Comparison of Multiple Feature Selection Techniques for Machine Learning-Based Detection of IoT Attacks

Viet Anh Phan
243760@vut.cz
Brno University of Technology
Brno, Czech Republic

Jan Jerabek
jerabekj@vut.cz
Brno University of Technology
Brno, Czech Republic

Lukas Malina
malina@vut.cz
Brno University of Technology
Brno, Czech Republic

ABSTRACT

The Internet of Things (IoT) has become increasingly practical in applications such as smart homes, autonomous vehicles, and environmental monitoring. However, this rapid expansion has led to significant cybersecurity threats. Detecting these threats is critical, and while machine learning techniques are valuable, they struggle with high-dimensional data. Feature selection helps by reducing computational costs while maintaining model generalization. Selecting the most effective feature selection method is a crucial task. This research addresses this gap by testing five feature selection methods: Random Forest (RF), Recursive Feature Elimination (RFE), Logistic Regression (LR), XGBoost Regression (XGBoost), and Information Gain (IG) using the CIC-IoT 2023 dataset. It evaluates these methods when being used with five machine learning models: Decision Tree (DT), Random Forest (RF), k-Nearest Neighbors (k-NN), Gradient Boosting (GB), and Multi-layer Perceptron (MLP) using metrics like accuracy, precision, recall, and F1-score across three datasets. The results show that RFE, especially with the RF model, achieves the highest accuracy (99.57%) with 30 features. RF is the most stable, with accuracy from 83% to 99.56%. Additionally, the 5-feature scheme is best for implementing IDS on resource-limited IoT devices, with RFE paired with the k-NN model being the optimal combination.

CCS CONCEPTS

• **Security and privacy → Intrusion detection systems**.

KEYWORDS

IoT, Anomaly Detection, IDS, Machine Learning, Feature Selection

1 INTRODUCTION

IoT has emerged as a significant technology. Consequently, an increasing number of IoT devices are being deployed across various

public and private settings, gradually becoming essential elements of daily living. The research [33] has pointed out that there were around 15.14 billion IoT devices in 2023, and the number of connected devices will sharply increase to 29.42 billion in 2030.

As IoT applications and devices continue to expand rapidly, cybercriminals are likely to develop increasingly sophisticated methods to exploit new vulnerabilities. This is presenting an increasingly significant threat to security compared to previous levels. For instance, there are a range of Distributed Denial of Service (DDoS) and Denial of Service (DoS) attacks [18] that are associated with different layers (Perception, Transport and Application). Besides, the existence of Sybil attacks [35], Brute force attacks [11], Man-in-the-middle (MITM) attacks [16] and SQL injection attacks [24] also makes IoT devices vulnerable.

Because of these significant threats, there is no doubt that Intrusion Detection System (IDS) is a suitable solution to protect IoT networks [25]. And machine learning approaches have been utilized within the IDS as one of the most effective models, along with the network traffic datasets. Nevertheless, these datasets often include an abundance of irrelevant or duplicated attributes, leading to a negative effect on the complexity and precision of models. Therefore, feature reduction is a popular phase to reduce the dimensionality of data inserted into the machine learning model, which decreases the computational complexity and latency [21].

In terms of the feature reduction phase, feature selection and feature extraction are the most common methods to solve the problem of extensive datasets. Feature selection involves choosing a subset of the most informative features from the dataset while deleting the irrelevant ones. Meanwhile, feature extraction involves transforming the original features into a new set of features in a lower dimensional space. Each of them has its own advantages and disadvantages. However, the work [21] has proven that feature extraction only has higher accuracy and more potential to improve the performance when the number of features is not large (such as 9, up to 22). Thus, feature selection is the better mechanism when dealing with the actual dataset such as CIC-IoT 2023 (46 features) [27].

In this paper, the process of training the Machine learning model to detect attacks in IoT networks is divided into 4 stages: Data preprocessing, Feature selection, Training model, and Metrics evaluation. The classification of intrusive network traffic is performed by 5 machine learning models: DT, RF, k-NN, GB and MLP. The first 4 models require less computational power and training time compared to deep learning models. It makes them more suitable when being used in devices with limited resources. We still include MLP in our analysis because deep learning models offer distinct advantages, such as the ability to capture complex patterns in the

data, which might be missed by simpler models. Additionally, MLP is a simpler form of neural network that can be effectively used even in devices with resource constraints. Those various models are combined with 5 different feature selection techniques (RF, RFE, LR, XGBoost, and IG). The efficiency level of each technique is analysed based on several criteria across the benchmark dataset CIC-IoT 2023.

The background and related work are mentioned in Section 2. The proposed methodology is explained in Section 3. In section 4, the training results and comparison among different Feature selection techniques are analysed. Section 5 concludes the paper.

2 RELATED WORKS AND LIMITATIONS

This section explores current research on employing machine Learning for anomaly detection in the IoT networks. The focus is on machine learning models that leverage feature selection techniques to safeguard IoT environments.

The paper [17] in 2020 introduced a machine learning-based intrusion detection approach that applied XGBoost algorithm [8] to select appropriate features from the dataset named UNSW-NB15 [26]. Then several machine learning models were implemented with the use of reduced feature space: Support Vector Machine (SVM), k-NN, LR, Artificial Neural Network (ANN) and DT. The accuracy of the classifiers ranged from 88.13 to 90.85% for binary classification. However, the performance became less convincing in the case of multi-class classification: 53.95 - 77.51%. The low performance suggested potential difficulties in the practical applicability of their approach.

Injadat et al. in 2021 designed a framework [14] for detecting anomaly in IoT networks. They utilized the methods known as Correlation-based (CBFS) and Information Gain (IGBFS) to specify features for the dataset CIC-IDS 2017 [5] and UNSW-NB 2015. The data was then fed into several types of k-NN model. The proposed solution detected network attacks with extremely high accuracy (over 99% for both feature selection techniques). However, their feature selection methods were limited to a single category named as Filter methods. This narrow focus limited the generalizability of their findings.

The work [10] in 2022 proposed an approach to select features based on Gini Impurity-based Weighted Random Forest. The dataset was then applied with several machine learning techniques such as DT, AdaBoost, Gradient Boosting, MLP and Deep Neural Networks (LSTM, GRU). The best result achieved from the IDS was 93.01% for the dataset UNSW-NB15 and 99.90% for the dataset named TON-IoT [13]. The first limitation is that they used only one feature selection method. Besides, the reliance on complex deep learning models posed significant computational challenges, making it difficult to deploy in real-time and resource-constrained IoT environments.

In 2023, the possibility of employing IDS was proved in [15] by ResNet-GRU model. Lasso-based technique was harnessed as a feature selection method to find the optimal subset from the dataset IoT-23 [1]. 97% accuracy rate was achieved after conducting experiment. But the complexity of the ResNet-GRU model resulted in high computational demands. Moreover, the use of only one feature selection method (Lasso) limited the study's generalizability

and might not capture the most relevant features across diverse datasets.

