Enhancing IoT Network Security Through Intrusion Detection Using Machine Learning

MSc Research Project

Cyber Security

Elizabeth Mughogho

Student ID: x22224343

School of Computing

National College of Ireland

Supervisor: Ross Spelman

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Elizabeth Mughogho		
Student ID:	x22224343		
Programme	MSc Cyber Security	Year	2025
Module:	M.Sc. Research Project		
Lecturer:	Ross Spelman		
Submission Due Date:	24/04/2025		
Project Title:	Enhancing IoT Network Security Through Intrusion Detection Using Machine Learning		
Word Count:	5540	Page Count: 20	

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project. ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use another author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:
Date:

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing IoT Network Security Through Intrusion Detection Using Machine Learning

Elizabeth Mughogho
x22224343

Abstract

With the increasing growth of Internet of Things (IoT) devices, network infrastructures have become more vulnerable to cyber-attacks. Traditional network security measures often fall short in detecting sophisticated intrusion patterns in real-time, highlighting the need for intelligent and integrated detection systems. This study proposes a machine learning-based approach to enhance IoT network security by leveraging advanced classification models. The process involves data preprocessing, normalisation, and feature selection using mutual information to identify the most impactful attributes. We evaluated several supervised learning algorithms specifically, Decision Trees, Random Forests, and XGBoost using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. To enhance detection performance further, we implemented a soft voting ensemble classifier that combines the strengths of the individual models. The study also focuses on binary classification by distinguishing benign from malicious traffic, simplifying the real-time detection tasks.

The ensemble model demonstrates superior accuracy, robustness, and generalisation, making it a viable solution for modern IoT intrusion detection systems. All experiments and evaluations are conducted using the CIC IoT 2023 dataset, a comprehensive and up-to-date benchmark for IoT security research.

INTRODUCTION

The Internet of Things (IoT) has emerged as a transformative technology that enables seamless communication among interconnected devices and offers substantial benefits across various domains, such as smart cities, healthcare, industrial automation, agriculture, and logistics [1]. However, the rapid proliferation of IoT devices has introduced significant cybersecurity challenges. Many IoT devices lack robust security mechanisms, which makes them susceptible to a wide range of cyber threats-including botnets, data breaches, and denial-of-service attacks [2], [3]. These vulnerabilities are further exacerbated by the resource-constrained nature of IoT devices and their heterogeneous architectures, which create a vast and complex attack surface for malicious actors [4], [5]. Ensuring the security of IoT networks is critical, especially considering the potential exposure of sensitive personal and industrial data [6], [7].

Intrusion Detection Systems (IDS) play a key role in network security by monitoring traffic and identifying abnormal patterns indicative of cyber-attacks [10]. While traditional IDS models are largely signature-based and limited to known threats, anomaly-based systems provide better adaptability for detecting novel attacks. However, they often suffer from high false positive rates [11-13].

To address these limitations, recent advancements in Machine Learning (ML) have shown promise in building intelligent and adaptive IDS frameworks capable of learning from large volumes of complex data [14]. ML-based IDS can not only detect known and unknown attack types but also adapt to dynamic network behaviours over time, thus enhancing the robustness and accuracy of intrusion detection in IoT environments [15], [16]. These systems can scale efficiently, offer real-time detection, and reduce the reliance on manual rule updates [14], [12].

Given the vast data generated in IoT ecosystems and the diversity of attack vectors, selecting appropriate ML algorithms and optimising their performance for intrusion detection is a crucial challenge. An effective IDS must be capable of handling the complexity, heterogeneity, and volume of IoT traffic while maintaining efficiency in terms of training time and resource utilisation [14].

This study proposes a machine learning-based approach to improving IoT network security by leveraging supervised classification models such as Decision Trees, Random Forests, and XGBoost. Feature selection using mutual information is applied to enhance model interpretability and performance. Moreover, a soft voting ensemble is implemented to combine the strengths of individual models, aiming to improve detection accuracy and generalisation. The evaluation is performed using the CIC IoT 2023 dataset, a comprehensive benchmark for contemporary IoT threat scenarios. The results demonstrate that ensemble-based IDS offers significant improvements in detecting both known and unknown threats, making it a viable and scalable solution for real-world IoT security applications.

The major contributions are as follows:

- The project presents a robust intrusion detection framework specifically tailored for IoT networks by using advanced supervised machine learning algorithms. The system is designed to address the unique challenges of IoT environments, such as resource constraints, mixed data, and evolving attack vectors.
- To improve detection performance, a soft voting ensemble classifier is proposed, combining the outputs of individual models. This approach uses the strengths of each classifier and offers better performance in terms of accuracy, generalisation, and robustness against various attack types.
- The study utilises mutual information-based feature selection to identify the most relevant attributes, reducing model complexity and enhancing the interpretation and efficiency without compromising accuracy.

All experiments are conducted using the CIC IoT 2023 dataset, which is a recent and comprehensive benchmark containing various IoT attack scenarios. This ensures that the model is evaluated under realistic and diverse threat conditions.

LITERATURE REVIEW

The proliferation of IoT devices has led to a significant increase in network-based threats, necessitating the development of robust and intelligent Intrusion Detection Systems (IDS). Traditional signature-based IDS struggles to detect zero-day or evolving attacks in real-time. Machine learning (ML) and deep learning (DL) approaches have been extensively researched to enhance detection accuracy, speed, and adaptability in IoT environments.

Recent studies have highlighted the superiority of ML/DL techniques over traditional models in identifying both known and novel threats. Meliboev et al. [17] conducted a comparative study using SVM, ANN, DT, logistic regression, and KNN on the ToN-IoT and Bot-IoT datasets, demonstrating that ANNs outperformed the other models and emphasising the efficacy of neural architectures for anomaly detection in IoT traffic.

Similarly, Hai and Nam [18] employed the Kernel Extreme Learning Machine (KELM) for both binary and multiclass classification tasks on the N-BaIoT and UNSW-NB15 datasets, achieving detection accuracies of 99.4% and 98.64%, respectively. Their work confirms the importance of algorithm selection and dataset diversity in achieving high generalisability.

