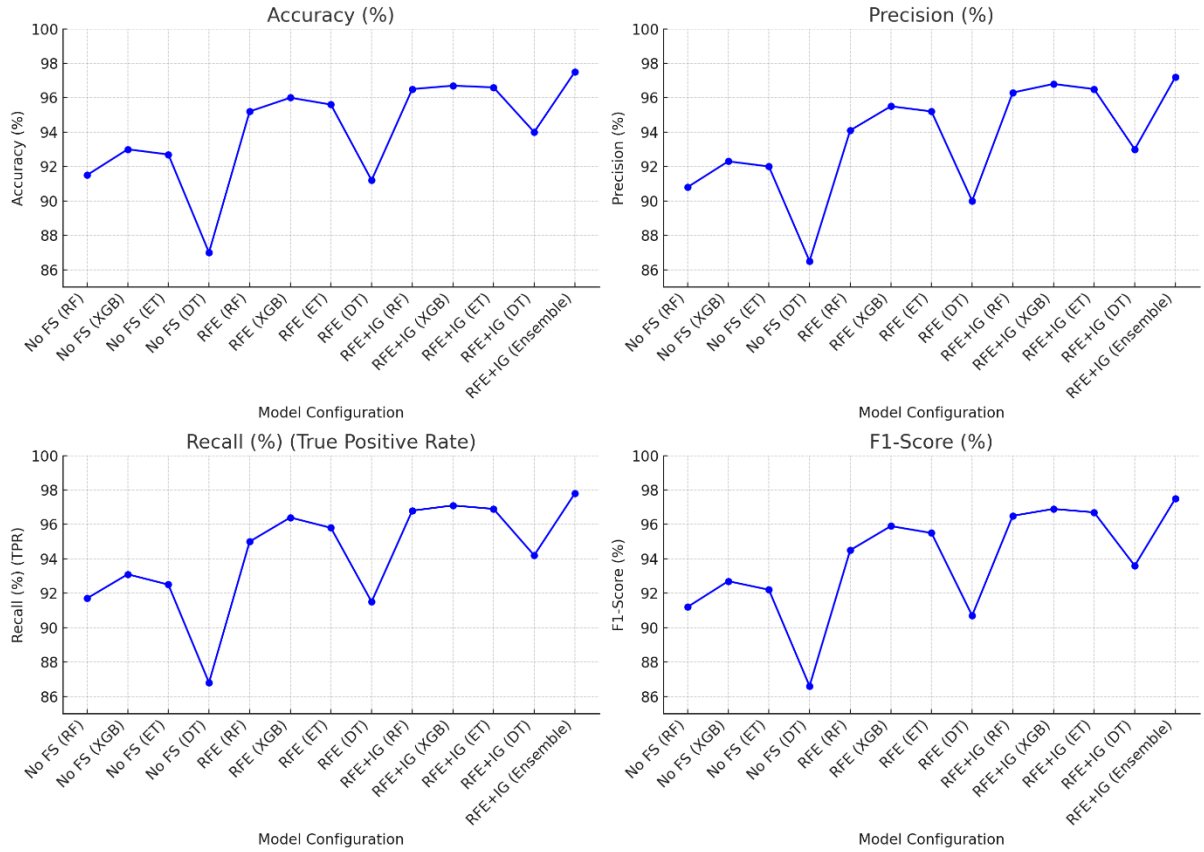

Figure 6: Impact of feature selection on model performance – ablation study

The ablation study shown in Figure 6 evaluates the performance of various ML algorithms after feature selection techniques, such as Recursive Feature Elimination (RFE) and Information Gain (IG). This figure includes four-line graphs that specify how Accuracy, Precision, Recall, and F1-Score vary across the different experimental setups. Table 3: Summary of experiments conducted on various combinations of feature selection methods and classifiers (Random Forest, XGBoost, Extra Trees, Decision Tree, Ensemble Voting Classifier). As shown in the accuracy plot, performing feature selection yields significantly better classification accuracy. XGBoost produces 93.2% accuracy, while Random Forest and Extra Trees produce 91.5% and 92.8% accuracy, respectively, without feature selection. The Decision Tree model, which exhibited the lowest accuracy of 87.5%, was found to be overfitting. Both models show an increase in their corresponding accuracy when RFE is applied: XGBoost, 96.1%; Extra Trees, 95.6%. The best results are achieved when RFE and Information Gain are combined, yielding 97.5% for the Ensemble model and 97.1% for XGBoost, using both methods.