In 2024, the research [21] conducted the experiment for detecting and classifying IoT attacks in the dataset TON-IoT with the use of the feature selection technique called Pearson's correlation coefficient (PCC). After reducing unnecessary features, the training phase was performed by machine learning models: DT, RF, k-NN, Naive Bayes (NB), and MLP. The result showed that the highest achieved accuracy was 77.25% for multi-class classification. The main drawback of this research was the low accuracy in detecting attacks.

A review of the significant research papers mentioned above reveals their results and limitations. The datasets used in these studies are also outdated and may not accurately reflect the current status of IoT cyberattacks. To address these issues, this paper utilizes five different feature selection methods and employs the most recent IoT attack dataset, CIC-IoT 2023, which has not been used in previous works. By using multiple feature selection methods in 3 categories (filter, wrapper, and embedded), we aim to provide a broader and more robust evaluation. Additionally, we compare the effectiveness of these methods to offer a comprehensive analysis and improve the generalizability of our findings.

3 RESEARCH METHOD

This section introduces the dataset used for detecting attacks in IoT networks and outlines our proposed approach. Firstly, we choose the available dataset named CIC-IoT 2023. The entire dataset is then preprocessed in the Preprocessing phase to normalize and scale the data, and to split the data into two sets for training and testing. After this operation, the Feature Selection process is performed to find the meaningful subsets of the dataset. Finally, we leverage machine learning techniques to identify multiple types of attack and provide the results based on several evaluation criteria. The designed approach is illustrated in Fig. 1.
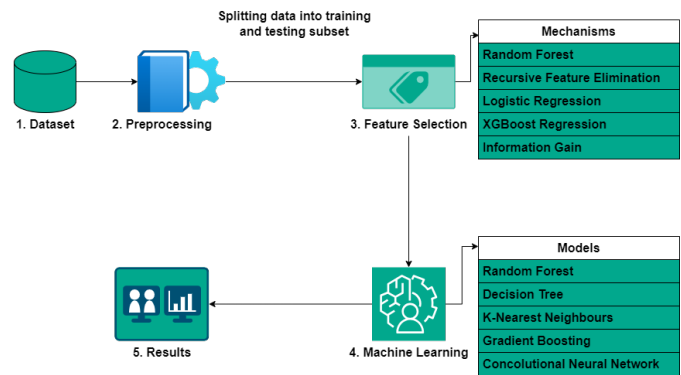

Figure 1: The overview of proposed methodology

3.1 Dataset

The CIC-IoT 2023 dataset includes various modern IoT attacks. Specifically, there are 33 types of IoT attack. However, in order to

make classification more effective in our work, we group these individual attacks into 7 categories: DDoS, DoS, Brute Force, Spoofing [3], Recon [4], Web-based [32] and Mirai [28].

The first category is named as DDoS which floods networks or devices with excessive traffic, causing disruptions and making services unavailable. The second category (DoS) is similar to DDoS but typically from a single source, also aims to disrupt service availability. Brute Force attacks attempt multiple password combinations to gain unauthorized access. Spoofing deceives devices by pretending to be legitimate ones, leading to data theft or malware distribution. Recon attacks gather network information to identify vulnerabilities. Web-based attacks exploit web application weaknesses for unauthorized access. Mirai botnet attacks compromise IoT devices, turning them into bots for large-scale DDoS attacks. This dataset provides a comprehensive overview of these current IoT threats.

Besides, the dataset also contains normal samples (benign). An overview of the gathered samples from dataset CIC-IoT 2023 after grouping is presented in Fig. 2 in an aggregated form.
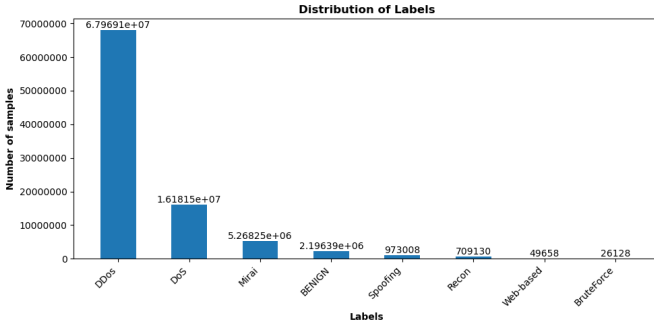

Figure 2: Number of attacks in defined categories and benign samples in the dataset CIC-IoT 2023

Fig. 2 highlights a significant imbalance among the attack categories and benign samples. Specifically, the DDoS category has the highest number of samples, followed by DoS and Mirai attacks. Other categories such as Spoofing, Recon, Web-based, and Brute Force have considerably fewer samples. However, we do not apply balancing methods to this imbalanced dataset because of maintaining the natural and real-time distribution of attack types performed by [27]. Moreover, utilized machine learning models such as RF and GB can handle imbalanced datasets effectively.

3.2 Preprocessing

Scaling is the process of adjusting the range of feature values in a dataset to a common scale, typically through methods such as normalization or standardization. This is important because many machine learning algorithms perform better when features are on a similar scale. In this case, the dataset has numerical features with different scales. For instance, the 'IAT' feature spans from a minimum of 0.0 to a maximum of approximately 167,639,426.32, indicating a vast range. Similarly, 'Covariance' also shows a broad scale, ranging from 0.0 to around 137,284,386.26. In contrast, 'Header_Length' has a smaller range, from 0.0 to 9,815,555.0, while 'Rate' and 'Srate'

both range from 0.0 to 7,340,032.0. That is the reason why scaling is applied in this work.

The first scaling method we applied is called standard scaling [2], which transforms each feature by subtracting the mean and dividing by the standard deviation. This results in features with a zero mean and a unit variance, leading to the following effect:

- Reduced sensitivity to feature scale: Machine learning algorithms can become heavily influenced by features with larger scales, potentially leading to biased model behavior. Standard scaling mitigates this issue by placing all features on a common scale.
- Improved convergence: Standardized features can aid in faster convergence during the optimization process (e.g. using gradient descent).

After implementing the first scaling method, another method called Min-Max scaling [7] (also known as normalization) is performed. This technique transforms each feature to a predefined range, typically between 0 and 1. When features are normalized, the calculations become more efficient. So this method helps reduce the computational complexity and time required for model training.

Lastly, the categorical labels are converted to number to go through the Feature Selection procedure. Specifically, the categories are marked by the following numbers: BENIGN - 0; DDoS - 1; DoS - 2; Mirai - 3; Spoofing - 4; Recon - 5; Web-based - 6; BruteForce - 7. Then, the dataset is divided into training and testing parts with the rate 0.7 and 0.3, respectively. This rate is widely used and allows for easier comparison of results across different studies and projects.

3.3 Feature Selection

This paper compares 5 different popular feature selection techniques, which belong to three main categories:

- Filter methods: Information Gain
- Wrapper methods: Recursive Feature Elimination
- Embedded methods: Random Forest, XGBoost, Logistic Regression

After getting the result from 5 different methods, we will compare the performance of each technique based on 3 schemes: 5, 15 and 30 selected features out of 46 features. The features are chosen from the highest score to the lowest within the predefined range. The case of no feature removal is also conducted to serve as a basis for comparison for all methods (referred to as "Not applied" in those tables presented later).