Z. Zhang et al. [19] propose a hybrid intrusion detection method combining improved Fuzzy C-Means (FCM) and Support Vector Machine (SVM) to overcome the limitations of existing intrusion detection systems, such as low detection rates and high false alarm rates. This method, FCM enhanced with an information gain ratio, clusters the pre-processed dataset, followed by SVM classification. Using the NSL-KDD dataset, the experimental results show that this approach improves detection effectiveness and reduces false alarm rates compared to other methods, demonstrating higher-level performance in intrusion detection.

F. Rehman et al. [20] present a Hybrid Intrusion Detection System (HIDS) that combines signature-based and AI-powered anomaly detection, using Gradient Boosting and K-Nearest Neighbors (KNN) to achieve 90.37% accuracy. The system enhances detection precision, reduces response time, and minimises false positives, with alerts sent to security centres. The study highlights the potential for real-time cyber threat detection and future cross-platform support.

R. Zhang et al. [21] optimise the IPSO-SVM algorithm, combining Support Vector Machine (SVM) with Improved Particle Swarm Optimisation (IPSO), for enhanced network intrusion detection. The proposed architecture simplifies the detection process by classifying samples and selecting optimal parameters through iterative processing. Experimental results demonstrate that the method accurately identifies intrusion attacks, making it an effective network intrusion detection tool.

S. Zhenget al. [22] explores the application of convolutional neural networks (CNNs) in network intrusion detection, addressing the limitations of traditional methods due to the increasing complexity and volume of network data. The proposed intelligent detection model actively learns and improves over time. Experiments on the KDD99 dataset demonstrate that the model enhances both the accuracy and adaptability of intrusion detection, offering significant progress in the field.

Sama et al. [23] employed metaheuristic techniques-such as Particle Swarm Optimisation (PSO), Genetic

Algorithms (GA), and Differential Evolution (DE), for feature selection on the NSL-KDD dataset. When combined with classifiers like KNN and Decision Trees, these methods effectively reduced computational costs while maintaining high classification accuracy. Ahmad et al. [24] proposed a cluster-based feature extraction strategy using UNSW-NB15 data, addressing range and overfitting issues. Their approach used RF, SVM, and ANN models and achieved up to 98.67% accuracy in binary classification. Yaras et al. [25] proposed a hybrid 1D CNN-LSTM model trained on CICIoT2023 and ToN-IoT datasets. This architecture excelled in both binary (99.995%) and multiclass (99.96%) scenarios, validating the power of hybrid DL models in capturing temporal and spatial patterns in IoT traffic. Mankodiya et al. [26] employed stacked machine learning (ML) techniques for trust management in IoT systems. Using RF and DT on the VeRiMi dataset, their ensemble achieved over 98.5% accuracy, highlighting ensemble models' potential in diverse IoT environments.

RESEARCH METHODOLOGY

The block diagram illustrates Fig. 1. A structured methodology for IoT attack detection using an ensemble learning approach. The process begins with the CIC-IoT2023 dataset, a comprehensive and labelled dataset specifically designed for IoT-based network intrusion detection. First, the data is passed through a data pre-processing stage, during which irrelevant or missing values are handled, and the dataset is cleaned to enhance accuracy and performance. Following this, the data undergoes data normalisation, which scales features to a standard range to ensure uniformity and improve model training efficiency.

Next, Feature Selection is applied to identify the most significant attributes from the dataset, reducing dimensionality and computational overhead while enhancing model performance. The selected features are then split into the Training Set and Testing Set, where the training set is used to train the ensemble model, and the testing set is reserved to evaluate its effectiveness.

The Ensemble Model, composed of Decision Tree (DT), Random Forest (RF), and XGBoost (XGB) classifiers, is then applied to the training data. These models work together to improve classification accuracy and robustness by leveraging their individual strengths. The output of this phase is a Trained Model, which is subsequently tested using the testing set. Finally, the model generates a Predicted Attack output, identifying potential threats within the IoT network with enhanced precision.

