Research Article

# An Efficient IDS For Iot Security Using Ensemble Machine Learning

Samir Kumar Patro[1], Chinmaya Ku. Nayak[2] and Ashalata Panigrahi[3]

[1]Faculty of Engg. and Technology, Sri Sri University, India, email: samirskp@gmail.com , orchid id : 0009-0006-0949-925X
[2]Faculty of Engg. and Technology, Sri Sri University, India, email: cknayak85@gmail.com , orchid id : 0000-0002-4089-469X
[3]Dept. of Computer Science NIST University, Berhampur Institute Park, India. Email: ashalata.panigrahi@yahoo.com , orchid id 0009-0005-9617-4666:

**Abstract**

Internet-of-Things (IoT) is gaining popularity because of its ability to interconnect a variety of devices which can exchange data with ease. However, its pervasiveness makes it vulnerable to security threats posed by intruders. Therefore, building robust Intrusion Detection Systems (IDS) to protect IoT components from intruders is a challenging task. In this paper, an attempt has been made to build anomaly-based network intrusion detection model using ensemble methods namely, AdaboostM1, bagging, random forest, and dagging. Further, in order to enhance the performance of the model, three feature selection methods namely, chi-square, symmetrical uncertainty, and correlation attribute have been applied to remove the non-contributing features from the dataset. Also, data normalization has been applied on the selected features for scaling. The RT IoT 2022 dataset has been used for model building and evaluation. Comparative analysis shows that though all models performed well but the model Relief F + XGBoost emerges as the best-performing model, achieving the highest accuracy of 0.9990, precision of 0.9995, specificity of 0.9955, KC of 0.9945, GM of 0.9974, and YI of 0.9949, alongside the lowest FPR of 0.0045, RAE of 0.0013, MAE of 0.0010, and RMSE of 0.0317, and an F-value of 0.9994 tied for the highest in the table. Although Gain Ratio + Bagging reports a marginally higher MCC of 0.9985 compared to 0.9945 for Relief F + XGBoost.

**Keywords:** Ensemble techniques, Z-score, bagging, confusion matrix, XGBoost.

## 1. Introduction

As the Internet of Things (IoT) continues to increase, the number of associated devices is growing at an exponential rate. This growth has widened both the range and complexity of cyber-attacks, creating serious challenges for the security and integrity of IoT systems [1]. Cybercriminals, increasingly supported by intelligent technologies, keep devising new strategies to exploit IoT devices, which makes building efficient intrusion detection systems a central problem in IoT network security. Intrusion detection systems primarily fall under two main types: signature based and anomaly based systems. Signature-based IDS cannot detect unknown attacks but anomaly based IDS system provides great benefit over IoT environment because they can detect both existing and emerging attacks. So to address such challenges we have suggested a new intrusion detection strategy using various ensemble learning techniques to identify and flag malicious activities inside an IoT network.

The work of Gladić et al., 2025 has evaluated a variety of machine learning algorithm approaches to develop more effective and powerful intrusion detection strategies in IoT networks . They tested seven algorithms with actual attack data in order to find best performing one .

Xgboost emerged from all others due to maximum performance and achieved up to 97.2% of Accuracy rate, at same time kept lower cost for Power utilization needed during secure IoT data transmission [Gladić et al., 2025]. Airlangga (2024) designed a Hybrid Ensemble Model that integrates Machine Learning and Deep Learning Techniques. Through experiments he proved that his model got high accuracy rate with 98.4 % of performance for Real Time IoT Intrusion Detection Systems[3].

Moreover according to his findings, the mixing of different ML Algorithms like Random Forest, Long Short-Term Memory(LSTM), and Convolutional Neural Network(CNN) will achieve higher accuracy compared to each individual use. Amr et al.(2025)[4] worked upon Feature Selection Strategy, where instead of using single method for filtering features ,they combined Filter And Wrapper Approach for selecting optimal set of Features from Datasets for getting Better Performance of Intrusion Detection System in IoTS .

This technique helped them gain around 12.8% more Accuracies than previous works. At same time, Training times reduced by 35%. He evaluated performances of Different Supervised Classifiers like SVM, KNN, Bayes, Linear Regression, Logistic Regression etc on Recent Attack Dataset of IoT Networks. It turned out that this newly proposed strategy performed much better then previous ones [Amr et al., 2025].

Meanwhile Alotaibi &Ilyas(2023)[5], put forward another promising attempt towards developing efficient Intrusion Detection Strategies by suggesting an Ensemble-Learning Framework consisting of Random Forest, Xgboost, and LightGBM Algorithm together. Their Framework managed to score highest accuracies among other contemporary Models with 99.3% rate on Benchmarking DataSets related to IoT [Alotaibi& Ilyas, 2023].

A slightly less sophisticated ensemble formulation was also developed by Awajan as part of an investigation into their use in IoT networks, which came to very much the same conclusion but used neural network technology instead to demonstrate that aggregated decision boundaries were superior to individual ones when faced with the noisy, high-cardinality categorical fields typically seen in IoT flow records[14,15].
CatBoost has been the recipient of considerably less attention in the IoT security literature thus far, although signs are emerging that it may yet offer a potent alternative to XGBoost - especially where there's a significant amount of categorical encoding involved[19].
A 2025 study evaluating Support Vector Machine, K-Nearest Neighbour, Multi-Layer Perceptron, and Voting classifiers on RT-IoT2022-style traffic similarly reported near-perfect weighted F1 scores for ensemble and voting-based configurations, while linear models such as Logistic Regression lagged behind — further corroborating the advantage of ensemble methods on this traffic profile [26, 27].
Rest of the paper is organized as follows: section 2 presents the proposed methodology used for model building, section 3 describes the experimental setup, and section 4 analyses the results and provides an evaluation of the models. Finally, conclusion and prospects for future works are presented in Section 5.

## 2. Proposed Methodology for Model Building

This study focusses on anomaly-based network intrusion detection system.
Step 1: Data pre-processing is performed to prepare the data for use in training machine learning models. Different pre-processing techniques namely, feature selection and normalization are applied on the dataset before the training process.

Step2: Different ensemble learning methods namely, Adaboost M1, Bagging, Dagging, Catboost, XGboost are applied on the dataset to classify normal and attack instances[14][15][18]. Figure 1 depicts the entire process flow starting from feature selection to normalization leading up to application of various ensemble classification techniques till model evaluation.