First method, IG is a statistical measure used in machine learning for feature selection. It assesses the reduction in uncertainty (entropy) about the target variable after considering a particular feature. Features with higher information gain are considered more relevant for predicting the target variable. The formula to compute information gain [30] of the feature A from total set and subset's entropy is described below.

$$IG(A) = \text{Entropy}(S) - \sum_{v \in vals(A)} \frac{|(S_v)|}{|(S)|} \cdot \text{Entropy}(S_v). \quad (1)$$

Here, IG(A) is the information gain value of feature A, with S being the total set. S_v represents the subset including all entries

with specific value v of the feature A. $|S|$ is are the total number of instances and $|S_v|$ is the number of instances having the value v.
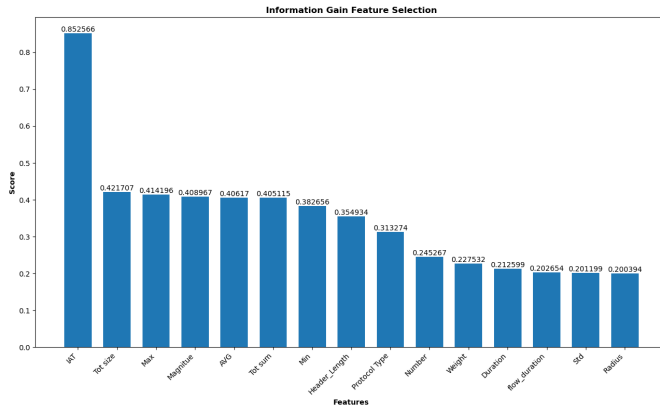

Figure 3: Graph of feature importance using IG (shortened graph with only 15 most relevant features)

This technique is considered to be fast and simple since calculating information gain is computationally inexpensive, making it suitable for large datasets like CIC-IoT 2023. It doesn't require complex model training, allowing for quick feature selection [22]. However, IG can favor features with a larger number of distinct values. These features can potentially split the data into more categories, leading to a higher reduction in entropy. This might not always reflect the true predictive power of the feature, especially if the additional categories don't provide much meaningful information. The obtained impact score of each feature in the descending order of chosen dataset is shown in Fig. 3. And the chosen features for each scheme are displayed in Tab. 1.

Table 1: Resulting features to be selected in 3 schemes using IG

	Description
5 selected features	'IAT', 'Tot size', 'Max', 'Magnitue', 'AVG'
15 selected features	'IAT', 'Tot size', 'Max', 'Magnitue', 'AVG',
	'Tot sum', 'Min', 'Header_Length', 'Protocol Type', 'Number',
	'Weight', 'Duration', 'flow_duration', 'Std', 'Radius'
30 selected features	'IAT', 'Tot size', 'Max', 'Magnitue', 'AVG',
	'Tot sum', 'Min', 'Header_Length', 'Protocol Type', 'Number',
	'Weight', 'Duration', 'flow_duration', 'Std', 'Radius',
	'Covariance', 'rst_count', 'Variance', 'urg_count', 'Rate',
	'Srate', 'TCP', 'syn_count', 'ack_flag_number', 'IPv',
	'LLC', 'ICMP', 'HTTPS', 'ack_count', 'fin_count'

The second model, RFE is a feature selection technique used in machine learning to identify a subset of relevant features for building a model. It works by iteratively removing features considered least important and refitting the model with the remaining features. In practice, RFE is used in combination with an estimator to provide feature importance measures. This information is crucial for RFE to decide which feature to remove. In this work, DT is chosen to be the estimator of RFE because of its proven high efficiency [23].

This technique is not computationally efficient for the feature selection, especially for large datasets with many features. But this method can assess how well features work together to provide a comprehensive picture for prediction. Thus it can enhance the model accuracy. The result after utilizing RFE for choosing 5, 15, and 30 features is represented in Tab. 2.

Table 2: Resulting features to be selected in 3 schemes using RFE

	Description
5 selected features	'flow_duration', 'Header_Length', 'Srate', 'rst_count', 'IAT'
15 selected features	'flow_duration', 'Header_Length', 'Protocol Type', 'Duration', 'Srate',
	'syn_count', 'fin_count', 'urg_count', 'rst_count', 'HTTPS',
	'Min', 'Tot size', 'IAT', 'Covariance', 'Weight'
30 selected features	'flow_duration', 'Header_Length', 'Protocol Type', 'Duration', 'Rate',
	'Srate', 'syn_flag_number', 'psh_flag_number', 'ack_count', 'syn_count',
	'fin_count', 'urg_count', 'rst_count', 'HTTP', 'HTTPS',
	'SSH', 'UDP', 'Tot sum', 'Min', 'Max',
	'AVG', 'Std', 'Tot size', 'IAT', 'Number',
	'Magnitue', 'Radius', 'Covariance', 'Variance', 'Weight'

The third technique, RF is an ensemble method consisting of multiple decision trees (usually hundreds or thousands) [34]. At each node of a decision tree, the model considers the chosen random subset of features. The algorithm selects the feature that leads to the best separation of data points based on the target variable (regression or classification). This separation is often measured by a metric like Gini impurity (classification) or variance reduction (regression).

Therefore, RF can be used to calculate the feature importance. Features that consistently lead to good separations (reduction in impurity/variance) across multiple splits in different trees are considered more important for predicting the target variable. The sum of the importance scores is 1, which can be seen in Fig. 4 together with the descending order of importance level for each feature. After applying RF, the result of selecting features is achieved in Tab. 3.

The fourth technique, LR is considered to be a popular algorithm for feature selection because of its simplicity and efficiency to identify the most meaningful features. After training the LR
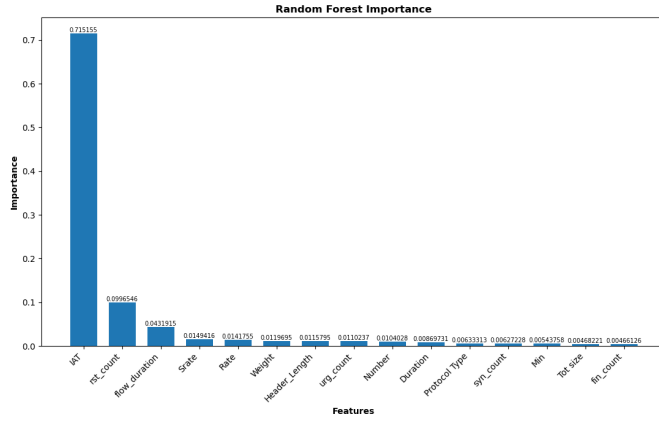
Figure 4: Graph of feature importance using RF (shortened graph with only 15 most relevant features)

Table 3: Resulting features to be selected in 3 schemes using RF

	Description
5 selected features	'IAT', 'rst_count', 'flow_duration', 'Srate', 'Rate'
15 selected features	'IAT', 'rst_count', 'flow_duration', 'Srate', 'Rate',
	'Weight', 'Header_Length', 'urg_count', 'Number', 'Duration',
	'Protocol Type', 'syn_count', 'Min', 'Tot size', 'fin_count'
30 selected features	'IAT', 'rst_count', 'flow_duration', 'Srate', 'Rate',
	'Weight', 'Header_Length', 'urg_count', 'Number', 'Duration',
	'Protocol Type', 'syn_count', 'Min', 'Tot size', 'fin_count',
	'Tot sum', 'HTTPS', 'Max', 'ack_count', 'Covariance',
	'Magnitue', 'AVG', 'Std', 'Radius', 'HTTP',
	'Variance', 'SSH', 'syn_flag_number', 'TCP', 'ack_flag_number'