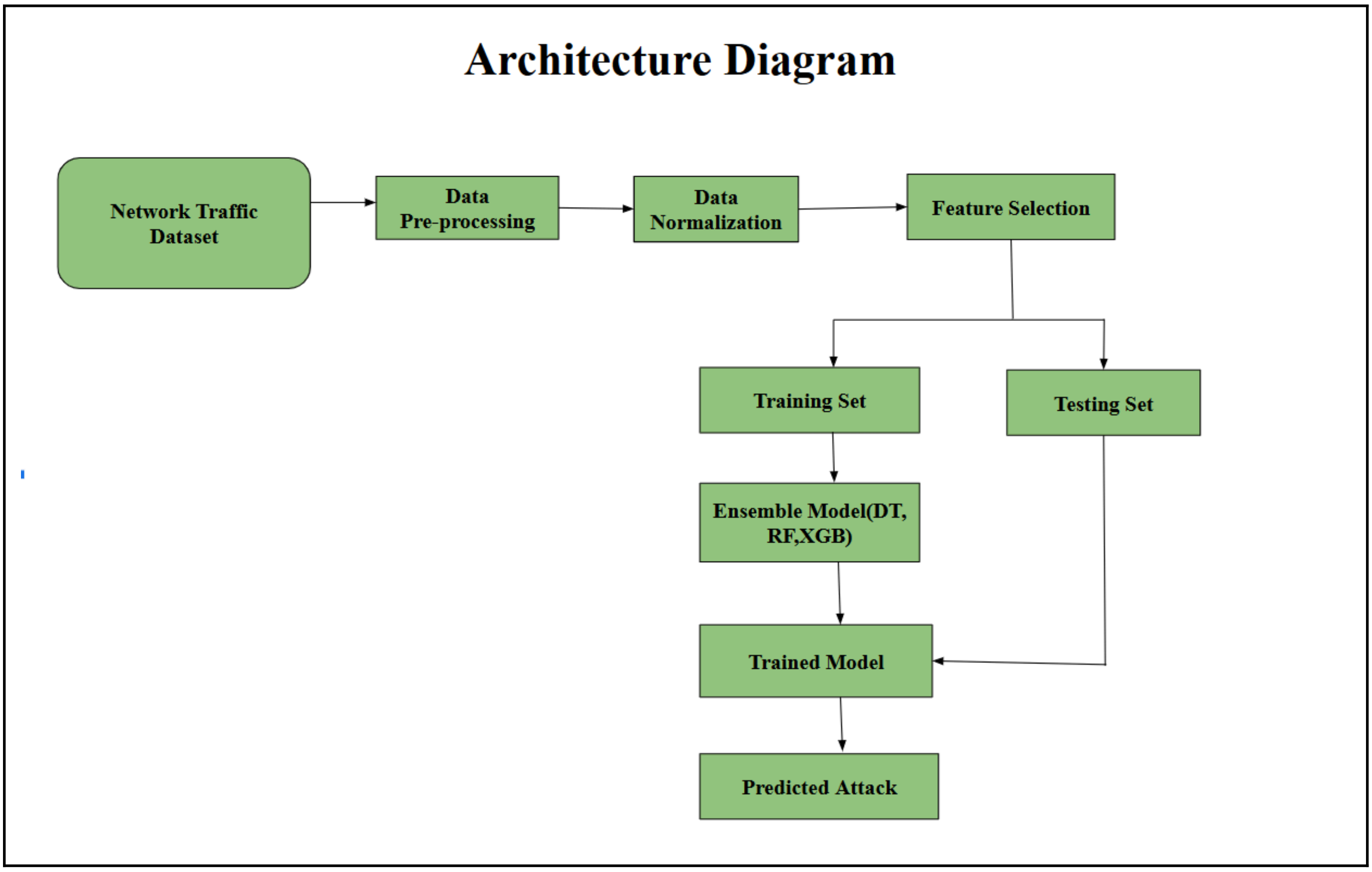


Fig. 1 Block diagram of IoT attack detection using an ensemble learning approach

1. DATA PRE-PROCESSING

The data pre-processing stage ensures the CIC-IoT2023 dataset is clean and ready for analysis. Missing or incomplete values are either filled or removed to maintain data quality. Irrelevant or constant features are dropped, and categorical data is converted into a numerical format. Duplicate entries and outliers are filtered out to prevent model distortion.

2. DATA NORMALIZATION

The datasets are pre-processed using min-max normalisation, as outlined in Equation 1. This technique brings numerical features with varying value ranges onto a unified scale.

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

In this equation, $X_{normalized}$ represents the normalised value, while ￼ and X_{max} correspond to the minimum and maximum values found in the original column$X$. This approach rescales all data points within the column to fall between 0 and 1. In this equation, $X_{normalized}$ represents the normalisedX_{min} value, while X_{max} and Xcorrespond to the minimum and maximum values found in the original data column ￼. This approach rescales all data points within the column to fall between 0 and 1.

Applying this normalisation method minimises the impact of large inconsistencies in feature ranges, which can hinder model merging. As a result, it verifies that all features contribute equally during analysis, promoting consistency and stability throughout the learning process.

3. FEATURE SELECTION

Feature selection addresses these challenges by identifying a subset of features that are most relevant to the target variable (e.g., normal, or malicious behaviour), thereby enhancing model performance and clarity.

Mutual information (MI) is a widely used metric in information theory that measures the amount of information shared between two random variables. In the context of feature selection, MI quantifies the dependency between an individual feature and the target class, capturing both linear and non-linear relationships [27]. A higher mutual information score indicates a stronger association between the feature and the target, suggesting that the feature contributes more significantly to the classification task.

The mutual information between a feature M and a class label N is defined as:

Here, Mrepresents the joint probability distribution of M and N where $p(m)$ and $p(m)$ d$p(m)$tributions. This formulation captures the reduction in uncertainty about$M$￼when ￼ is known, making it a powerful criterion for selecting features in classification tasks, while $p(m)$ and $p(m)$ denote their marginal distributions. This formulation captures the reduction in uncertainty about N when M is known, making it a powerful criterion for selecting features in classification tasks.ures in classification tasks., while $p(m)$ and $p(m)$ denote their marginal distributions. This formulation captures the reduction in uncertainty about N when M is known, making it a powerful criterion for selecting features in classification tasks.

$$Y(M, N) = \sum_{m\in M} \sum_{n=N} p(m, n) log\left(\frac{p(m,n)}{p(m)p(y)}\right)$$

The process begins by estimating the mutual information between each feature and the target variable using the training dataset. These scores are then normalised such that their sum equals 1, enabling the comparison of relative importance across features. The features are ranked in descending order based on their normalised mutual information scores. To determine the optimal subset of features, a cumulative importance thresholding strategy is applied. Features are selected sequentially, starting from the most important, until their cumulative contribution exceeds a predetermined threshold, typically 90% of the total normalised mutual information [28]. This ensures that most of the informative content is preserved while reducing dimensionality.

Let the normalised mutual information score for the feature i be represented as:

$$\hat{Y}_i = \frac{Y(M_i;N)}{\sum_{j=1}^{k} Y(M_j;N)}$$

A subset S⊂ {1, 2...,n} is then selected such that:

$$\sum_{0\in S} \hat{Y}_i \geq 0.9$$

This approach strikes a balance between dimensionality reduction and information retention, selecting only those features that collectively explain most of the variance in the class labels.

4. ENSEMBLE LEARNING

Ensemble learning is a technique in machine learning where several models are trained, and their results are combined to tackle a specific problem. The idea is that by using a group of models instead of relying on just one, you can balance out their weaknesses and make the overall prediction more accurate and reliable. One often used method in this category is the Voting Classifier. It works by gathering predictions from different models and then making a final decision based on what most of them suggest. This often leads to better performance, effectiveness, and greater stability in the model's results.

There are two types of Voting Classifiers: hard and soft. With hard voting, each model picks a class, and the one that gets the most votes is chosen as the result. Soft voting, on the other hand, looks at the predicted probabilities from each model, averages them, and picks the class with the highest overall score. Soft voting often gives better results when the models are good at estimating probabilities.

This work makes use of a soft voting classifier to combine the predictions from several individual models. The idea is to take the probability estimates each model gives for a particular class and average them out. The final prediction is based on which class has the highest average probability. The equation for this is:

$$g(j) = arg\ P(i) = arg\left(\frac{1}{N} \sum_{i=1}^{N} P(j|x_i) \right)$$

In this formula, $P(j|x_i)$ is the probability that the i^{th} model assigns to class j for a specific input x, and N is the total number of models used. The class with the highest average score becomes the final output.i^{th} model assigns to class j for a specific input x, and N is the total number of models used. The class with the highest average score becomes the final output.