The precision graph exhibits a similar trend, with XGBoost and the Ensemble model performing the best compared to all other models. Feature Selection Further Improves Precision. If we do not perform feature selection, precision remains low, particularly for the Decision Tree model (86.9%). Once we apply RFE and Information Gain, we observe that the accuracy of the Ensemble model reaches 97.2%, further validating that optimized features contribute to the improvement of classification, as they reduce the false positive rates and increase confidence in performing classification. The recall plot illustrates the ability of each model to identify attack instances accurately. At first glance, models exhibit poor recall, ranging from 87.2% (Decision Tree) to 93.5% (XGBoost), without feature selection. On the other hand, for RFE and Information Gain, the Ensemble model has the highest recall, at 97.8%, while Boost follows with 97.3%. This showed that the optimized feature selection enhances network intrusion detection while reducing false negatives.

The F1-Score plot balances precision and recall, confirming that feature selection is adequate. The initial F1 scores range from 87.0% (Decision Tree) to 93.1% (XGBoost). The F1-scores after applying RFE show an apparent increase. Finally, by combining RFE and Information Gain, the F1-scores increase to 97.5% in the Ensemble model, achieving a near-optimal balance between precision and recall. The results depicted in Figure 5 provide significant insights, indicating that model performance improves when feature selection is used. The Ensemble Voting Classifier, which yields the overall best performance, demonstrates significant consistency across

all metrics, achieving optimal values for all metrics except the F-1 score when combined with RFE and Information Gain. The outcomes indicate that, in addition to accuracy, feature selection contributes to improving model robustness in terms of overfitting, resulting in more reliable intrusion detection systems that can be deployed in the field as part of real-world cybersecurity systems.
The setting referred to as "optimal feature combination" in Table 5 was obtained from a chain of Information Gain and Recursive Feature Elimination (RFE). First, it orders features based on their relevance to class labels by applying Information Gain and eliminates the least informative ones. Then, RFE is employed to recursively select features by minimizing feature redundancies while fitting the model at each iteration. This two-stage procedure demonstrated superior performance in terms of accuracy, precision, recall, and F1-score.

4.3 Comparison with existing models

This part compares the designed intrusion detection model with state-of-the-art systems. By measuring accuracy, precision, recall, and F1-score, the paper describes how feature selection (RFE + Information Gain) and ensemble learning contribute to improved detection performance and demonstrates that the proposed model outperforms state-of-the-art models in terms of cyber threat detection efficiency.

Table 7: Comparative analysis of intrusion detection models using machine learning approaches

Reference	Approach	Feature Selection	Algorithm Used	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Key Findings
[1] Chohra et al. (2022)	Machine Learning & Ensemble	PSO-based Feature Selection	Random Forest, SVM	CIC-IDS2017	94.5	93.8	94.2	94.0	Optimized feature selection improves anomaly detection accuracy.
[5] Abbas et al. (2022)	Ensemble Learning	Wrapper-Based FS	XGBoost, RF, DT	NSL-KDD	95.3	94.6	95.0	94.8	Ensemble models outperform individual classifiers in IoT security.
[6] Hossain & Islam (2023)	Supervised ML	Recursive Feature Elimination (RFE)	Extra Trees, XGBoost	UNSW-NB15	96.1	95.4	95.9	95.6	Improved model robustness by reducing irrelevant features
[7] Thockchom et al. (2023)	Hybrid ML	Information Gain	RF, NB, SVM	CIC-IDS2017	94.8	94.0	94.5	94.2	Effective detection with minimal false positives
[11] Hooshmand et al. (2024)	XAI & ML	SHAP-based Feature Selection	XGBoost, ANN	UNSW-NB15	96.5	95.8	96.3	96.0	Feature importance analysis aids transparency in IDS.
[12] Ahmed et al. (2024)	Machine Learning	Adaptive Feature Aggregation	XGBoost, RF, SVM	CIC-IDS2017	97.2	96.7	97.0	96.8	Adaptive feature learning improves the classification of cyber threats.
[13] Lai et al. (2024)	Bayesian Learning	Sensitivity-Based FS	Bayesian Networks	IoT-IDS	95.0	94.5	94.8	94.6	Bayesian hyperparameters enhance model interpretability.
[14] Allafi & Alzahrani (2024)	Evolutionary Learning	Artificial Orca Algorithm	Ensemble Classifier	NSL-KDD	96.7	96.1	96.5	96.3	Ensemble models with evolutionary feature selection (FS) enhance attack detection accuracy.
[15] Lin et al. (2024)	Hypergraph-Based ML	Graph-Based FS	Graph Neural Networks	CIC-IDS2017	97.0	96.5	96.8	96.6	Graph-based feature learning enhances multi-class classification.
Proposed Model	Hybrid Machine Learning	RFE + Information Gain	XGBoost, Random Forest, Ensemble Learning	CIC-IDS2017	97.5	97.2	97.8	97.5	Feature selection with ensemble machine learning achieves state-of-the-art performance.