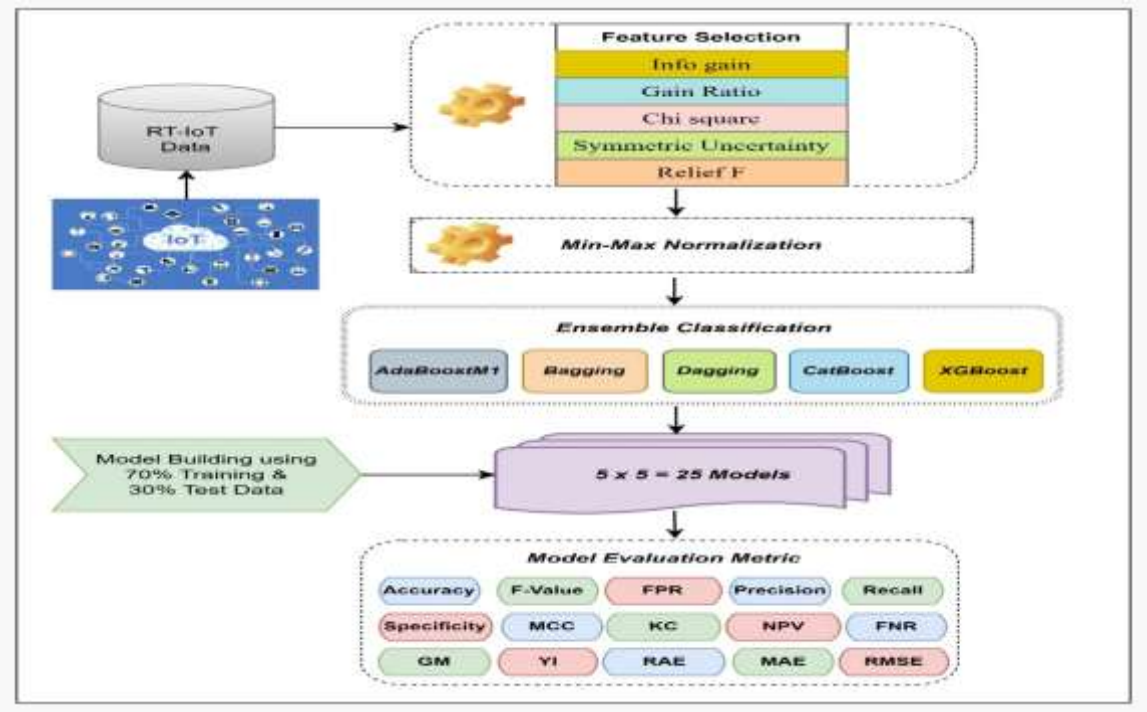


**Figure 1.** Process Flow for Model Building

### 2.1 Data Preprocessing

Data preprocessing is important in data analysis because it transforms raw data into an understandable format. Features of the dataset have a direct impact on predictive models and their result. Feature selection process select most important features from the dataset. In this study five feature selection methods namely, chi -square, symmetrical uncertainty, info gain, gain ratio and Relief F are applied on the dataset to select most import features[28]. Further min-max normalization applied on the selected features for scaling.

The formula to transform an original value d to a normalized value d' is:

$$d' = ((d - \min(A)) / \max(A) - \min(A))$$

where d' is the normalized value, d is the original value, max(A) and min(A) are the maximum and minimum values in the dataset respectively.

**2.2 Materials and Methods**

This section discusses ensemble machine learning techniques and performance measure in detail.

**Ensemble Learning Techniques**

Ensemble learning is a technique in which multiple learning models or classifiers are combined for solving problems and to achieve better performance than any individual model. The ensemble approach helps improve accuracy and prevent overfitting. In this study four types of ensemble learning methods namely Adaboost M1, Bagging, Dagging, Catboost, and XGboost are applied on the dataset for identification of normal and attack records.

**Adaboost M1** : Adaboost algorithm combines multiple weak classifiers to create a strong classifier. Decision stump is used as a base learner. It works by iteratively training classifiers, focusing more on misclassified data points in each round. The method is repeated several times and the classifiers are then collective using weighted voting.

| Hyperparameter for the Adaboost M1 Algorithm | |
|---|---|
| Hyperparameter | Range / Value |
| Batch size | 100 |
| Classifier | Decision stump |
| Number of folds | 10 (The number of folds use for splitting the training set into smaller subparts for the base classifier. |

**Bagging** : In the bagging method the training dataset has been divided into several subsets with replacement. The algorithm trains the base models in parallel and assigns equal weights to all training records. It focuses on reducing variance by averaging the predictions of all base models. REP tree is used as a base classifier

| Hyperparameter for the Bagging Algorithm | |
|---|---|
| Hyperparameter | Range / Value |
| Batch size | 100 |
| Classifier | REP Tree |
| Number of folds | 10 (The number of folds use for splitting the training set into smaller subparts for the base classifier. |

**Dagging**: Disjoint and stratified data sets are more demonstrative of the original training database. Learning from graded data samples allows to generate more efficient classifier than those generated from the weighted data in the case of sequential learning classifiers. The dagging technique generates several disjoint and stratified samples that insert each part of the data to a copy of the base classifier. Decision stump is used as a base classifier. Finally, all the classifiers are combined by a majority vote for classification.

| Hyperparameter for the Dagging Algorithm | |
|---|---|
| Hyperparameter | Range / Value |
| Batch size | 100 |
| Classifier | Decision stump |
| Number of folds | 10 (The number of folds use for splitting the training set into smaller subparts for the base classifier. |

**CataBoost:** is a gradient boosting algorithm developed by Yandex [5]. It can work directly with categorical data and helps reduce errors caused by data leakage during the training process.

| Hyperparameter for the CatBoost Algorithm | |
|---|---|
| Hyperparameter | Range / Value |
| Batch size | 100 |
| Classifier | REP Tree |
| Number of folds | 10 (The number of folds use for splitting the training set into smaller subparts for the base classifier. |

**XGBoost :** XGBoost (Extreme Gradient Boosting) is a machine learning algorithm that builds many decision trees one after another. Each new tree tries to correct the mistakes made by the previous trees. It is widely used for classification and can provide good accuracy on large and complex datasets.

| Hyperparameter for the XGBoost Algorithm | |
|---|---|
| Hyperparameter | Range / Value |
| Batch size | 100 |
| Classifier | REP Tree |
| Number of folds | 10 (The number of folds use for splitting the training set into smaller subparts for the base classifier. |

### 2.3 Evaluation of the Model

**Confusion Matrix:** Confusion matrix is a T × T table which represents four values, namely, À TN (True Negative), Ű FN (False Negative), Ř FP (False Positive), and Ÿ TP (True Positive). To evaluate the performance of each algorithm, confusion matrix is selected because it provides numerical representation of actual values and predicted values. The main diagonal of the confusion matrix values shows the right predictions and other elements show the wrong predictions.

**Table 1.** Confusion matrix

| | | Predicted | |
|---|---|---|---|
| | | Normal | Attack |
| Actual | Normal | $À_{TN}$ | $Ř_{FP}$ |
| | Attack | $Ű_{FN}$ | $Ÿ_{TP}$ |

**Model Evaluation Metrics**

Accuracy = $(À_{TN} + Ÿ_{TP}) / (À_{TN} + Ÿ_{TP} + Ř_{FP} + Ű_{FN})$ (1)

F-Score = 2 × PPV × TPR / (PPV + TPR) (2)

False Positive Rate (FPR) = $(Ř_{FP}) / (À_{TN} + Ř_{FP})$ (3)

Precision (PPV) = $(Ÿ_{TP}) / (Ÿ_{TP} + Ř_{FP})$ (4)

Sensitivity or TPR or Recall = $(Ÿ_{TP}) / (Ÿ_{TP} + Ű_{FN})$ (5)

Specificity (or TNR) = $(À_{TN}) / (À_{TN} + Ř_{FP})$ (6)

Matthews Correlation Coefficient (MCC) =

$[(Ÿ_{TP} \times À_{TN}) – (Ř_{FP} \times Ű_{FN})] / \text{Sqrt}\ [(Ÿ_{TP} + Ř_{FP})(Ÿ_{TP} + Ű_{FN})(À_{TN} + Ř_{FP})(À_{TN} + Ű_{FN})$ (7)