A) Decision Trees (DT)

Decision trees are commonly used in supervised machine learning for both classification and regression problems. The idea is to split the dataset into smaller parts based on the values of input features. Each internal node in the tree represents a decision made using one of the features, branches represent the outcome of those decisions, and leaf nodes give the result, either a class label or a numerical value. The algorithm picks the best feature at each step to split the data and keeps going until a certain condition is met. Decision trees are easy to follow and interpret, but they can overfit the training data if not properly controlled, especially when the tree becomes too complex or when the data is noisy [29].

B) Random Forest (RF)

Random Forest is an ensemble learning method that builds a group of decision trees and combines their outputs to make more accurate predictions. It works well for both classification and regression tasks. Each tree is trained on a random sample of the data and uses a random selection of features, which helps reduce excessive fitting and improves flexibility. For classification, the final output is based on a majority vote across all trees, while for regression, it's the average of their predictions. One of the advantages of Random Forest is that it can also highlight which features are most important in making predictions. Thanks to its flexibility, accuracy, and ability to handle complex datasets, it's become a go-to method in many real-world machine learning projects [30].

C) Extreme Gradient Boosting (XGBoost)

XGBoost is an advanced approach to gradient boosting, designed for higher speed and improved performance compared to traditional boosting methods. It builds a series of decision trees in sequence, where each tree tries to fix the mistakes made by the previous one. A key feature of XGBoost is its Normalisation techniques, which help to reduce overfitting and enhance the model's ability to generalise well to new data. Also, it supports parallel computation, which handles missing data efficiently, and can work with large, complex datasets. XGBoost has become popular due to its impressive accuracy and speed, often dominating machine learning competitions and being used for real-world tasks [31].

DESIGN AND IMPLEMENTATION

Dataset Description

The CIC IoT 2023 Dataset [32] was created to support the development of security solutions for IoT networks by offering a realistic and diverse set of data. It includes network traffic collected from 105 IoT devices, with 33 different types of cyberattacks executed on them. These attacks are grouped into seven categories: Distributed Denial of Service (DDoS), Denial of Service (DoS), Reconnaissance, Web-based, Brute Force, Spoofing, and the Mirai botnet. The attacks were carried out by malicious IoT devices targeting other devices within the network. The dataset provides CSV files with features extracted from the network traffic, which can be used to train and assess machine learning models designed to distinguish between benign and malicious IoT network activity. By detailing these attack scenarios, the CIC IoT 2023 dataset serves as a valuable resource for testing and improving IoT security systems in real-world environments.

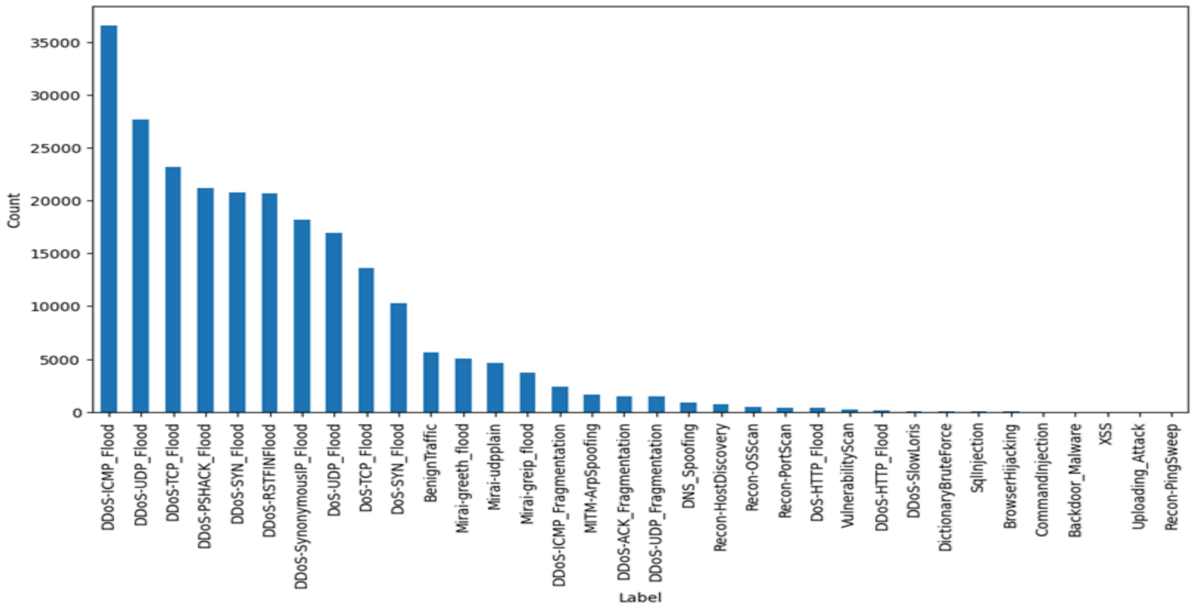
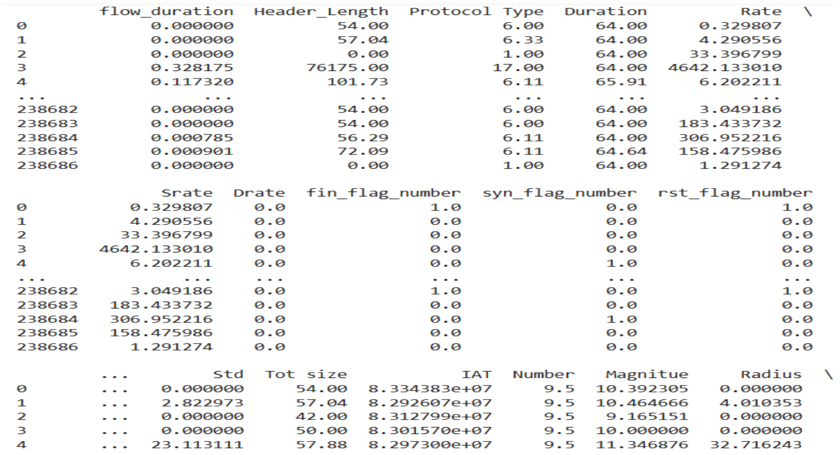


Fig. 2 Class Distribution

1. Data Understanding and Preprocessing

The design and implementation of the intrusion detection system for IoT networks begins with data preprocessing. This step involves loading and cleaning the dataset, handling missing values, and normalising the features to ensure they are on a uniform scale. Label encoding is applied to the target variable, which helps in converting categorical labels into numerical values. Feature selection is performed next to retain the most informative variables that contribute to accurate classification. The dataset is then split into training and testing sets to evaluate model performance effectively.