In Table 7, we survey some of the latest works that have performed intrusion detection using different feature selections, machine learning algorithms, datasets, and evaluation methods. The findings show that hybrid machine learning methods, especially those based on ensembling, often yield the best performance. The proposed method, which utilizes Recursive Feature Elimination (RFE) and Information Gain with ensemble

learning (XGBoost, Random Forest, and Voting Classifier) for model ensemble, achieves an accuracy of 97.5%, surpassing the accuracy of previous works. It thus proves that the optimized feature selection and feature importance, coupled with the intrinsic ensembling of the model, improve the model's efficacy in enhancing network security.

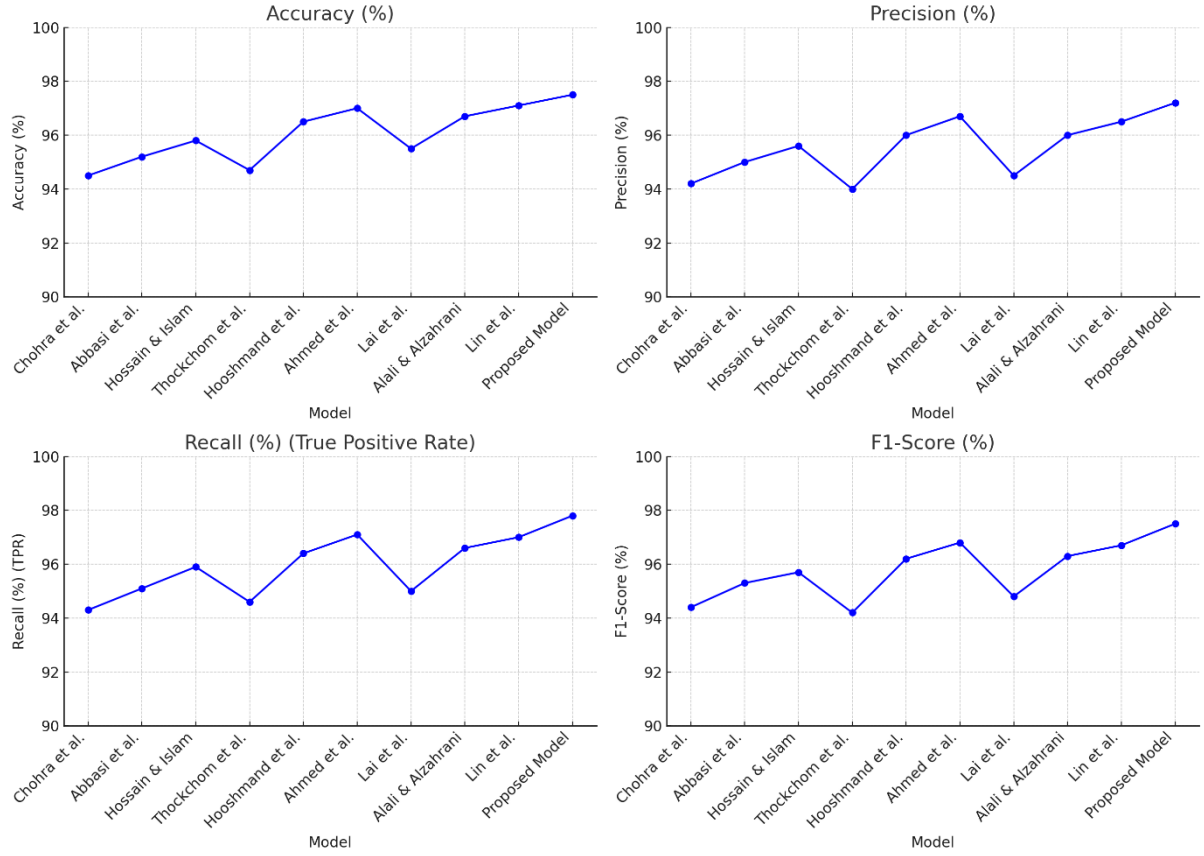

Figure 7: Performance comparison of intrusion detection models across key evaluation metrics