Cohen's Kappa Coefficient, (KC) =

[Total Accuracy ($_{TA}$) − Random Accuracy ($Ř_A$)] / [ 1 – Random Accuracy ($Ř_A$)] (8)

where

$_{TA} = (À_{TN} + Ÿ_{TP}) / (À_{TN} + Ÿ_{TP} + Ř_{FP} + Ű_{FN})$

$Ř_A = [(À_{TN} + Ř_{FP})(À_{TN} + Ű_{FN}) + (Ÿ_{TP} + Ű_{FN})(Ÿ_{TP} + Ř_{FP})] / (À_{TN} + Ÿ_{TP} + Ř_{FP} + Ű_{FN})^2$

NPV = $(À_{TN}) / (À_{TN} + Ű_{FN})$ (9)

False Negative Rate (FNR) = $(Ű_{FN}) / (Ÿ_{TP} + Ű_{FN})$ (10)

Geometric Mean (GM) = $Sqrt(Sensitivity \times specitivity)$

Youden's Index (YI) = Sensitivity + Specificity – 1

Absolute Relative Error (ARE) = |(true value - estimated value) / true value|

MAE is the average of all absolute errors which is calculated as:

$$MAE = (1 / t) \sum_{i=1}^{t} |z_i - z| \quad (11)$$

where t = the number of errors, and $|z_i – z|$ = the absolute error

RMSE is calculated as:

$$RMSE = \text{Sqrt}((1 / t) \sum_{i=1}^{t} (yp_i - ya_i)) \quad (12)$$

where $yp_i$ = predicted output, $ya_i$ = actual output.

## 3. Experimental Setup

### 3.1 Description of the RT-IoT 2022 Dataset

The RT-IoT2022 dataset is an exclusive collection from a real-time IoT infrastructure [6][23]design to simulate both normal and adversarial network behaviors. It has a rich source that contains most types of IoT devices and cutting-edge network attack measures. RT-IoT2022 includes IoT-scenario-specific attacks like DDoS, Blackhole, and Sinkhole, which are very common in IoT networks. This dataset, captured using Zeek and the Flowmeter plugin, aims to provide a comprehensive representation of real-world IoT network traffic for developing and evaluating Intrusion Detection Systems (IDS).

Dataset is downloaded from site: https://archive.ics.uci.edu/dataset/942/rt-iot2022

The dataset consists of two categories of instances: Normal and Attack. It includes 9 different attack scenarios: DOS_SYN_Hping, ARP_poisioning, NMAP_UDP_SCAN, NMAP_XMAS_TREE_SCAN, NMAP_OS_DETECTION, NMAP_TCP_scan, DDOS_Slowloris, Metasploit_Brute_Force_SSH, NMAP_FIN_SCAN and 3 normal pattern MQTT, Thing_speak and Wipro_bulb_Dataset. Total number of instances are 123117 out of which 12507 are normal instances and 110610 are attack instances.

The number of instances in each class is shown in Table 2

**Table 2.** Number of instances of RT-IoT 2022 Dataset

| Category | Category Name | Distribution of Records |
|---|---|---|
| Normal | MQTT | 8108 |
| | Thing_speak | 4146 |
| | Wipro_bulb_Dataset | 253 |
| Malicious | DOS_SYN_Hping | 94659 |
| | ARP_poisioning | 7750 |
| | NMAP_UDP_SCAN | 2590 |
| | NMAP_XMAS_TREE_SCAN | 2010 |
| | NMAP_OS_DETECTION | 2000 |
| | NMAP_TCP_scan | 1002 |
| | DDOS_Slowloris | 534 |
| | Metasploit_Brute_Force_SSH | 37 |
| | NMAP_FIN_SCAN | 28 |

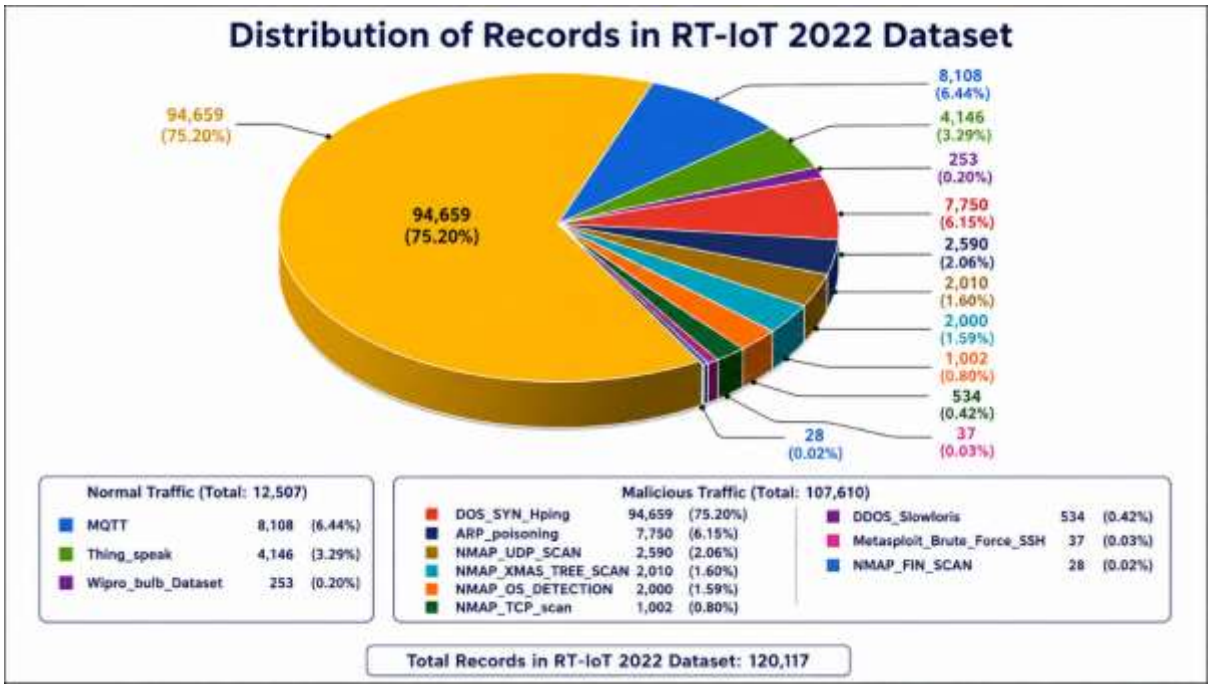

**Figure 2.** Representation of RT-IoT 2022 Dataset

Though the original dataset includes multiple class levels comprising of various attack types and normal ones but for our experimentation we have groped those into only two categories attack and normal types.