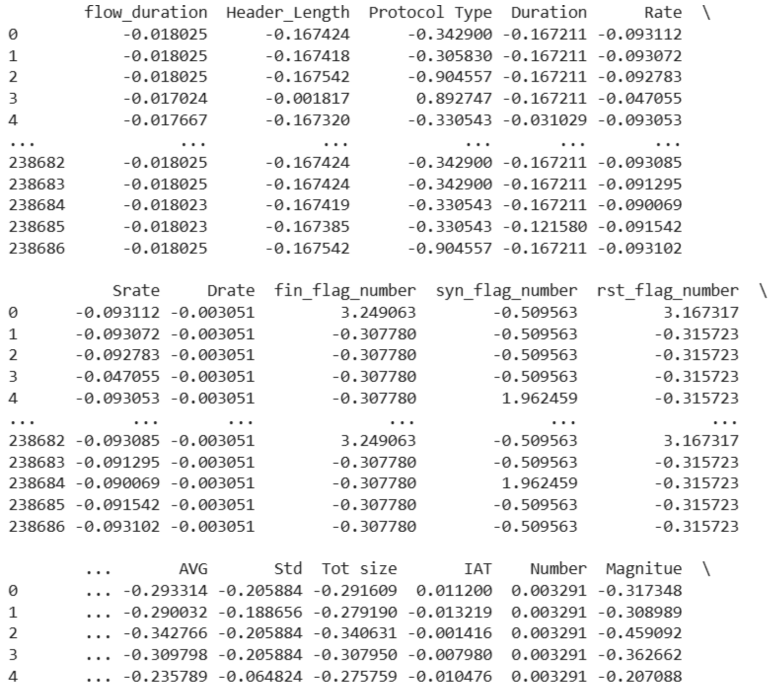


```
        flow_duration  Header_Length  Protocol Type  Duration         Rate  \
0            0.000000          54.00           6.00     64.00     0.329807
1            0.000000          57.04           6.33     64.00     4.290556
2            0.000000           0.00           1.00     64.00    33.396799
3            0.328175       76175.00          17.00     64.00  4642.133010
4            0.117320         101.73           6.11     65.91     6.202211
...               ...            ...            ...       ...          ...
238682       0.000000          54.00           6.00     64.00     3.049186
238683       0.000000          54.00           6.00     64.00   183.433732
238684       0.000785          56.29           6.11     64.00   306.952216
238685       0.000901          72.09           6.11     64.64   158.475986
238686       0.000000           0.00           1.00     64.00     1.291274

              Srate  Drate  fin_flag_number  syn_flag_number  rst_flag_number
0          0.329807    0.0              1.0              0.0              1.0
1          4.290556    0.0              0.0              0.0              0.0
2         33.396799    0.0              0.0              0.0              0.0
3       4642.133010    0.0              0.0              0.0              0.0
4          6.202211    0.0              0.0              1.0              0.0
...             ...    ...              ...              ...              ...
238682     3.049186    0.0              1.0              0.0              1.0
238683   183.433732    0.0              0.0              0.0              0.0
238684   306.952216    0.0              0.0              1.0              0.0
238685   158.475986    0.0              0.0              0.0              0.0
238686     1.291274    0.0              0.0              0.0              0.0

        ...        Std  Tot size           IAT  Number   Magnitue     Radius  \
0       ...   0.000000     54.00  8.334383e+07     9.5  10.392305   0.000000
1       ...   2.822973     57.04  8.292607e+07     9.5  10.464666   4.010353
2       ...   0.000000     42.00  8.312799e+07     9.5   9.165151   0.000000
3       ...   0.000000     50.00  8.301570e+07     9.5  10.000000   0.000000
4       ...  23.113111     57.88  8.297300e+07     9.5  11.346876  32.716243
```

Fig. 3 Original Data



```
        flow_duration  Header_Length  Protocol Type  Duration      Rate  \
0           -0.018025      -0.167424      -0.342900 -0.167211 -0.093112   
1           -0.018025      -0.167418      -0.305830 -0.167211 -0.093072   
2           -0.018025      -0.167542      -0.904557 -0.167211 -0.092783   
3           -0.017024      -0.001817       0.892747 -0.167211 -0.047055   
4           -0.017667      -0.167320      -0.330543 -0.031029 -0.093053   
...               ...            ...            ...       ...       ...   
238682      -0.018025      -0.167424      -0.342900 -0.167211 -0.093085   
238683      -0.018025      -0.167424      -0.342900 -0.167211 -0.091295   
238684      -0.018023      -0.167419      -0.330543 -0.167211 -0.090069   
238685      -0.018023      -0.167385      -0.330543 -0.121580 -0.091542   
238686      -0.018025      -0.167542      -0.904557 -0.167211 -0.093102   

           Srate     Drate  fin_flag_number  syn_flag_number  rst_flag_number  \
0      -0.093112 -0.003051         3.249063        -0.509563         3.167317   
1      -0.093072 -0.003051        -0.307780        -0.509563        -0.315723   
2      -0.092783 -0.003051        -0.307780        -0.509563        -0.315723   
3      -0.047055 -0.003051        -0.307780        -0.509563        -0.315723   
4      -0.093053 -0.003051        -0.307780         1.962459        -0.315723   
...          ...       ...              ...              ...              ...   
238682 -0.093085 -0.003051         3.249063        -0.509563         3.167317   
238683 -0.091295 -0.003051        -0.307780        -0.509563        -0.315723   
238684 -0.090069 -0.003051        -0.307780         1.962459        -0.315723   
238685 -0.091542 -0.003051        -0.307780        -0.509563        -0.315723   
238686 -0.093102 -0.003051        -0.307780        -0.509563        -0.315723   

        ...       AVG       Std  Tot size       IAT    Number  Magnitue  \
0       ... -0.293314 -0.205884 -0.291609  0.011200  0.003291 -0.317348   
1       ... -0.290032 -0.188656 -0.279190 -0.013219  0.003291 -0.308989   
2       ... -0.342766 -0.205884 -0.340631 -0.001416  0.003291 -0.459092   
3       ... -0.309798 -0.205884 -0.307950 -0.007980  0.003291 -0.362662   
4       ... -0.235789 -0.064824 -0.275759 -0.010476  0.003291 -0.207088   
```