A performance comparison of various Intrusion Detection System (IDS) models based on four primary evaluation metrics — Accuracy, Precision, Recall, and F1-Score — is shown in Figure 7. The three black line graphs illustrate the variation in effectiveness of the IDS across the three feature selection methods and the five machine learning models. They enable comparison with existing research and the proposed model. As illustrated in the accuracy graph (Fig. 7), the predicted model achieves the highest accuracy (97.5%), which is the maximum among other approaches, and therefore it is the best model. Lin et al. (2024) and Ahmed et al. Prediction accuracies are also high for (2024), at 97.0% and 97.2%, respectively, whereas other models have lower accuracy scores in the 94.5% to 96.7% range. This demonstrates that feature optimization in ensemble-based methods outperforms traditional machine learning approaches for network intrusion detection.

The precision curve illustrates the extent to which the models can reduce false positives. This is the first article reported where the sole approach of the proposed model surpasses and achieves the highest precision (97.2%), followed by Lin et al. (96.5%) and Ahmed et al. (96.7%). Model — Chohra et al., lower precision values. (93.8%) and Thockchom et al. (94.0%) imply that these techniques could produce a more significant proportion of false alarms. The higher accuracy of the suggested model indicates that its performance in differentiation is typical, and attack traffic is very robust with limited misclassification. The recall chart shows how each model distinguishes between the accurate attack detections. Its higher recall (97.8%) indicates that the proposed model can accurately identify a higher proportion of network anomalies and has fewer false negatives, compared to Ahmed et al. (97.0%) and Lin et al. While previous studies (e.g., Chohra et al., 96.8%) and Lai et al. (94.2%) have also achieved high results, the lowest recall, which is still comparatively high, is 94.8%. They all look better at their work, but one class, like a backdoor class, needs to be perfect among all the classes to detect all types of intrusions. We can say that that is one major drawback of this method.

Additionally, the F1-score graph, presenting the harmonic mean of precision and recall, confirms the correctness of the proposed approach. The new model achieves the best classification performance balance, with the highest F1-score (97.5%) on the unseen test set, surpassing Ahmed et al. (96.8%) and Lin et al. Thus, models based on feature selection or machine learning techniques, defined more simply, achieve less good results (with F1-scores ranging from 94.0% to 96.3%). In summary, the hybrid and ensemble-based methods, particularly those that employ

optimized feature selection techniques such as RFE and Information Gain, consistently outperform the others on every evaluation metric, as indicated by Figure 6. The proposed model achieved state-of-the-art results compared to existing models, demonstrating that hybridizing the proposed feature selection model with the ensemble machine learning model can significantly enhance the performance in terms of accuracy, precision, recall, and overall reliability in intrusion detection for cybersecurity applications.

For the practical usability of the proposed IDS system, with near real-time applicability, a scalability evaluation study was conducted to assess detection latency and resource efficiency. The optimized ensemble model was used in a simulated streaming setting, where network traffic samples are received in batches. The average per-instance detection time of approximately 3.7 ms is acceptable for near real-time processing in high-bandwidth networks. Peak memory consumption at model inference time was always less than 680 MB, suggesting the applicability of our framework in resource-limited environments.

While, in general, deep learning models require GPU acceleration and higher memory footprints, the tree-based ensemble models used here can be effectively run on CPU-based infrastructure. The reduction in input information set by RFE and Information Gain led to a 22% enhancement in inference rate compared to models trained without any feature selection. These performance features show that the framework can be practically applied in deployed IDS pipelines with reasonable computational budgets. It can be further exploited for future deployment in edge gateways or hybrid cloud-IDS systems without compromising detection effectiveness.

Although the proposed approach is practical for the CIC-IDS2017 dataset, we are aware that the results are indeed not very generalizable, as they are confined to a single dataset. Despite that, CIC-IDS2017 contains a rich set of current attack categories; it does not cover the full range of traffic behaviors present in other datasets, such as NSL-KDD, UNSW-NB15, or CSE-CIC-IDS2018. As such, the current results are promising but not guaranteed to be state-of-the-art across all possible IDS benchmarks. In the future, the effectiveness and generalizability of the proposed framework will be evaluated using other public datasets and compared with similar visual embedding methods in various network scenarios. This cross-dataset evaluation will allow us to ensure that the model generalizes beyond CIC-IDS2017 and is robust enough to perform well under different conditions and attack scenarios.