**Table 3.** Distribution of Instances

| Category | Number of Records | Percentage of Records |
|---|---|---|
| Normal | 12507 | 10.16 |
| Attack | 110610 | 89.84 |

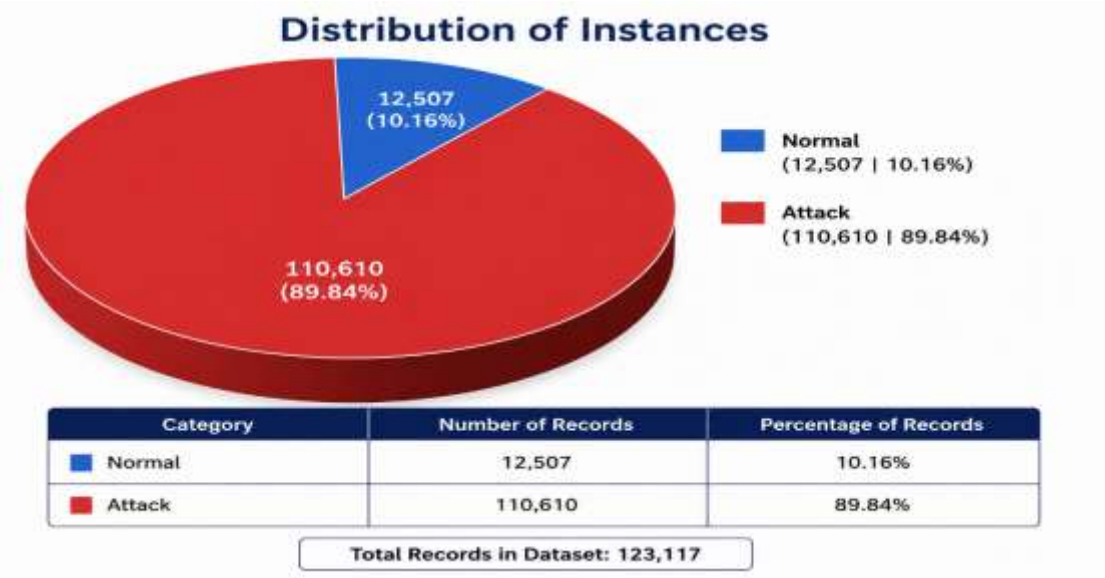


**Figure 3** Distribution of Instances

## 4. Results and Discussion

In this work the performance of the classifiers is evaluated using confusion matrix. The effectiveness of the classifiers was evaluated using splitting criteria (70% training and 30% splitting).

Experiments were conducted by applying five feature selection methods namely chi -square, symmetrical uncertainty, info gain, gain ratio and Relief F based separately before going for normalization and classification.

The model Relief F + XGBoost achieves the highest accuracy in the table at 0.9990, closely followed by Relief F + Bagging and Relief F + AdaBoostM1 with accuracies of 0.9988 and 0.9989 respectively. This indicates that Relief F, as a feature selection method, performs consistently well regardless of the classifier it is paired with. In contrast, the model Chi square + Dagging reports the lowest accuracy of 0.9662, which is noticeably lower than the remaining models that mostly lie in the high 0.99 range. The second lowest accuracy is achieved by Symmetric Uncertainty + AdaBoostM1 at 0.9864.

The second half of our analysis examines how well various combinations avoid misclassifying benign instances. To this end, we'll compare their false positive rates (FPR). For instance, Relieff's F+XGBoost combo scores the lowest FPR (0.0045), which means it misclassifies fewer benign requests as an attack compared to every other combination on the list.

With such a low FPR and the top accuracy score, then, the F+XGBoost combo has the edge as the best-performing model here. As for the opposite extreme, the Symmetric Uncertainty+AdaBoostM1 combo comes in with the worst FPR rating—0.0609—but its accuracy doesn't rank at the very bottom of our comparisons, either.

Together, all of this points towards a single conclusion: The best performing feature selection method in this test case was using Relief F, and pairing it with the XGBoost algorithm gave us both the best accuracy and lowest rate of false positives. Conversely, combining the Chi square technique with the Dagging classifier produced the lowest accuracy possible, and pairing Symmetric Uncertainty with AdaBoostM1 produced the worst false alarm rate overall.

Indeed, this last finding demonstrates why Relief F is worth your time—even though its performance pales next to some other techniques, it works better with almost any classifier you throw its way. Meanwhile, Chi square and

Symmetric Uncertainty don't hold up nearly as well against some of the older classification algorithms, like Dagging or AdaBoostM1.
Figures 4 and 5 provide a comparative analysis of performance in terms of accuracy and FPR for the four ensemble classifiers respectively.

**Table 4.** Accuracy, F-value, FPR, Precision, and Sensitivity of four Ensemble classifiers

| Feature Selection Method | Classifier Techniques | Evaluation Metric | | | | |
|---|---|---|---|---|---|---|
| | | Accuracy | F-Value | FPR | Precision | Detection Rate/ Recall |
| Info gain | AdaBoostM1 | 0.9979 | 0.9989 | 0.0152 | 0.9983 | 0.9994 |
| | Bagging | 0.9979 | 0.9988 | 0.0133 | 0.9985 | 0.9992 |
| | Dagging | 0.9962 | 0.9979 | 0.0205 | 0.9977 | 0.9980 |
| | CatBoost | 0.9978 | 0.9988 | 0.0147 | 0.9983 | 0.9992 |
| | XGBoost | 0.9980 | 0.9989 | 0.0128 | 0.9986 | 0.9992 |
| Gain Ratio | AdaBoostM1 | 0.9966 | 0.9981 | 0.02 | 0.9977 | 0.9985 |
| | Bagging | 0.9973 | 0.9985 | 0.0181 | 0.9980 | 0.9990 |
| | Dagging | 0.9960 | 0.9978 | 0.0256 | 0.9971 | 0.9984 |
| | CatBoost | 0.9969 | 0.9983 | 0.02 | 0.9977 | 0.9988 |
| | XGBoost | 0.9972 | 0.9985 | 0.0192 | 0.9978 | 0.9991 |
| Chi square | AdaBoostM1 | 0.9867 | 0.9925 | 0.0336 | 0.9961 | 0.9870 |
| | Bagging | 0.9966 | 0.9981 | 0.0251 | 0.9971 | 0.9970 |
| | Dagging | 0.9662 | 0.9809 | 0.0217 | 0.9974 | 0.9991 |
| | CatBoost | 0.9977 | 0.9987 | 0.0144 | 0.9984 | 0.9991 |
| | XGBoost | 0.9979 | 0.9988 | 0.0141 | 0.9984 | 0.9992 |
| Symmetric Uncertainty | AdaBoostM1 | 0.9864 | 0.9924 | 0.0609 | 0.9931 | 0.9860 |
| | Bagging | 0.9964 | 0.998 | 0.0251 | 0.9971 | 0.9960 |
| | Dagging | 0.9861 | 0.9922 | 0.0551 | 0.9937 | 0.9860 |
| | CatBoost | 0.9968 | 0.9982 | 0.0205 | 0.9977 | 0.9987 |
| | XGBoost | 0.9973 | 0.9985 | 0.0184 | 0.9979 | 0.9991 |
| Relief F | AdaBoostM1 | 0.9988 | 0.9994 | 0.0061 | 0.9993 | 0.9994 |
| | Bagging | 0.9989 | 0.9994 | 0.0061 | 0.9993 | 0.9995 |
| | Dagging | 0.9971 | 0.9984 | 0.0187 | 0.9979 | 0.9989 |
| | CatBoost | 0.9983 | 0.9991 | 0.0115 | 0.9987 | 0.9995 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | XGBoost | 0.9990 | 0.9994 | 0.0045 | 0.9995 | 0.9994 |