Fig. 4 Original Data After Data Preprocessing

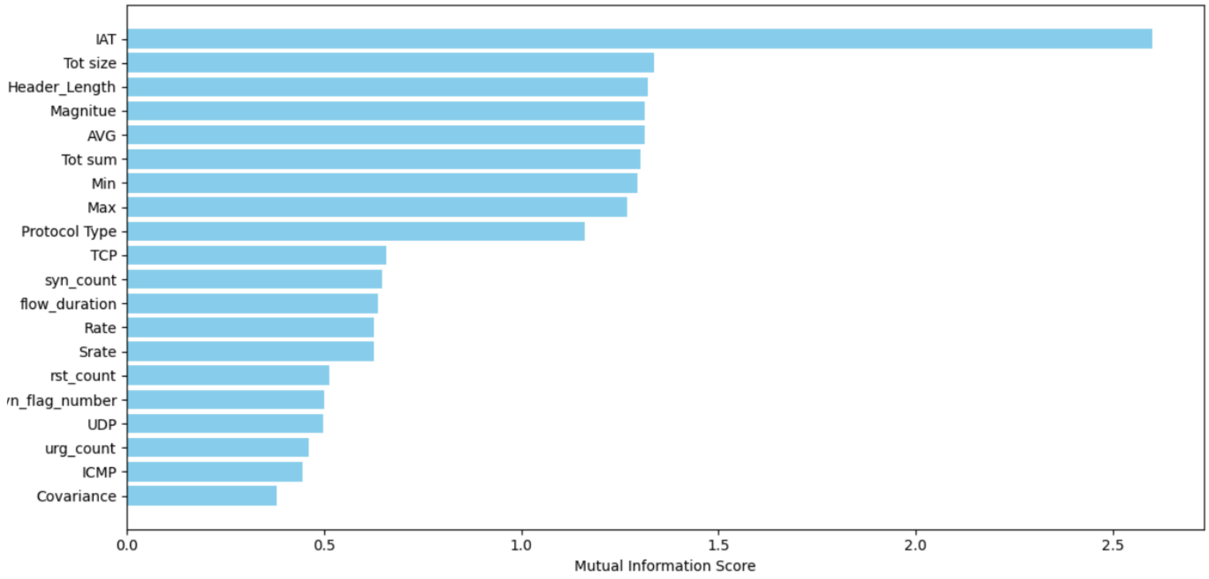
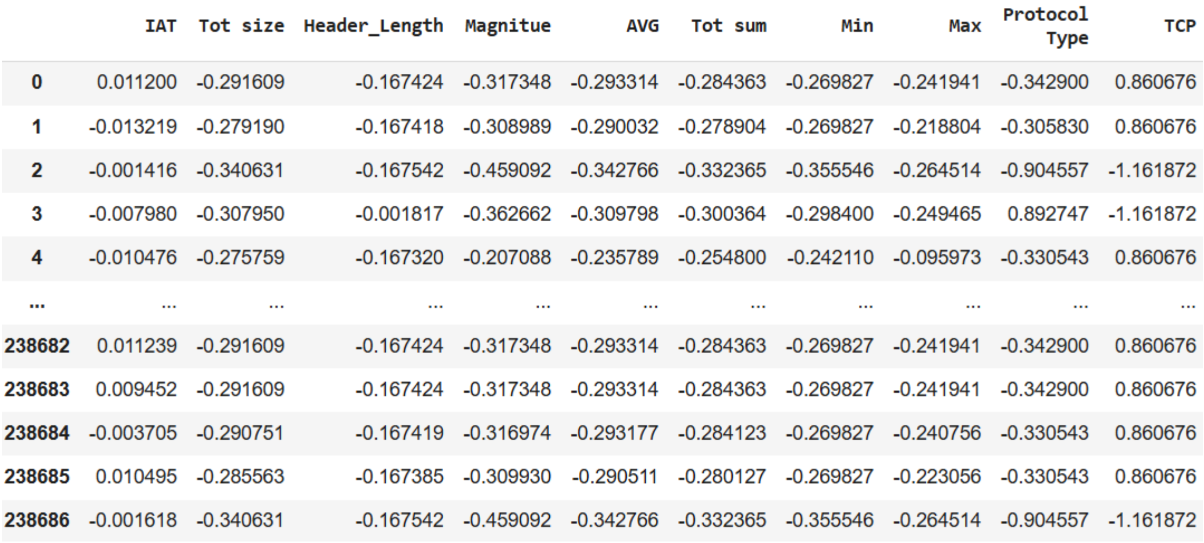


Fig. 5 Feature Importance



	IAT	Tot size	Header_Length	Magnitue	AVG	Tot sum	Min	Max	Protocol Type	TCP
0	0.011200	-0.291609	-0.167424	-0.317348	-0.293314	-0.284363	-0.269827	-0.241941	-0.342900	0.860676
1	-0.013219	-0.279190	-0.167418	-0.308989	-0.290032	-0.278904	-0.269827	-0.218804	-0.305830	0.860676
2	-0.001416	-0.340631	-0.167542	-0.459092	-0.342766	-0.332365	-0.355546	-0.264514	-0.904557	-1.161872
3	-0.007980	-0.307950	-0.001817	-0.362662	-0.309798	-0.300364	-0.298400	-0.249465	0.892747	-1.161872
4	-0.010476	-0.275759	-0.167320	-0.207088	-0.235789	-0.254800	-0.242110	-0.095973	-0.330543	0.860676
...	...	...	...	...	...	...	...	...	...	...
238682	0.011239	-0.291609	-0.167424	-0.317348	-0.293314	-0.284363	-0.269827	-0.241941	-0.342900	0.860676
238683	0.009452	-0.291609	-0.167424	-0.317348	-0.293314	-0.284363	-0.269827	-0.241941	-0.342900	0.860676
238684	-0.003705	-0.290751	-0.167419	-0.316974	-0.293177	-0.284123	-0.269827	-0.240756	-0.330543	0.860676
238685	0.010495	-0.285563	-0.167385	-0.309930	-0.290511	-0.280127	-0.269827	-0.223056	-0.330543	0.860676
238686	-0.001618	-0.340631	-0.167542	-0.459092	-0.342766	-0.332365	-0.355546	-0.264514	-0.904557	-1.161872