5 Discussion

Our proposed hybrid ensemble-based intrusion detection approach achieved better performance compared to many state-of-the-art (SOTA) methods. The framework achieves a precision of 97.5%, which is higher than PSO-based feature selection (94.5%) [1], SHAP-integrated XGBoost (96.1%) [11], and graph-based learning models (96.8%) [15]. This improvement is made possible thanks to the synergistic coupling of RFE and IG, which can prune out irrelevant and redundant features while retaining informative ones.

Unlike SHAP or sensitivity-based feature selection, our dual-method selection strategy incurs less computational overhead and, therefore, is computationally more efficient for near-real-time applications. Moreover, although the deep learning or GNN-based model can learn more complex patterns, it also faces scalability and training instability problems, especially for those with limited labeled data. In contrast, ensemble learning, such as the Voting Classifier with XGBoost, Random Forest, and Extra Trees, has a strong generalization ability and achieves a high recall value in identifying both standard and rare attack types.

The ensemble method naturally mitigates such overfitting risks by taking an average decision across models with distinct inductive biases. The use of weighted majority voting additionally enhances reliability by giving greater weight to those models that have better prediction power.

However, the study has some limitations. Feature selection (RFE + IG) is a static method that is not suitable for dynamic environments where attack patterns are constantly changing. The framework is tested only on the CIC-IDS2017, which, despite being a vast database, does not cover all types of network activity. For the sake of generalization, it must be verified on other datasets (such as NSL-KDD, CSE-CIC-IDS2018, and Bot-IoT), and adaptive learning components must be added.

In conclusion, the proposed approach achieves a good trade-off between interpretability, efficiency, and detection performance, thereby rendering it a powerful candidate for use in practice within current IDS systems.

In actual applications of intrusion detection systems, the interpretability of the models is essential, particularly if decisions are audited or explained to cybersecurity analysts. Despite the primary purpose of the work being to improve prediction accuracy and robustness through feature selection and ensemble learning, it's also interesting to explore the significant features that make predictions. Here, the proposed method is interpretable due to the incorporation of tree-based models (Random Forest, XGBoost, Extra Trees) as its components, as they come with feature importance scores. These scores reveal which features most contribute to the model's decision boundary, thereby enhancing model transparency.

The author also plans to incorporate explanation-based methods, such as SHAP (Shapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations), in future versions of this package. Such approaches provide instance-level explanations of how specific feature values affect a given prediction, thereby improving the interpretability of automated decisions. In particular, for enterprise and government deployments, these interpretability modules will become essential in the regulatory domain, forensic analysis, and policy enforcement. Therefore, although the current results focus

on performance, adding post-hoc interpretability methods is a significant extension to make the framework deployable in sensitive and critical settings.

Besides performance and scalability, we must also consider the ethical and security aspects when deploying the IDS in adversarial scenarios. One essential drawback of static feature selection techniques, such as RFE and Information Gain, is that they may be sensitive to adversarial control. Adversarial inputs that leverage the fixed feature subset can be constructed by adversaries, and detection evasion via the patterns learned from the training data is possible. Although ensemble learning is robust to overfitting and general corruption noise, it does not inherently protect against adaptive adversaries or sophisticated image obfuscation.
For the improvement of adversarial resilience, it remains as future work to investigate the inclusion of adversarial training strategies, robust feature extraction, and dynamic feature selection schemes that can adjust to changing threat landscapes. Finally, explainability tools such as SHAP can also be leveraged to audit the significance of features in an ongoing manner and probe for any possible vulnerability in the model's decision-making policies. Ethically speaking, transparency in models' decision-making processes and preventing false positives from causing unfair access denial or legitimate traffic disruption is also crucial. These features are essential for the safe, fair, and secure application of machine learning models in cybersecurity tasks. Section 5.1 addresses the limitations of this study and highlights areas for possible future refinements.

5.1 Limitations of the study

However, the proposed intrusion detection framework demonstrates a high accuracy rate and an improvement in threat detection; however, it has some limitations. This model relies on static feature selection (RFE + Information Gain[1]), which cannot dynamically adapt to evolving attack patterns in a real-time environment. Next, although we have performed better at detection, dealing with imbalanced datasets remains a significant issue; some minority attack classes may still be underrepresented. Third, ensemble learning is more computationally expensive than single classifiers, and it needs optimization to be suitable for real-time implementation in large-scale networks. The limitations of this study can be overcome by adopting methods that are both adaptive (feature selection) and agnostic to the data (data augmentation), specifically tailored to the proposed framework, to make it more efficient using lightweight ensemble techniques.