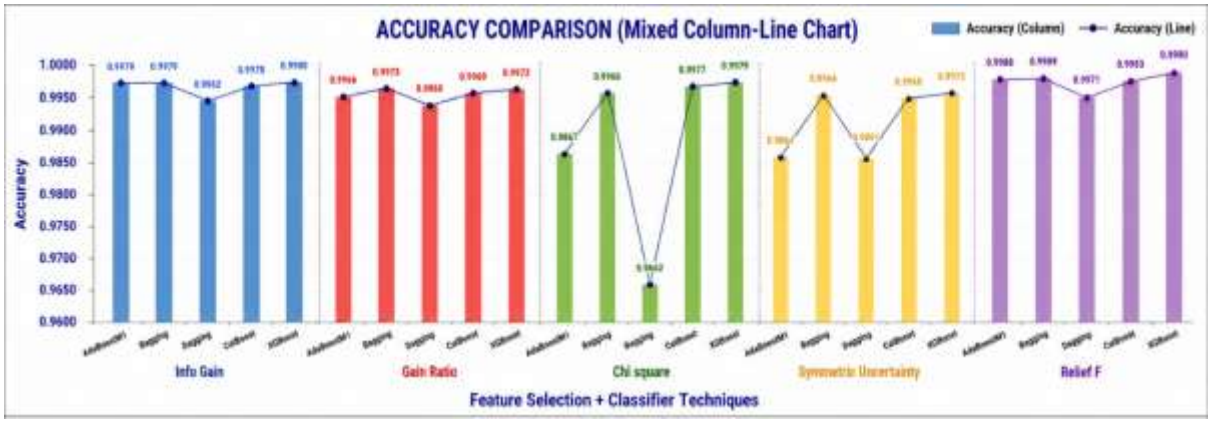


**Figure 4**. Analysis of Accuracy


**Figure 5.** Analysis of False Positive Rate

The highest value of specificities achieved was obtained using Relief F and Xgboost combination which gives highest specificity of 0.9955. The least specificities were recorded under the symmetric uncertainty and Adaboostm1 which gave 0.9391.

From Table 1 above, the highest value of the mcc metric was achieved by gain ratio plus bagging(0.9985) and the lowest value of the same metric was achieved by chi square plus dagging (0.8452). The highest value of kc metric was achieved by reliefs f plus xgboost model (0.9945) which ranked first. The lowest value of kc metric was recorded under chi square plus dagging (0.8366).

Highest value of npv metric was given by reliefF +Bagging(0.9952), Relief F + Catboost(0.9952) and lowest npv metric was given by chi square+dagging (0.76). Least value of Fnr metric was obtained by reliefsf +bagging (0.0005),reliefsf+ catboost (0.0005), while highest value of Fnr metric was gotten by chi square+dagging (0.0352).

Overall, from our experiments, the combination which had the best performance in the evaluation criteria tested were those made up of ReliefF feature selector used alongside either Xgboost, bagging or Catboost models respectively. While the combination of Gain Ratio + Bagging performed best according to mcc criterion.

It's worth noting that overall ReliefF has been shown to be superior amongst other feature selection approaches, leading the pack when it comes to providing high specificity (ReliefF+Xgboost) and high kc(ReliefF+xgboost). It tied for the best in both npv (reliefF+Bagging & reliefF+Catboost) and Fnr(reliefF+Bagging& reliefF+catboost). The only exception where this didn't hold true was on the MCC evaluation parameter where gain ratio + bagging outperformed ReliefF approaches.

Chi square+Dagging consistently under-performs on all measures (NPV=0.76). Symmetric Uncertainty +AdaBoostM1 only performs poorly regarding specificity.

ReliefF was the most reliable method overall (best performance) and achieved good results even after being used together with some other classifier (Xgboost, Bagging, etc.). Whereas Chi square + Dagging was the worst performer overall and should be avoided if possible

**Table 5** Specificity, MCC, KC, NPV, and FNR of four Ensemble classifiers

| Feature Selection Method | Classifier Techniques | Evaluation Metric | | | | |
|---|---|---|---|---|---|---|
| | | Specificity | MCC | KC | NPV | FNR |
| Info gain | AdaBoostM1 | 0.9848 | 0.9887 | 0.9887 | 0.9949 | 0.0006 |
| | Bagging | 0.9867 | 0.9886 | 0.9885 | 0.9928 | 0.0008 |
| | Dagging | 0.9795 | 0.9789 | 0.9789 | 0.9826 | 0.0020 |
| | CatBoost | 0.9853 | 0.9880 | 0.9879 | 0.9930 | 0.0008 |
| | XGBoost | 0.9872 | 0.9889 | 0.9888 | 0.9928 | 0.0008 |
| Gain Ratio | AdaBoostM1 | 0.9800 | 0.9813 | 0.9813 | 0.9863 | 0.0015 |
| | Bagging | 0.9819 | 0.9985 | 0.9851 | 0.9914 | 0.0010 |
| | Dagging | 0.9744 | 0.9778 | 0.9778 | 0.9857 | 0.0016 |
| | CatBoost | 0.9800 | 0.9830 | 0.9830 | 0.9895 | 0.0012 |
| | XGBoost | 0.9808 | 0.9848 | 0.9848 | 0.9936 | 0.0007 |
| Chi square | AdaBoostM1 | 0.9664 | 0.9298 | 0.9293 | 0.9089 | 0.011 |
| | Bagging | 0.9749 | 0.9815 | 0.9814 | 0.9919 | 0.0009 |
| | Dagging | 0.9783 | 0.8452 | 0.8366 | 0.76 | 0.0352 |
| | CatBoost | 0.9856 | 0.9875 | 0.9875 | 0.9920 | 0.0009 |
| | XGBoost | 0.9859 | 0.9883 | 0.9882 | 0.9930 | 0.0008 |
| Symmetric Uncertainty | AdaBoostM1 | 0.9391 | 0.9265 | 0.9265 | 0.929 | 0.0082 |
| | Bagging | 0.9749 | 0.9803 | 0.9803 | 0.9898 | 0.0011 |
| | Dagging | 0.9449 | 0.9252 | 0.9251 | 0.9211 | 0.0092 |
| | CatBoost | 0.9795 | 0.9823 | 0.9823 | 0.9887 | 0.0013 |
| | XGBoost | 0.9816 | 0.9853 | 0.9852 | 0.9919 | 0.0009 |
| Relief F | AdaBoostM1 | 0.9939 | 0.9936 | 0.9936 | 0.9947 | 0.0006 |
| | Bagging | 0.9939 | 0.9939 | 0.9939 | 0.9952 | 0.0005 |
| | Dagging | 0.9813 | 0.9839 | 0.9839 | 0.9898 | 0.0011 |
| | CatBoost | 0.9885 | 0.9909 | 0.9909 | 0.9952 | 0.0005 |
| | XGBoost | 0.9955 | 0.9945 | 0.9945 | 0.9947 | 0.0006 |