Figure 5: Graph of feature importance using LR (shortened graph with only 15 most relevant features)

Table 4: Resulting features to be selected in 3 schemes using LR

	Description
5 selected features	'Header_Length', 'Magnitue', 'Variance', 'Duration', 'Min'
15 selected features	'Header_Length', 'Magnitue', 'Variance', 'Duration', 'Min',
	'Protocol Type', 'Tot sum', 'rst_count', 'TCP', 'AVG',
	'syn_flag_number', 'fin_flag_number', 'ICMP', 'ack_flag_number', 'psh_flag_number'
30 selected features	'Header_Length', 'Magnitue', 'Variance', 'Duration', 'Min',
	'Protocol Type', 'Tot sum', 'rst_count', 'TCP', 'AVG',
	'syn_flag_number', 'fin_flag_number', 'ICMP', 'ack_flag_number', 'psh_flag_number',
	'IPv', 'LLC', 'UDP', 'urg_count', 'HTTP',
	'syn_count', 'rst_flag_number', 'Tot size', 'Radius', 'Std',
	'HTTPS', 'Max', 'IAT', 'ack_count', 'Weight'

model, features with larger absolute coefficient values (positive or negative) are generally considered more important for predicting the target variable [19].

This technique is performed on our training dataset and provides the results as shown in Fig. 5. The selected features are mentioned in Tab. 4 for 3 situations.

The last method applied for feature selection in our paper is called XGBoost, which works by building an ensemble of decision trees, similar to RF. Each tree is trained on a subset of data points (bootstrapping) and a random subset of features [9].

But unlike RF, XGBoost calculates feature importance scores (Gain, Cover or Weight) during the training process. So, XGBoost importance scores directly reflect how much a feature contributes to improving the model's performance across the entire ensemble,
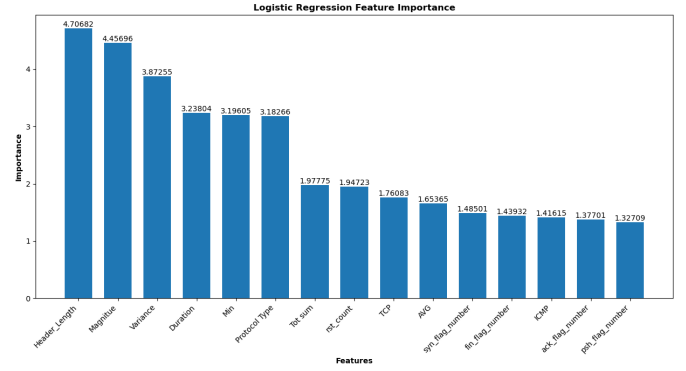
potentially providing a more accurate picture of feature importance compared to other methods.

The features' importance weights for the whole training dataset is illustrated in Fig. 6. And the selected features when using this technique is shown in Tab. 5.

As can be seen from the graphs and tables above, across all selected feature sets (5, 15, and 30 features), certain features consistently appear in the selected sets, such as 'IAT', 'rst_count', 'flow_duration', 'Srate', 'Rate', 'Header_Length', 'Magnitue', 'Variance', 'Duration', and 'Min'. This consistency suggests these features may hold high predictive power or informational value across different methods. All methods except for LR share the same result about feature 'IAT', which is placed in the top 5 most important features among them with a high score. However, no techniques
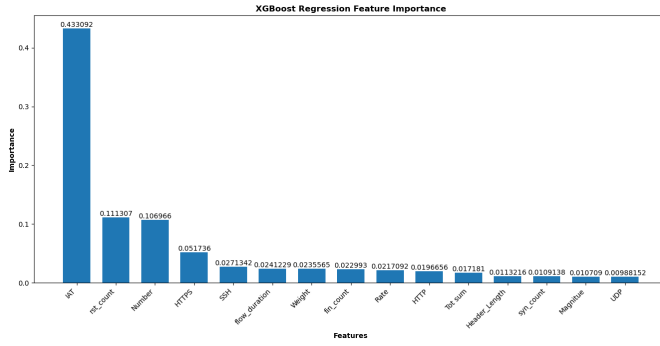
Figure 6: Graph of feature importance using XGBoost (shortened graph with only 15 most relevant features)

Table 5: Resulting features to be selected in 3 schemes using XGBoost

	Description
5 selected features	'IAT', 'rst_count', 'Number', 'HTTPS', 'SSH'
15 selected features	'IAT', 'rst_count', 'Number', 'HTTPS', 'SSH',
	'flow_duration', 'Weight', 'fin_count', 'Rate', 'HTTP',
	'Tot sum', 'Header_Length', 'syn_count', 'Magnitue', 'UDP'
30 selected features	'IAT', 'rst_count', 'Number', 'HTTPS', 'SSH',
	'flow_duration', 'Weight', 'fin_count', 'Rate', 'HTTP',
	'Tot sum', 'Header_Length', 'syn_count', 'Magnitue', 'UDP',
	'Min', 'urg_count', 'TCP', 'Protocol Type', 'Duration',
	'ack_count', 'DNS', 'Tot size', 'Variance', 'AVG',
	'Max', 'ack_flag_number', 'Covariance', 'Std', 'Radius'

have the same results. This is evident from the results that lie in their approaches to identifying relevant features.

3.4 Machine learning model

In this paper, we use the machine learning models: DT, RF, k-NN, GB, and MLP. DT is intuitive and easy to interpret, suitable for understanding decision-making processes. RF combines multiple decision trees to improve predictive accuracy and handles complex datasets. kNN relies on local similarity, making it effective for classification tasks with localized patterns. GB sequentially builds models to enhance predictive performance, particularly useful for tackling complex relationships in the data. MLP, as the neural network, excels at capturing complex patterns in large-scale datasets, making them a good tool for tasks requiring high-dimensional and non-linear data representations.

Overall, these classifiers provide flexibility and performance across a wide range of machine learning applications. The general description about these models is shown below.

Described as a first one, DT is a type of supervised learning algorithm used for both classification and regression tasks. It works by recursively partitioning the feature space into smaller regions, making decisions based on the values of input features [31]. At each node of the tree, the algorithm selects the feature and the corresponding threshold that best splits the data (with the use of measures like Gini impurity or information gain), aiming to maximize the homogeneity of the target variable within each subset. This process continues until a stopping criterion is met, such as reaching a maximum tree depth or when further splitting does not improve predictive performance. Finally, the tree assigns a class label (in classification) or a numerical value (in regression) to each leaf node, allowing for straightforward interpretation and prediction.