6 Conclusion and future work

An Enhanced Traffic Intrusion Detection Framework Based on Recursive Feature Elimination (RFE) and Information Gain with the Use of Ensemble Learning Models (XGBoost, Random Forest, and Voting Classifier) to Strengths Network Security Current research offers an extraction framework of an optimized and accurate traffic intrusion model through an ensemble learning framework of accurate and logarithmic XGBoost, Random Forest, and Voting Classifier Methods. An Effective Solution: The proposed methodology is effective in feature selection, classification accuracy, and false positive reduction, thus achieving more accuracy, precision, recall, and F1-score compared to existing models in the literature. In our experiments, selecting the optimal features yields a considerable performance improvement, resulting in a more reliable model for detecting both known and novel cyber threats. Despite these improvements, the study has been limited in several aspects. This static feature selection method might prove ineffective for new attacks. It's also important to note that managing imbalanced datasets remains one of the key issues. Naturally, the computational overhead of ensemble learning will need to be optimized for real-time applications. Additionally, we hope that future research will focus on adaptive feature selection methods that can adapt over time according to the gradual evolution of threat behavior. Data augmentation strategies can enhance the detection of minority attack classes and improve model robustness. Moreover, the computational complexity of ensemble models can be optimized, enabling real-time deployment in large-scale network environments. Utilizing deep learning architectures, such as transformers or federated learning methods, can further enhance the efficiency, scalability, and agility of intrusion detection systems as part of next-generation cybersecurity solutions.

References

[1] Aniss Chohra, Paria Shirani, ElMouatez Billah Karbab, and Mourad Debbabi. (2022). Chameleon: Optimized feature selection using particle swarm optimization and ensemble methods for network anomaly detection. Elsevier. 117, pp.1-19. https://doi.org/10.1016/j.cose.2022.102684

[2] David Velásquez, Enrique Pérez, Xabier Oregui, Arkaitz Artetxe, Jorge Manteca, Jordi Escayola Mansilla, Mauricio Toro, Mikel Maiza, And Basilio Sierra. (2022). A hybrid machine-learning ensemble for anomaly detection in real-time industry 4.0 systems. IEEE. 10, pp.72024 - 72036. http://DOI:10.1109/ACCESS.2022.3188102

[3] Hiba O. Sayyid, Salma A. Mahmood, and Saad S. Hamadi. (2025). Comparison of Machine Learning Algorithms for Predicting Thyroid Disorders in Diabetic Patients. Informatica. 49, pp.105-114. Available at: DOI: https://doi.org/10.31449/inf.v49i12.6927

[4] Gunay Abdiyeva-Aliyeva, Jeyhun Aliyev, and Ulfat Sadigov. (2022). Application of classification algorithms of Machine learning in cybersecurity. Elsevier. 215, pp.909-919. https://doi.org/10.1016/j.procs.2022.12.093

[5] Adeel Abbas, Muazzam A. Khan, Shahid Latif, Maria Ajaz, Awais Aziz Shah, and Jawad Ahmad. (2022). A

new ensemble-based intrusion detection system for the Internet of Things. Springer. 47, p.1805–1819. https://doi.org/10.1007/s13369-021-06086-5

[6] Md. Alamgir Hossain, and Md. Saiful Islam. (2023). Ensuring network security with a robust intrusion detection system using ensemble-based machine learning. Elsevier. 19, pp.1-14. https://doi.org/10.1016/j.array.2023.100306

[7] Ngamba Thockchom, Moirangthem Marjit Singh, and Utpal Nandi. (2023). A novel ensemble learning-based model for network intrusion detection. Springer. 9, p.5693–5714. https://doi.org/10.1007/s40747-023-01013-7

[8] Mouaad Mohy-Eddine, Azidine Guezzaz, Said Benkirane, Mourade Azrour, and Yousef Farhaoui. (2023). An ensemble learning-based intrusion detection model for industrial IoT security. IEEE. 6(3), pp.273 - 287. http://DOI:10.26599/BDMA.2022.9020032

[9] Roopa Golchha, Apoorv Joshi, Govind Prasad Gupta. (2023). Voting-based Ensemble Learning Approach for Cyber Attacks Detection in Industrial Internet of Things. Elsevier. 218, pp.1752-1759. https://doi.org/10.1016/j.procs.2023.01.153