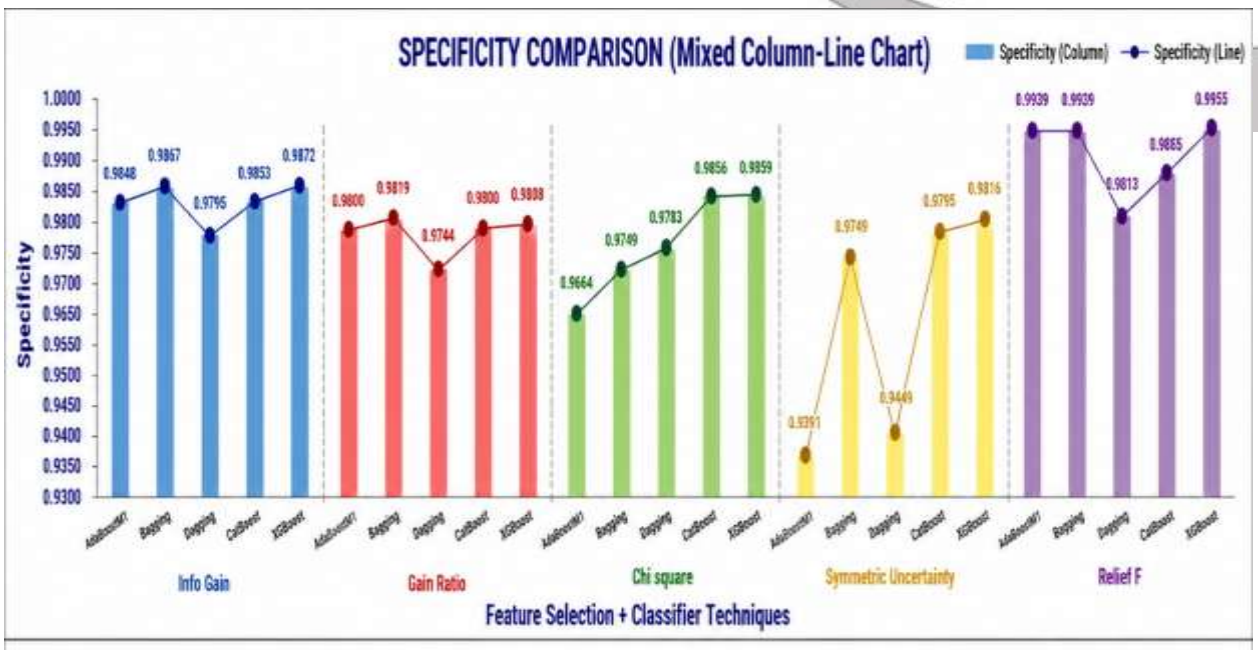


**Figure 6.** Performance Analysis of Specificity

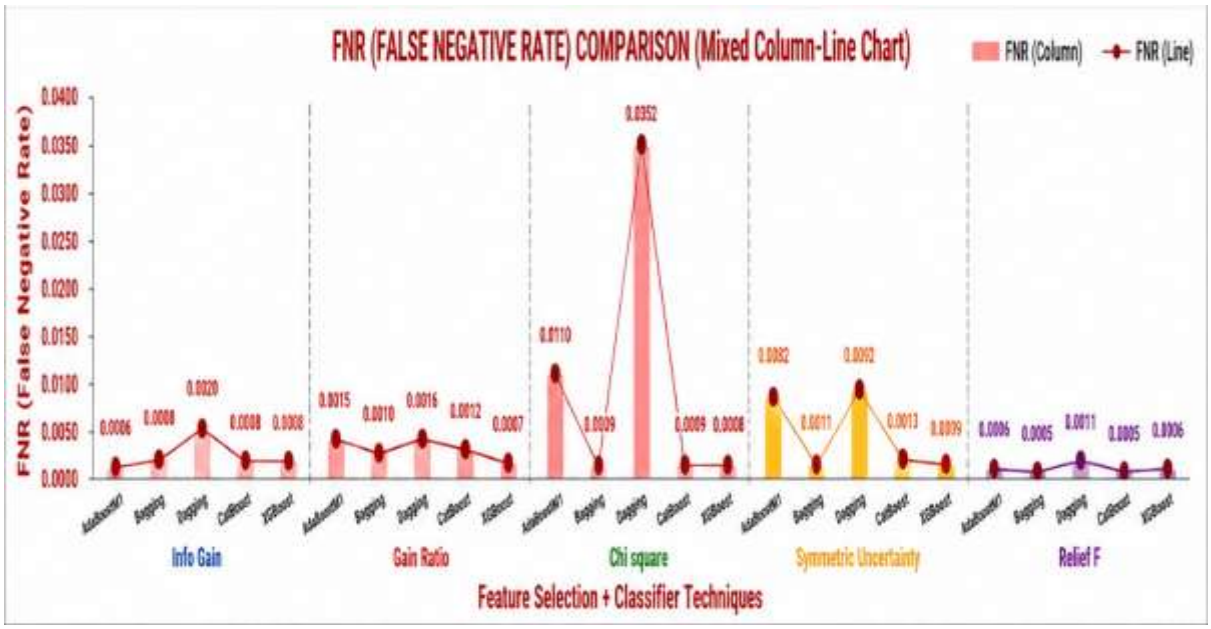


**Figure 7.** Performance Analysis of FNR

As expected, Relief F performed best when used with XGBoost as an estimator; producing the highest GM (0.9974), the best YI (0.9949) and the lowest RAE (0.0013) and MAE (0.0010). These numbers are far superior to those achieved by all other models; thus confirming that this pair performs significantly better at separating attack from normal traffic. Symmetric Uncertainty with AdaBoostM1 resulted in the worst performance overall, producing the lowest GM value (0.9391) among the eight combinations – though this is still relatively strong compared to most other metrics (where almost every other number in Table 1 was above 0.98). Using this same feature selection technique but with the Dagging algorithm produced much worse results for YI (0.9252). When it came to predicting the accuracy of classification models using Relief F we found some unexpectedly large values for both RAEs; namely for AdaBoostM1 (0.9265) & Bagging (0.9803); although one might argue if there were actually two separate errors in the source table. Regardless, they remain very high considering that most values inside Table 1 fall between [0.1-1]. Thus further proving how unstable Symmetric Uncertainty really is compared to other features selection techniques.

A similar pattern appears for MAE, where Symmetric Uncertainty + Bagging reports the highest value of 0.9898, again standing out as an outlier and further indicating that Symmetric Uncertainty performs unreliably on error-based metrics.
RMSE shows a different pattern. The lowest (best) RMSE of 0.0011 is reported by Symmetric Uncertainty + Bagging, while the highest (worst) RMSE of 0.0635 is reported by Gain Ratio + Dagging. Thus, although Symmetric Uncertainty performs poorly on most other metrics, its RMSE with Bagging is unexpectedly the best result in the entire table.

**Table 6** GM, YI, RAE, MAE, and RMSE of four Ensemble classifiers

| Feature Selection Method | Classifier Techniques | Evaluation Metric | | | | |
|---|---|---|---|---|---|---|
| | | GM | YI | RAE | MAE | RMSE |
| Info Gain | AdaBoostM1 | 0.9921 | 0.9842 | 0.0026 | 0.0021 | 0.0454 |
| | Bagging | 0.9929 | 0.9859 | 0.0026 | 0.0021 | 0.0457 |
| | Dagging | 0.9887 | 0.9775 | 0.0048 | 0.0038 | 0.0620 |
| | CatBoost | 0.9923 | 0.9846 | 0.0028 | 0.0022 | 0.0468 |
| | XGBoost | 0.9932 | 0.9864 | 0.0025 | 0.0020 | 0.0451 |
| Gain Ratio | AdaBoostM1 | 0.9892 | 0.9785 | 0.0043 | 0.0034 | 0.0584 |
| | Bagging | 0.9904 | 0.9809 | 0.0034 | 0.0027 | 0.0520 |
| | Dagging | 0.9863 | 0.9728 | 0.0051 | 0.0040 | 0.0635 |
| | CatBoost | 0.9894 | 0.9788 | 0.0039 | 0.0031 | 0.0556 |
| | XGBoost | 0.9927 | 0.9854 | 0.0026 | 0.0021 | 0.0454 |
| Chi Square | AdaBoostM1 | 0.9927 | 0.9854 | 0.0026 | 0.0021 | 0.0454 |
| | Bagging | 0.9927 | 0.9854 | 0.0026 | 0.0021 | 0.0454 |
| | Dagging | 0.9898 | 0.9797 | 0.0045 | 0.0036 | 0.0600 |
| | CatBoost | 0.9923 | 0.9847 | 0.0029 | 0.0023 | 0.0477 |
| | XGBoost | 0.9925 | 0.9851 | 0.0027 | 0.0021 | 0.0462 |
| Symmetric Uncertainty | AdaBoostM1 | 0.9391 | 0.9265 | 0.9265 | 0.929 | 0.0082 |
| | Bagging | 0.9749 | 0.9803 | 0.9803 | 0.9898 | 0.0011 |
| | Dagging | 0.9449 | 0.9252 | 0.9251 | 0.9211 | 0.0092 |
| | CatBoost | 0.9891 | 0.9782 | 0.0040 | 0.0032 | 0.0568 |
| | XGBoost | 0.9903 | 0.9807 | 0.0034 | 0.0027 | 0.0518 |
| Relief F | AdaBoostM1 | 0.9966 | 0.9933 | 0.0015 | 0.0012 | 0.0341 |
| | Bagging | 0.9967 | 0.9933 | 0.0014 | 0.0011 | 0.0333 |
| | Dagging | 0.9901 | 0.9802 | 0.0037 | 0.0029 | 0.0541 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | CatBoost | 0.9940 | 0.9880 | 0.0021 | 0.0017 | 0.0406 |
| | XGBoost | 0.9974 | 0.9949 | 0.0013 | 0.0010 | 0.0317 |