Another model, RF, is an algorithm that operates by constructing multiple decision trees during training and outputting the mode of the classes (classification) or the mean prediction (regression) of the individual trees [6]. Each tree is built from a bootstrap sample of the training data, and at each node, a random subset of features is considered for splitting. This randomness helps to decorrelate the trees, reducing overfitting and improving generalization performance. The final prediction is determined by aggregating the predictions of all trees. Advantages of RF include its robustness to overfitting, ability to handle large datasets with high dimensionality, and automatic feature selection.

The third model, k-NN, is a simple yet effective algorithm for both classification and regression tasks. It works by storing all available cases and classifying new cases based on a similarity measure (e.g., Euclidean distance) to the k nearest neighbors in the training set [20]. The predicted class or value is determined by a majority vote (for classification) or averaging (for regression) among the k nearest neighbors. Advantages of k-NN include its simplicity, flexibility to handle multi-class problems, and no assumptions about the underlying data distribution.

As the fourth, GB is an ensemble learning technique that combines multiple weak learners, typically decision trees, to create a strong predictive model. It works by sequentially adding new models to correct the errors made by the existing models. Each new model is trained on the residuals (the difference between the actual and predicted values) of the previous model [12]. The final prediction is the weighted sum of the predictions from all models.

Lastly, we introduce MLP is a type of artificial neural network that consists of multiple layers of nodes (neurons), including an input layer, one or more hidden layers, and an output layer. Each neuron in one layer is connected to every neuron in the next layer [36]. During training, the network adjusts the weights of connections between neurons using optimization algorithms like gradient descent to minimize the difference between predicted outputs and actual targets.

In our work, the package Scikit-learn [29] is utilized to implement these machine learning models with the following hyperparameters in Tab. 6.

4 RESULTS

When evaluating the results of each feature selection technique, as well as the performance of different machine learning models, we

Table 6: Hyper-parameters in machine learning models

ML models	Hyper-parameters
DT	criterion: 'gini', splitter: 'best', max_depth: None, random_state: 42
RF	criterion: 'gini', n_estimators: 100, max_depth: None, random_state: 42
k-NN	n_neighbors: 5, weights: 'uniform', algorithm: 'auto'
GB	loss: 'log_loss', learning_rate: 0.1, n_estimators: 100, random_state: 42
MLP	hidden_layer_sizes: (100, 50), solver: 'lbfgs', random_state: 42

apply the most popular indicators: accuracy, precision, recall and F1-score. The formulas to compute these metrics are shown below:

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN}, \quad (2)$$

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (3)$$

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (4)$$

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (5)$$

Where TP (True Positive) represents the number of instances that are correctly classified as positive by the model, TN (True Negative) for the case of classifying as negative by the model. While, TP and TN indicate correct predictions, FP (False Positive) and FN (False Negative) indicate incorrect predictions.

After training the dataset and testing with the evaluation metrics above, the results were processed. Tab. 7 summarizes the results for the case of 5 selected features, Tab. 8 for 15 selected features and Tab. 9 for 30 selected features. The highest outcomes from feature selection methods for each classifier are indicated by the highlighted values, which are bold and red. "Not applied" means the case when not applying feature selection techniques (keeping all 46 features). This provides the base insight for comparison.

From Tab. 7, RFE consistently outperforms other techniques across various models. For the DT, RF, k-NN, and GB models, RFE achieves the highest accuracy scores, such as 99.31% for DT and 99.45% for RF. Moreover, the use of RFE with RF model achieves higher score compared to not applying any feature selection. This implies that RFE can enhance model performance, indicating its efficiency in selecting the most relevant features. Conversely, LR selection tends to yield the lowest performance, particularly for the DT and GB models, with accuracies dropping to 71.58% and 53.48%, respectively.

With 15 features in Tab. 8, RFE continues to excel, leading in the DT and GB models with accuracies of 99.30% and 99.48%, respectively. It also tops the RF model at 99.52%, with higher score than not applying feature selection techniques. For the k-NN model, IG produces the highest accuracy at 96.72%, slightly surpassing RFE. LR remains the weakest link, particularly evident in the GB model's performance, which stays at a modest 80.31%.

When the number of features increases to 30 (Tab. 9), RFE maintains its top position, especially in the DT model, with a notable 99.38% accuracy. In the RF and GB models, RFE, RF, and XGBoost feature selections all achieve close to the highest accuracies of 99.57%. For k-NN, LR selection provides the best results at 94.29%,

a significant improvement over fewer features. Meanwhile, IG and XGBoost show lower performance compared to RFE in most cases.

Across 3 different schemes, RFE yields the highest performance, particularly excelling in RF model. On the contrary, LR frequently results in the lowest performance, especially in scenarios with fewer features. Although IG and XGBoost perform moderately, they do not consistently match the efficacy of RFE. Overall, RFE stands out as the most accurate feature selection technique.

Table 7: Results when selecting 5 features

DT		Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
	IG	98.68	98.76	98.68	98.71
	RFE	**99.31**	**99.32**	**99.31**	**99.31**
	RF	99.22	99.23	99.22	99.22
	LR	71.58	77.98	71.58	73.83
	XGBoost	98.79	98.70	98.79	98.73
	Not applied	98.96	98.88	98.96	98.90
RF		**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-score (%)**
	IG	98.96	99.00	98.96	98.96
	RFE	**99.45**	**99.45**	**99.45**	**99.44**
	RF	99.37	99.37	99.37	99.37
	LR	80.85	80.20	80.85	80.46
	XGBoost	98.85	98.78	98.85	98.80
	Not applied	99.48	99.48	99.48	99.45
k-NN		**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-score (%)**
	IG	89.72	88.67	89.72	87.01
	RFE	**98.86**	**98.82**	**98.86**	**98.80**
	RF	98.82	98.77	98.82	98.76
	LR	54.17	73.09	54.17	58.54
	XGBoost	98.42	98.29	98.42	98.24
	Not applied	92.13	91.92	92.13	91.77
GB		**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-score (%)**
	IG	98.95	99.03	98.95	98.91
	RFE	**99.43**	**99.43**	**99.43**	**99.42**
	RF	99.31	99.31	99.31	99.29
	LR	53.48	74.22	53.48	57.74
	XGBoost	98.79	98.99	98.79	98.77
	Not applied	99.04	99.08	99.04	99.05
MLP		**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-score (%)**
	IG	82.05	80.57	82.05	76.12
	RFE	76.47	62.50	76.47	67.37
	RF	**88.85**	**89.54**	**88.85**	**88.88**
	LR	82.03	81.34	82.03	76.58
	XGBoost	75.13	59.23	75.13	64.84
	Not applied	83.28	81.64	83.28	79.53

In our previous evaluation, we focus on identifying the feature selection technique that can help achieving the highest accuracy with a specific machine learning model. However, it is also important to consider the stability and compatibility of these techniques across various models. The following graphs (Fig. 7) provide insights into their robustness and ability to maintain high performance consistently.

With only 5 features (as shown in Fig. 7 (a)), RF demonstrates the most stable and quite high performance. It leads with the highest cumulative accuracy of 485.57, showcasing strong results across all models, especially excelling with RF (99.37%) and DT (99.22%). This consistency across different model types makes RF a robust choice for scenarios with limited features. On the other hand, LR emerges as the weakest, providing the lowest cumulative accuracy of 342.11.