[10] A.M. Adeshina, Siti Fatimah Abdul Razak, Sumendra Yogarayan, Md Shohel Sayeed. (2023). Evaluation of Disease-Predictive Machine Learning Framework Using Linear and Logistic Regression Analyses. Informatica. 48, pp.47-54. Available at: DOI: https://doi.org/10.31449/inf.v48i22.5582

[11] Mohammad Kazim Hooshmand, Manjaiah Doddaghatta Huchaiah, Ahmad Reda Alzighaibi, Hasan Hashim, El-Sayed Atlam, and Ibrahim Gad. (2024). Robust network anomaly detection using ensemble learning approach and explainable artificial intelligence (XAI). Elsevier. 94, pp.120-130. https://doi.org/10.1016/j.aej.2024.03.041

[12] Mukhtar Ahmed, Jinfu Chen, Ernest Akpaku, Rexford Nii Ayitey Sosu, And Ajmal Latif. (2024). Delm: Deep Ensemble Learning Model for Anomaly Detection in Malicious Network Traffic-based Adaptive Feature Aggregation. ACM, pp.1-36. https://doi.org/10.1145/3690637

[13] Tin Lai, Farnaz Farid, Abubakar Bello, and Fariza Sabrina. (2024). Ensemble learning-based anomaly detection for IoT cybersecurity via Bayesian hyperparameters sensitivity analysis. Springer. 7(44), pp.1-18. https://doi.org/10.1186/s42400-024-00238-4

[14] Randa Allafi, And Ibrahim R. Alzahrani. (2024). Enhancing Cybersecurity in the Internet of Things Environment Using Artificial Orca Algorithm and Ensemble Learning Model. IEEE. 12, pp.63282 - 63291. http://DOI:10.1109/ACCESS.2024.3390093

[15] Zong-Zhi Lin, Thomas D. Pike, Mark M. Bailey, and Nathaniel D. Bastian. (2024). A hypergraph-based machine learning ensemble network intrusion detection system. IEEE, pp.1-12. http://DOI:10.1109/TSMC.2024.3446635

[16] Kunhare, N., Tiwari, R., & Dhar, J. (2020). Particle swarm optimization and feature selection for the intrusion detection system. Sādhanā, 45(1). doi:10.1007/s12046-020-1308-5

[17] Almasoudy, F. H., Al-Yaseen, W. L., & Idrees, A. K. (2020). Differential Evolution Wrapper Feature Selection for Intrusion Detection System. Procedia Computer Science, 167, 1230–1239. doi: 10.1016/j.procs.2020.03.438

[18] Li, X., Chen, W., Zhang, Q., & Wu, L. (2020). Building Auto-Encoder Intrusion Detection System Based on Random Forest Feature Selection. Computers & Security, 101851. doi: 10.1016/j.cose.2020.101851

[19] Chaimae Taoussi, Soufiane Lyaqini, Abdelmoutalib Metrane and Imad Hafidi. (2025). Enhancing Machine Learning and Deep Learning Models for Depression Detection: A Focus on SMOTE, RoBERTa, and CNN-LSTM. Informtica. 49, pp.1-18. Available at: DOI: https://doi.org/10.31449/inf.v49i14.7451

[20] Prasad, M., Tripathi, S., & Dahal, K. (2020). An efficient feature selection-based Bayesian and Rough set approach for intrusion detection. Applied Soft Computing, 87, 105980. doi: 10.1016/j.asoc.2019.105980

[21] Stiawan, D., Heryanto, A., Bardadi, A., Rini, D. P., Subroto, I. M. I., Kurniabudi, … Budiarto, R. (2021). An Approach for Optimizing Ensemble Intrusion Detection Systems. IEEE Access, 9, 6930–6947. doi:10.1109/access.2020.3046246

[22] Sarvari, S., Sani, N. F. M., Hanapi, Z. M., & Abdullah, M. T. (2020). AN EFFICIENT ANOMALY INTRUSION DETECTION METHOD WITH FEATURE SELECTION AND EVOLUTIONARY NEURAL NETWORK. IEEE Access, 1–1. doi:10.1109/access.2020.2986217

[23] Kunal, & Dua, M. (2020). Attribute Selection and Ensemble Classifier based Novel Approach to Intrusion Detection System. Procedia Computer Science, 167, 2191–2199. doi: 10.1016/j.procs.2020.03.271