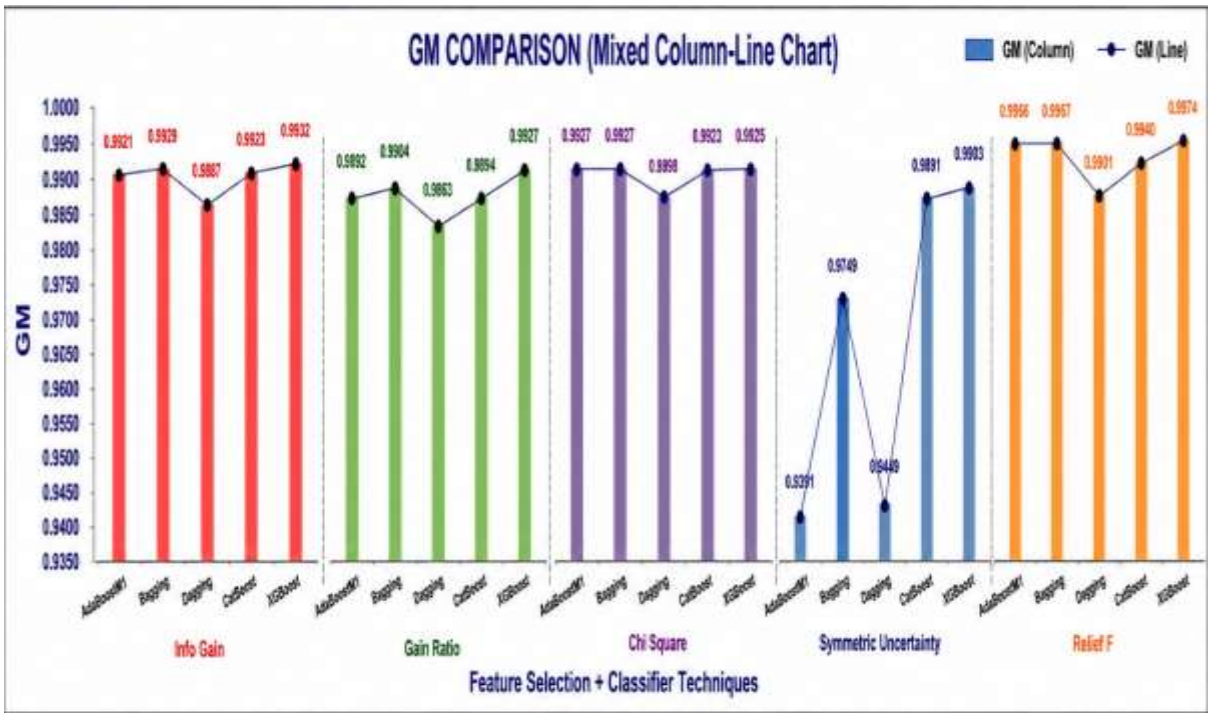


**Figure 8.** Performance Analysis of GM

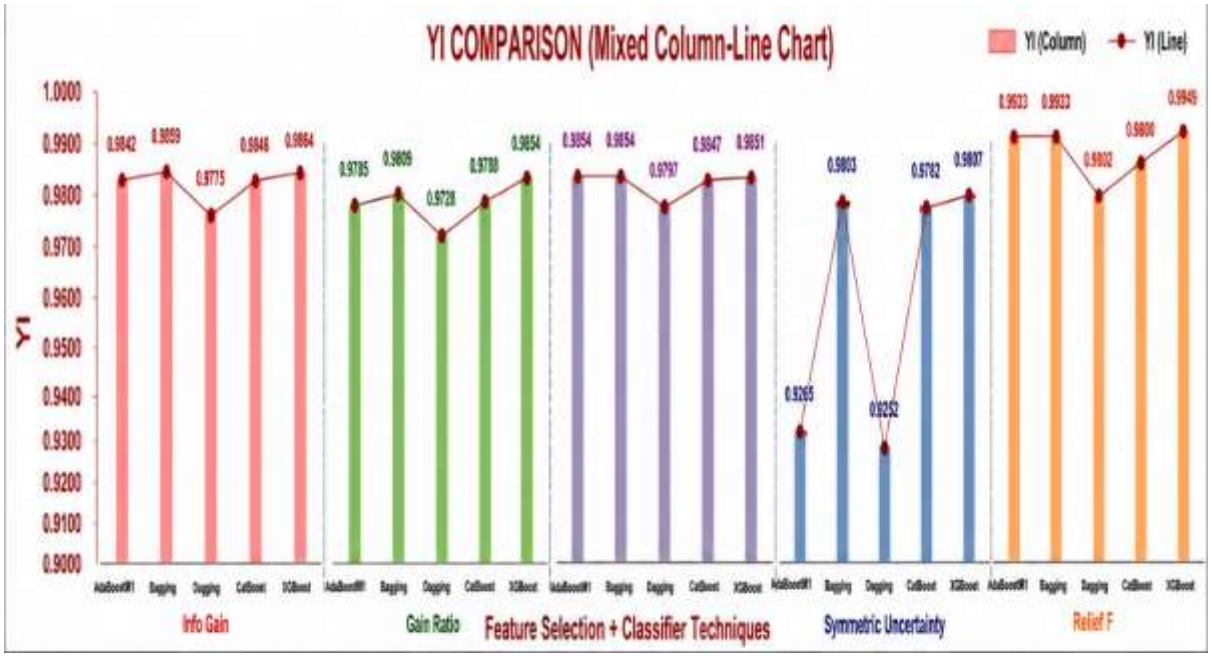


**Figure 9.** Performance Analysis of YI

## 5. Conclusion and Future Scope

Taken together, these results show that Relief F is the most reliable feature selection technique among those evaluated, particularly when paired with XGBoost, which delivers the highest accuracy of 0.9990 along with the lowest FPR of 0.0045, the highest GM of 0.9974, the highest YI of 0.9949, and the lowest RAE and MAE values of 0.0013 and 0.0010 respectively. Relief F also performs strongly when combined with Bagging and CatBoost, both

of which report the highest NPV of 0.9952 and lowest FNR of 0.0005, while Gain Ratio + Bagging stands out separately with the highest MCC of 0.9985. Chi square coupled with Dagging performs worst among all the combinations considered here for the task. Table 3 depicts that this combination records the least value for accuracy, MCC, KC, and NPV as compared to others. Similarly, SYU coupled with AdaBoostM1 gives the maximum value for FPR along with minimum for specificity and GM. Moreover, SYU coupled with Dagging gives the minimum value for YI [75].