When the feature count increases to 15 (through Fig. 7 (b)), IG stands out as the most compatible technique, delivering the highest cumulative accuracy of 477.44. It maintains high performance across various models, particularly in GB (99.29%) and RF (99.42%). While IG outperforms RF by a small margin, RF still delivers excellent

Table 8: Results when selecting 15 features

DT		Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
	IG	99.00	99.09	99.00	99.04
	RFE	**99.30**	**99.33**	**99.30**	**99.31**
	RF	98.95	98.88	98.95	98.90
	LR	82.22	81.97	82.22	82.09
	XGBoost	98.83	98.78	98.83	98.79
	Not applied	98.96	98.88	98.96	98.90
RF		Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
	IG	99.42	99.42	99.42	99.40
	RFE	**99.52**	**99.52**	**99.52**	**99.50**
	RF	99.51	99.51	99.51	99.49
	LR	80.35	82.63	80.35	81.17
	XGBoost	99.49	99.49	99.49	99.47
	Not applied	99.48	99.48	99.48	99.45
k-NN		Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
	IG	**96.72**	**96.69**	**96.72**	**96.66**
	RFE	93.01	92.92	93.01	92.92
	RF	93.13	93.05	93.13	93.04
	LR	73.25	76.58	73.25	74.57
	XGBoost	93.97	93.95	93.97	93.92
	Not applied	92.13	91.92	92.13	91.77
GB		Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
	IG	99.29	99.34	99.29	99.30
	RFE	**99.48**	**99.43**	**99.43**	**99.42**
	RF	99.38	99.43	99.38	99.38
	LR	80.31	79.97	80.31	80.06
	XGBoost	99.36	99.38	99.36	99.36
	Not applied	99.04	99.08	99.04	99.05
MLP		Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
	IG	83.01	**83.54**	83.01	77.88
	RFE	81.19	77.58	81.19	74.17
	RF	80.89	76.58	80.89	73.57
	LR	**83.02**	82.70	**83.02**	**78.42**
	XGBoost	82.44	82.65	82.44	77.16
	Not applied	83.28	81.64	83.28	79.53

Table 9: Results when selecting 30 features

| DT		Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
	IG	98.95	98.87	98.95	98.89
	RFE	**99.38**	**99.39**	**99.38**	**99.38**
	RF	98.98	98.91	98.98	98.92
	LR	99.31	99.33	99.31	99.32
	XGBoost	98.98	98.90	98.98	98.92
	Not applied	98.96	98.88	98.96	98.90
RF		Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
	IG	99.56	99.57	99.56	99.54
	RFE	**99.57**	**99.57**	**99.57**	**99.55**
	RF	99.56	99.56	99.56	99.54
	LR	99.49	99.50	99.49	99.47
	XGBoost	**99.57**	**99.57**	**99.57**	**99.55**
	Not applied	99.48	99.48	99.48	99.45
k-NN		Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
	IG	93.32	93.17	93.32	93.20
	RFE	93.62	93.47	93.62	93.47
	RF	93.45	93.31	93.45	93.31
	LR	**94.29**	**94.21**	**94.29**	**94.22**
	XGBoost	93.49	93.34	93.49	93.35
	Not applied	92.13	91.92	92.13	91.77
GB		Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
	IG	99.27	99.36	99.27	99.25
	RFE	99.18	99.17	99.18	99.16
	RF	99.13	99.08	99.13	99.07
	LR	**99.39**	**99.39**	**99.39**	**99.37**
	XGBoost	99.13	99.08	99.13	99.09
	Not applied	99.04	99.08	99.04	99.05
MLP		Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
	IG	82.47	80.58	82.47	78.40
	RFE	83.05	80.89	83.05	79.71
	RF	83.00	80.85	83.00	**79.77**
	LR	**83.12**	**82.02**	**83.12**	79.05
	XGBoost	82.72	80.39	82.72	79.30
	Not applied	83.28	81.64	83.28	79.53
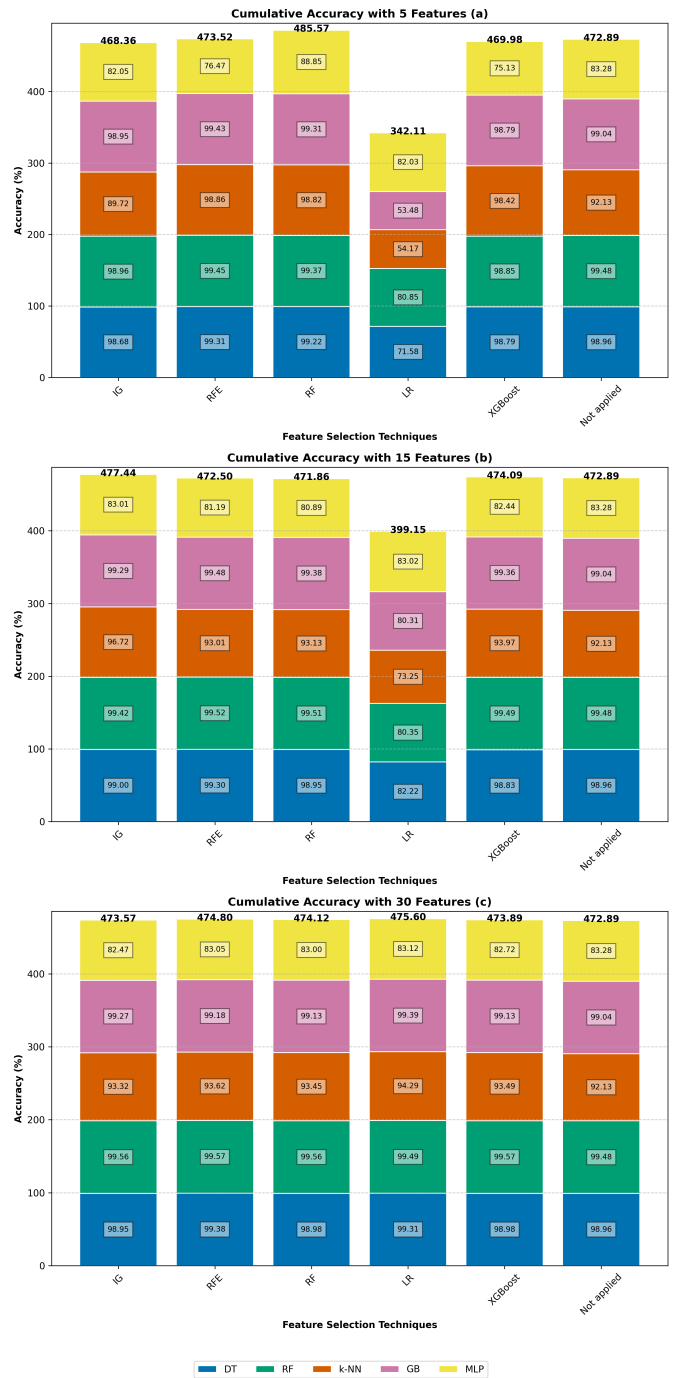
Figure 7: Cumulative model accuracy with different feature selection techniques in three schemes: a) 5-feature. b) 15-feature. c) 30-feature

and competitive results, underscoring its reliability. In contrast, LR again shows the weakest results with a cumulative accuracy of 399.15.