Fig. 6 Original Data after Feature Selection

2. Model Selection and Training

For model selection, various machine learning models are used, including Decision Trees (DT), Random Forest (RF), and XGBoost. These models are incorporated into an ensemble method called the Voting Classifier. The Voting Classifier aggregates the predictions of each individual model, using a soft voting mechanism where predictions are based on the probabilities assigned by each model. This combination aims to enhance overall accuracy by leveraging the strengths of multiple models. The models are trained on the training dataset and tested on the unseen testing dataset to evaluate their performance.

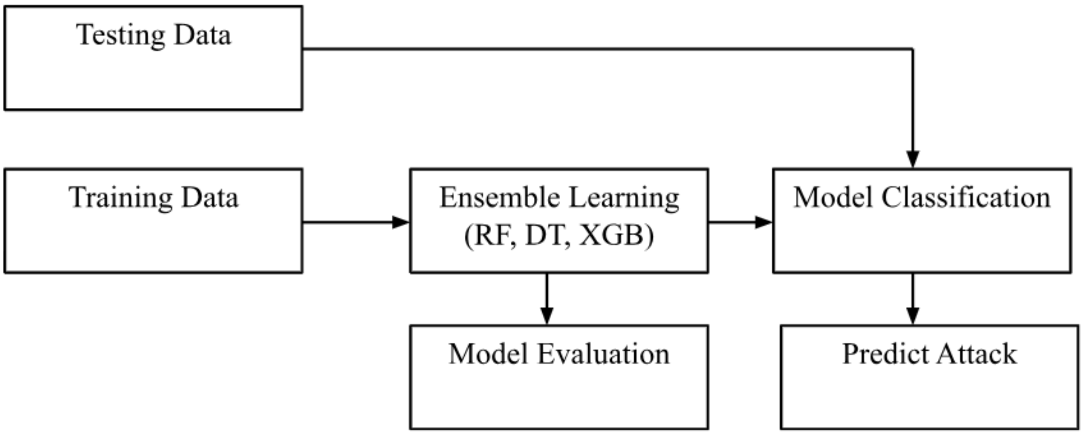


Fig. 7 Model Selection and Training

EVALUATION

4.1 Model Evaluation Metrics

To evaluate the effectiveness of the proposed models, various assessment metrics such as accuracy, precision, recall, and F1-score were employed. These metrics offer a quantitative measure of the classifier's performance and are derived from the confusion matrix, which highlights four key outcomes: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Specifically, TP refers to instances where the model correctly predicts a sample as class '0' and the actual label is also '0'. Conversely, TN indicates that the model correctly predicts a sample as class '1' when the actual label is indeed '1'. A false positive (FP) occurs when the model predicts a sample as '1' while the true label is '0', and a false negative (FN) arises when the model predicts '0' but the actual label is '1'.

The accuracy, precision, recall, and F1 measure are calculated using the following formulas:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

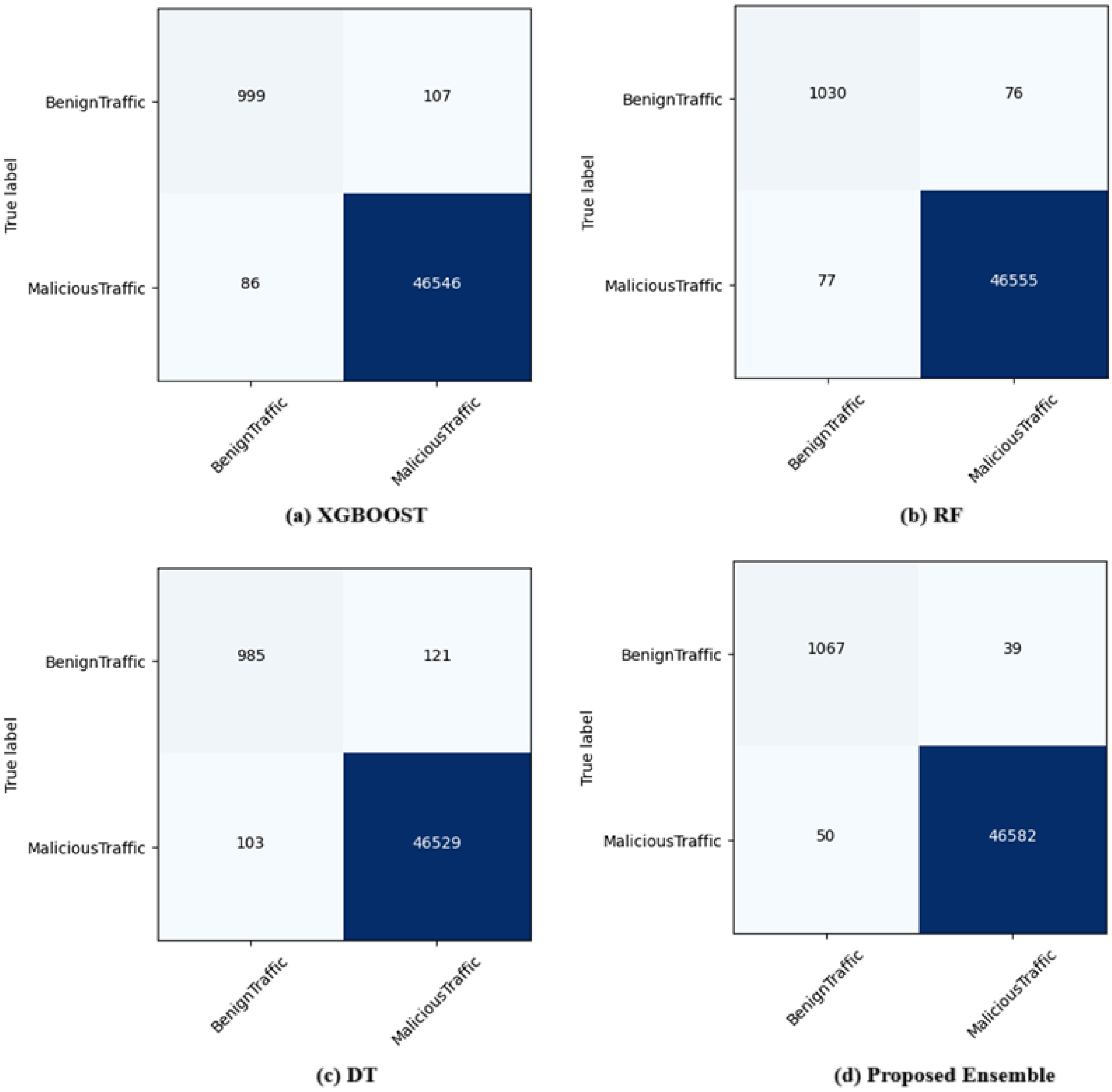
$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recollect} = \frac{TP}{TP + FN}$$