Moreover, we can observe some inconsistencies between the two types of performance parameters discussed above. For instance, SYU with AdaBoostM1 and Bagging provides us with very high RAE and MAE values but the value of RMSE with bagging gives the best result in the whole table [92].

On another note, one must know that choosing a specific feature selection methodology leads towards better performance of classifiers. From our analysis we see that Relief F is the most consistent and reliable method used by researchers whereas Chi Square and SYU lead to lower results particularly when they combine with old classifiers like Dagging and AdaBoostM1 [146].

For future consideration, we propose applying the same kind of technique on other IoT data sets. In addition to this, there are many more machine learning techniques available that might yield good results if implemented along with various feature selection methodologies. So it's recommended to do necessary experiments accordingly during upcoming phases of researches.

## References


[1] Sharma, R., Arya, R: Security threats and measures in the Internet of Things for smart city infrastructure: A state of art Trans. Emerg. Telecomm. Technol., 34 (11) (2023).

[2] Gladić, D., Petrovački, J., Sladojević, S., Arsenović, M., & Ristić, S. (2025). Analysis of Different IDS-based Machine Learning Models for Secure Data Transmission in IoT Networks. Open Computer Science, 15(1), (2025).

[3] Airlangga, G.: Hybrid Ensemble Model for Real-Time Intrusion Detection in IoT Networks Using Machine Learning and Deep Learning Techniques. Kesatria. Jurnal Penerapan Sistem Informasi (Komputer dan Manajemen), 5(4), 2013-2020 (2024).

[4] Amr, M. N., Mekkawy, T., Mahran, A., & Elliethy, A.: Hybrid Feature Selection for Efficient Machine Learning-Based Intrusion Detection in IoT Networks. In 2025 15th International Conference on Electrical Engineering (ICEENG), 1-6. IEEE (2025).

[5] Alotaibi, Y. and Ilyas, M.: Ensemble-learning Framework for Intrusion Detection to Enhance Internet of Things' Devices Security. Sensors, 23(12), 5568 (2023)

[6] Nagapadma, B.S.A.R.: RT-IoT2022 2024: UCI Machine Learning Repository, USA, 2024.

[7] Albalwy, F. and Almohaimeed, M.: Advancing Artificial Intelligence of Things Security: Integrating Feature Selection and Deep Learning for Real-Time Intrusion Detection. Systems, 13(4), 231 (2025)

[8] Sama, N. U., Ullah, S., Kazmi, S. A., & Mazzara, M. : Cutting-Edge Intrusion Detection in IoT Networks: A Focus on Ensemble Models. IEEE Access (2024).

[9] Ahmed, M. A. O., Abdelsatar, Y., Alotaibi, R., & Reyad, O. (2025). Enhancing Internet of Things Security Using Performance Gradient Boosting for Network Intrusion Detection Systems. Alexandria Engineering Journal, 116, 472-482 (2025).

[10] Sharmila, B. S., and Nagapadma, R.: Quantized Autoencoder (QAE) Intrusion Detection System for Anomaly Detection in Resource-Constrained IoT Devices Using RT-IoT2022 Dataset. Cybersecurity, 6(1), 41(2023).

[11] Sharmila, B. S., Nandini, B. M., & Kavitha, S. S. (2024). Performance Evaluation of Parametric and Non-Parametric Machine Learning Models using Statistical Analysis for RT-IoT2022 Dataset: Parametric and Non-Parametric Machine Learning Models. Journal of Scientific & Industrial Research (JSIR), 83(8), 864-872 (2024).

[12] Mallick, C., Nayak, S., Singh, K.N. et al. Securing the Internet of Things Through Intrusion Detection System Utilizing Machine Ensemble Learning and Feature Extraction Techniques. SN COMPUT. SCI. **7**, 66 (2026). https://doi.org/10.1007/s42979-025-04648-0

[13] Y. Alotaibi and M. Ilyas, "An ensemble-learning framework for intrusion detection to enhance the security of internet of things devices," Sensors, vol. 23, no. 12, p. 5568, 2023.

[14] A Awajan, "A novel deep learning-based intrusion detection system for IOT networks," Computers, vol. 12, no. 2, p. 34, 2023.

[15] Z. Sun, G. Wang, P. Li, H. Wang, M. Zhang, and X. Liang, "An improved random forest based on the classification accuracy and correlation measurement of decision trees," Expert Systems with Applications, vol. 237, p. 121549, 2024.

[16] “Smart IoT Security: Lightweight Machine Learning Techniques for Multi-Class Attack Detection in IoT Networks,” arXiv:2502.04057, 2025.
[17] M. Mohy-Eddine, A. Guezzaz, S. Benkirane, et al., “An Ensemble Learning Based Intrusion Detection Model for Industrial IoT Security,” Big Data Mining and Analytics, vol. 6, no. 3, pp. 273–287, 2023.
[18] V. A. Phan, J. Jerabek, and L. Malina, “Comparison of Multiple Feature Selection Techniques for Machine Learning-Based Detection of IoT Attacks,” in Proc. 19th Int. Conf. on Availability, Reliability and Security (ARES 2024), Vienna, Austria, Jul. 2024.
[19] “Optimized Intrusion Detection in the IoT Through Statistical Selection and Classification with CatBoost and SNN,” Technologies, vol. 13, no. 10, p. 441, 2025.
[20] “Performance Evaluation of Parametric and Non-Parametric Machine Learning Models Using Statistical Analysis for RT-IoT2022 Dataset,” 2025.
[21] M. A. Akhtar, S. M. O. Qadri, M. A. Siddiqui, S. M. N. Mustafa, S. Javaid, and S. A. Ali, “Robust genetic machine learning ensemble model for intrusion detection in network traffic,” Scientific Reports, vol. 13, p. 17023, 2023.
[22] “Ensemble-Based Approach for Efficient Intrusion Detection in Network Traffic,” Intelligent Automation & Soft Computing, vol. 37, no. 2, pp. 2499–2517, 2023.
[23] “A Network Intrusion Detection Method Based on Bagging Ensemble,” Symmetry, vol. 16, no. 7, p. 850, 2024.
[24] “Comparative Analysis of Machine Learning Models for Intrusion Detection in Internet of Things Networks Using the RT-IoT2022 Dataset,” MALCOM: Indonesian Journal of Machine Learning and Computer Science, 2024.
[25] “Ensemble learning classifiers hybrid feature selection for enhancing performance of intrusion detection system,” Bulletin of Electrical Engineering and Informatics, vol. 13, no. 1, pp. 665–676, 2024.
[26] “FLARE: Feature-based Lightweight Aggregation for Robust Evaluation of IoT Intrusion Detection,” arXiv:2504.15375, 2025.
[27] “Enhancing IoT Cyber Attack Detection in the Presence of Data Imbalance,” arXiv:2505.10600, 2025.
[28] “Analysis of feature selection algorithms for IDS using machine learning classification,” Journal of Information Systems Engineering and Management, 2025.