At 30 features (Fig. 7 (c)), the performance among techniques becomes more balanced, but LR slightly edges out with a cumulative accuracy of 475.60, showcasing its effectiveness in models like RF (99.49%) and DT (99.31%). This indicates that with a larger number of features, LR can be quite competitive. RFE and RF also provide strong, stable results, both offering high cumulative accuracies of 474.80 and 474.12 respectively, proving their reliability across different models.

Overall, RF emerges as the most stable and high-performing technique across different feature counts, consistently delivering top results, especially notable at the 5-feature level. Meanwhile, LR shows significant improvement with increased features but remains the weakest performer at lower feature counts, making it less reliable for smaller datasets. This analysis highlights RF as a superior choice for stably high performance when being used with arbitrary machine learning model.

Last but not least, one of the biggest benefits of feature selection is reducing the size of the dataset. This is even more crucial when implementing machine learning-based IDS on IoT devices with limited resources. As the highest prediction results from 3 schemes are all very good, selecting the fewest features will be the best option for IoT devices. The detail about data size reduction is shown in Tab. 10.

Table 10: Memory usage of datasets for 3 schemes on the overall dataset

Selected features scheme over total 46 features	5	15	30
Percentage / overall dataset (%)	12.77	34.04	65.96
Data size (GB) [overall dataset: 12.8]	1.64	4.36	8.44

As can be seen, the one that significantly reduces the data size is the scheme where only 5 features are selected out of the total 46 available features. This scheme shows the lowest memory usage percentage compared to the overall dataset, with a value of 12.77%. This translates to a data size of approximately 1.64 GB, which is notably smaller compared to the original dataset size of 12.8 GB.

In selecting the best combination (feature selection technique and machine learning model) within the scheme of 5-selected features, we examine the training time and performance metrics across various best combination candidates. Tab. 11 presents the results, including the training time (in seconds), accuracy, precision, recall, and F1-score for each combination.

Table 11: Result including training time for the best candidates in the scheme selecting 5 features

DT		Training time (s)	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
	RFE	22.68	99.31	99.32	99.31	99.31
RF		**Training time (s)**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-score (%)**
	RFE	714.11	99.45	99.45	99.45	99.44
k-NN		**Training time (s)**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-score (%)**
	RFE	12.96	98.86	98.82	98.86	98.80
GB		**Training time (s)**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-score (%)**
	RFE	8476.83	99.43	99.43	99.43	99.42
MLP		**Training time (s)**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-score (%)**
	RF	11727.80	88.85	89.54	88.85	88.88

As can be seen from Tab. 11, RFE stands out with the shortest training time of 12.96 seconds and impressive performance metrics.

It achieves a high accuracy of 98.86%, precision of 98.82%, recall of 98.86%, and F1-score of 98.80%. These results suggest that RFE with k-NN model trained on the 5-selected feature scheme offers the highest effectiveness in IoT attacks classification.

5 CONCLUSION

In this paper, we focused on comparing the effectiveness of feature selection methods such as IG, RFE, RF, LR, and XGBoost when used in combination with several machine learning models.

Our analysis proved that RFE has been the most accurate feature selection technique. This method achieved the average accuracy of 95.55% among different machine learning models and schemes. RFE has reached the highest accuracy of 99.57% when being used in combination with RF in case of 30 selected features. When resource constraint was taken into account, RFE was still the best choice when having the highest accuracy of 99.45% with only 5 selected features out of 46 features.

We also found that RF has been the most stable feature selection method, consistently delivering high results across all types of machine learning models and schemes. The lowest accuracy observed with RF was 83%, while the highest reached 99.56%. Therefore, if users seek a reliable method that provides good results with any model, RF is the best choice.

Our analysis revealed that selecting only 5 features out of 46 significantly reduces memory usage while maintaining strong predictive performance. This is important, particularly in resource-constrained IoT environments. RFE with k-NN emerges as a standout combination, demonstrating the effectiveness in IoT attack classification.

The newest available dataset (CIC-IoT 2023) was used. Therefore, our results have been directly applicable into the current cybersecurity solutions of IDS.

Our future work focuses on testing more feature selection methods from the categories Filter, Wrapper, Embedded and Hybrid. More indicators apart from accuracy, precision, recall and F1-score will be utilized to evaluate the strengths of each method in different aspects. Afterward, we will proceed to apply these findings to IDS implementations on IoT devices to assess their effectiveness. Besides, the dataset will be updated to the latest one if available.

ACKNOWLEDGMENTS

REFERENCES

[1] Alia Ahli, Ayesha Raza, Kevser Ovaz Akpinar, and Mustafa Akpinar. 2023. Binary and Multi-Class Classification on the IoT-23 Dataset. *2023 Advances in Science and Engineering Technology International Conferences, ASET 2023* (2023). https://doi.org/10.1109/ASET56582.2023.10180848

[2] Md Manjurul Ahsan, M. A. Parvez Mahmud, Pritom Kumar Saha, Kishor Datta Gupta, and Zahed Siddique. 2021. Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance. *Technologies* 9, 3 (2021). https://doi.org/10.3390/technologies9030052

[3] Waleed Aldosari. 2023. Deep Learning-Based Location Spoofing Attack Detection and Time-of-Arrival Estimation through Power Received in IoT Networks. *Sensors* 23, 23 (2023). https://doi.org/10.3390/s23239606

[4] Andrew Blower and Gerald Kotonya. 2019. RECON: A Real-time Entity Based Access Control for IoT. *2019 6th International Conference on Internet of Things:*

Systems, Management and Security, IOTSMS 2019 (2019), 142 – 146. https://doi.org/10.1109/IOTSMS48152.2019.8939218

[5] Akram Boukhamla and Javier Coronel Gaviro. 2021. CICIDS2017 dataset: Performance improvements and validation as a robust intrusion detection system testbed. *International Journal of Information and Computer Security* 16, 1-2 (2021), 20 – 32. https://doi.org/10.1504/IJICS.2021.117392

[6] Leo Breiman. 2001. Random forests. *Machine Learning* 45, 1 (2001), 5 – 32. https://doi.org/10.1023/A:1010933404324

[7] Xi Hang Cao, Ivan Stojkovic, and Zoran Obradovic. 2016. A robust data scaling algorithm to improve classification accuracies in biomedical data. *BMC Bioinformatics* 17, 1 (2016). https://doi.org/10.1186/s12859-016-1236-x

[8] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 13-17-August-2016 (2016), 785 – 794. https://doi.org/10.1145/2939672.2939785

[9] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 13-17-August-2016 (2016), 785 – 794. https://doi.org/10.1145/2939672.2939785

[10] Raisa Abedin Disha and Sajjad Waheed. 2022. Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique. *Cybersecurity* 5, 1 (2022). https://doi.org/10.1186/s42400-021-00103-8

[11] Christopher Faircloth, Gavin Hartzell, Nathan Callahan, and Suman Bhunia. 2022. A Study on Brute Force Attack on T-Mobile Leading to SIM-Hijacking and Identity-Theft. In *2022 IEEE World AI IoT Congress (AIIoT)*. 501–507. https://doi.org/10.1109/AIIoT54504.2022.9817175