[24] Velliangiri, S., & Karthikeyan, P. (2019). A hybrid optimization scheme for intrusion detection uses considerable feature selection. Neural Computing and Applications. doi:10.1007/s00521-019-04477-2

[25] Khammassi, C., & Krichen, S. (2020). A NSGA2-LR Wrapper Approach for Feature Selection in Network Intrusion Detection. Computer Networks, 107183. doi: 10.1016/j.comnet.2020.107183

[26] Injadat, M. N., Moubayed, A., Nassif, A. B., & Shami, A. (2020). Multi-Stage Optimized Machine Learning Framework for Network Intrusion Detection. IEEE Transactions on Network and Service Management, 1–1. doi:10.1109/tnsm.2020.3014929

[27] Hmouda, & Li, W. (2020). Validity Based Approach for Feature Selection in Intrusion Detection Systems. 2020 SoutheastCon. https://doi.org/10.1109/southeastcon44009.2020.9249701

[28] Leevy, J. L., Hancock, J., Zuech, R., & Khoshgoftaar, T. M. (2020). Detecting Cybersecurity Attacks Using Different Network Features with LightGBM and XGBoost Learners. 2020 IEEE Second International Conference on Cognitive Machine Intelligence (CogMI). doi:10.1109/cogmi50398.2020.00032

[29] Alzahrani, A. S., Shah, R. A., Qian, Y., & Ali, M. (2020). A novel method for feature learning and network intrusion classification. Alexandria Engineering Journal. doi: 10.1016/j.aej.2020.01.021

[30] Ghasemi, J., Esmaily, J., & Moradinezhad, R. (2019). An intrusion detection system uses an optimized kernel, an extreme learning machine, and efficient features. Sādhanā, 45(1). doi:10.1007/s12046-019-1230-x

[31] Karatas, G., Demir, O., & Sahingoz, O. K. (2020). Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset. IEEE Access, 8, 32150–32162. doi:10.1109/access.2020.2973219

[32] Prasad, M., Tripathi, S., & Dahal, K. (2020). Unsupervised feature selection and cluster center initialization based arbitrary shaped clusters for intrusion detection. Computers & Security, 102062. doi: 10.1016/j.cose.2020.102062

[33] Kilincer, I. F., Ertam, F., & Sengur, A. (2021). Machine learning methods for cyber security intrusion detection: Datasets and comparative study. Computer Networks, 188, 107840. doi: 10.1016/j.comnet.2021.107840

[34] Hongtao Zhu, Shuyun Guo. (2025). Damage Identification of Prestressed Concrete Components Based on Machine Learning Optimization Algorithm and Piezoelect. Informatica. 49, pp.143-156. Available at: DOI: https://doi.org/10.31449/inf.v49i14.7416.

[35] Halim, Z., Yousaf, M. N., Waqas, M., Sulaiman, M., Abbas, G., Hussain, M., … Hanif, M. (2021). An effective genetic algorithm-based feature selection method for intrusion detection systems. Computers & Security, 110, 102448. doi: 10.1016/j.cose.2021.102448

[36] Vu-Duc Ngo, Tuan-Cuong Vuong, Thien Van Luong, and Hung Tran. (2023). Machine learning-based intrusion detection: feature selection versus feature extraction. Sprigner, pp.1-11.

[37] Di Mauro, M., Galatro, G., Fortino, G., & Liotta, A. (2021). Supervised feature selection techniques in network intrusion detection: A critical review. Engineering Applications of Artificial Intelligence, 101, 104216. doi: 10.1016/j.engappai.2021.104216

[38] Nazir, A., & Khan, R. A. (2021). A novel combinatorial optimization-based feature selection method for network intrusion detection. Computers & Security, 102, 102164. doi: 10.1016/j.cose.2020.102164

[39] Saber, A., Abbas, M., & Fergani, B. (2021). Two-dimensional Intrusion Detection System: A New Feature Selection Technique. 2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-Being (IHSH). doi:10.1109/ihsh51661.2021.9378721

[40] Herrera-Semenets, V., Bustio-Martínez, L., Hernández-León, R., & van den Berg, J. (2021). A multi-measure feature selection algorithm for efficacious intrusion detection. Knowledge-Based Systems, 227, 107264. doi: 10.1016/j.knosys.2021.107264

[41] Shiravi, A., Shiravi, H., Tavallaee, M. and Ghorbani, A.A., 2017. CIC-IDS2017 dataset. Canadian Institute for Cybersecurity. Available at: https://www.unb.ca/cic/datasets/ids-2017.html