[12] Jerome H. Friedman. 2002. Stochastic gradient boosting. *Computational Statistics and Data Analysis* 38, 4 (2002), 367 – 378. https://doi.org/10.1016/S0167-9473(01)00065-2

[13] Abdallah R. Gad, Mohamed Haggag, Ahmed A. Nashat, and Tamer M. Barakat. 2022. A Distributed Intrusion Detection System using Machine Learning for IoT based on ToN-IoT Dataset. *International Journal of Advanced Computer Science and Applications* 13, 6 (2022), 548 – 563. https://doi.org/10.14569/IJACSA.2022.0130667

[14] Mohammadnoor Injadat, Abdallah Moubayed, Ali Bou Nassif, and Abdallah Shami. 2021. Multi-Stage Optimized Machine Learning Framework for Network Intrusion Detection. *IEEE Transactions on Network and Service Management* 18, 2 (2021), 1803 – 1816. https://doi.org/10.1109/TNSM.2020.3014929

[15] A. Jyotsna and Mary E.A. Anita. 2023. A Novel Paradigm for IoT Security: ResNet-GRU Model Revolutionizes Botnet Attack Detection. *International Journal of Advanced Computer Science and Applications* 14, 12 (2023), 298 – 310. https://doi.org/10.14569/IJACSA.2023.0141231

[16] James Jin Kang, Kiran Fahd, Sitalakshmi Venkatraman, Rolando Trujillo-Rasua, and Paul Haskell-Dowland. 2019. Hybrid Routing for Man-in-the-Middle (MITM) Attack Detection in IoT Networks. In *2019 29th International Telecommunication Networks and Applications Conference (ITNAC)*. 1–6. https://doi.org/10.1109/ITNAC46935.2019.9077977

[17] Sydney M. Kasongo and Yanxia Sun. 2020. Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset. *Journal of Big Data* 7, 1 (2020). https://doi.org/10.1186/s40537-020-00379-6

[18] Kulwinder Kaur and John Ayoade. 2023. Analysis of DDoS Attacks on IoT Architecture. In *2023 10th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*. 332–337. https://doi.org/10.1109/EECSI59885.2023.10295766

[19] Naman Kaur and Himanshu. 2023. Logistic Regression: A Basic Approach. *Lecture Notes in Networks and Systems* 623 LNNS (2023), 481 – 488. https://doi.org/10.1007/978-981-19-9638-2_41

[20] James M. Keller and James A. Givens. 1985. A Fuzzy K-Nearest Neighbor Algorithm. *IEEE Transactions on Systems, Man and Cybernetics* SMC-15, 4 (1985), 580 – 585. https://doi.org/10.1109/TSMC.1985.6313426

[21] Jing Li, Mohd Shahizan Othman, Hewan Chen, and Lizawati Mi Yusuf. 2024. Optimizing IoT intrusion detection system: feature selection versus feature extraction in machine learning. *Journal of Big Data* 11, 1 (2024). https://doi.org/10.1186/s40537-024-00892-y

[22] Ling Li, Huawen Liu, Zongjie Ma, Yuchang Mo, Zhengjie Duan, Jiaqing Zhou, and Jianmin Zhao. 2014. Multi-label feature selection via information gain. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8933 (2014), 345 – 355. https://doi.org/10.1007/978-3-319-14717-8_27

[23] Wenjuan Lian, Guoqing Nie, Bin Jia, Dandan Shi, Qi Fan, and Yongquan Liang. 2020. An Intrusion Detection Method Based on Decision Tree-Recursive Feature Elimination in Ensemble Learning. *Mathematical Problems in Engineering* 2020 (11 2020), 1–15. https://doi.org/10.1155/2020/2835023

[24] Gowtham M and Pramod H B. 2022. Semantic Query-Featured Ensemble Learning Model for SQL-Injection Attack Detection in IoT-Ecosystems. *IEEE Transactions on Reliability* 71, 2 (2022), 1057–1074. https://doi.org/10.1109/TR.2021.3124331

[25] Nivedita Mishra and Sharnil Pandya. 2021. Internet of Things Applications, Security Challenges, Attacks, Intrusion Detection, and Future Visions: A Systematic Review. *IEEE Access* 9 (2021), 59353–59377. https://doi.org/10.1109/ACCESS.2021.3073408

[26] Nour Moustafa and Jill Slay. 2015. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). *2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings* (2015). https://doi.org/10.1109/MilCIS.2015.7348942

[27] Euclides Carlos Pinto Neto, Sajjad Dadkhah, Raphael Ferreira, Alireza Zohourian, Rongxing Lu, and Ali A. Ghorbani. 2023. CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment. *Sensors* 23, 13 (2023). https://doi.org/10.3390/s23135941

[28] Tarun Ganesh Palla and Shahab Tayeb. 2021. Intelligent mirai malware detection for iot nodes. *Electronics (Switzerland)* 10, 11 (2021). https://doi.org/10.3390/electronics10111241

[29] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825 – 2830. https://www.scopus.com/inward/record.uri?eid=2-s2.0-80555140075&partnerID=40&md5=63e53cee7a9711760872d4d103e5453a

[30] Maria Irmina Prasetiyowati, Nur Ulfa Maulidevi, and Kridanto Surendro. 2021. Determining threshold value on information gain feature selection to increase speed and prediction accuracy of random forest. *Journal of Big Data* 8, 1 (2021). https://doi.org/10.1186/s40537-021-00472-4

[31] J.R. Quinlan. 1986. Induction of Decision Trees. *Machine Learning* 1, 1 (1986), 81 – 106. https://doi.org/10.1023/A:1022643204877

[32] Abdu Salam, Faizan Ullah, Farhan Amin, and Mohammad Abrar. 2023. Deep Learning Techniques for Web-Based Attack Detection in Industry 5.0: A Novel Approach. *Technologies* 11, 4 (2023). https://doi.org/10.3390/technologies11040107

[33] In Statista. 2023. Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2023, with forecasts from 2022 to 2030 (in billions). https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/. Accessed: 2024-05-05.

[34] Vladimir Svetnik, Andy Liaw, Christopher Tong, J. Christopher Culberson, Robert P. Sheridan, and Bradley P. Feuston. 2003. Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. *Journal of Chemical Information and Computer Sciences* 43, 6 (2003), 1947 – 1958. https://doi.org/10.1021/ci034160g

[35] Kuan Zhang, Xiaohui Liang, Rongxing Lu, and Xuemin Shen. 2014. Sybil Attacks and Their Defenses in the Internet of Things. *IEEE Internet of Things Journal* 1, 5 (2014), 372–383. https://doi.org/10.1109/JIOT.2014.2344013

[36] Dingding Zhou, Wei Liu, Wengang Zhou, and Shi Dong. 2014. Research on network traffic identification based on multi layer perceptron. *Telkomnika (Telecommunication Computing Electronics and Control)* 12, 1 (2014), 201 – 208. https://doi.org/10.12928/TELKOMNIKA.v12i1.1051