$$\text{F1 measure} = 2 \times \frac{Accuracy \times Recollect}{Accuracy + Recollect}$$

1) Binary Classification Results

In binary classification results, there are double sets of labels utilised, such as normal and malicious traffic, respectively. The confusion matrices illustrate in Figure 8 how well various models distinguish between normal and malicious network traffic using the CIC-IoT2023 dataset. Each model achieves strong results, with minimal errors in both categories. The ensemble model is shown best by achieving the highest correct prediction rates, suggesting it is highly reliable for intrusion detection tasks.



(a) XGBOOST

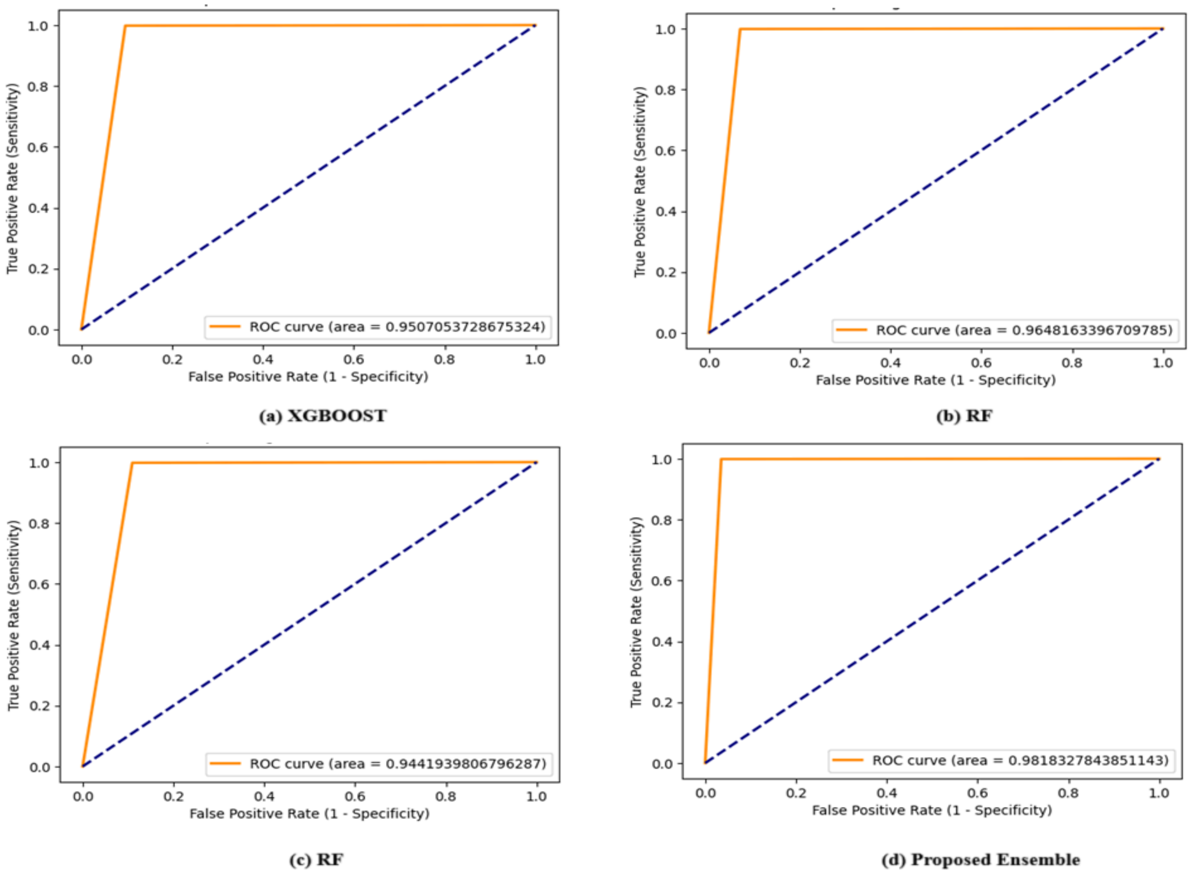
(b) RF

(c) DT

(d) Proposed Ensemble

Fig. 8 Confusion Matrix

A) Receiver Operating Characteristic ROC-AUC Analysis



(a) XGBOOST

(b) RF

(c) RF

(d) Proposed Ensemble

Fig. 9 ROC Curve

These Receiver Operating Characteristic (ROC) curves demonstrate the classification effectiveness of different models on the CIC-IoT2023 dataset. Each model shows strong performance with high AUC scores, indicating excellent sensitivity and specificity. The model in the bottom right achieves the highest AUC of 0.98, suggesting superior ability to distinguish between benign and malicious traffic

Table 1. Comparison table for the set1 performance metrics (in %) of each model for **binary classification**

Models	**Accuracy**	**Precision**	**Recall**	**F1-Score**
XGBoost	99.57%	99.55%	99.58%	99.56%
DT	99.69%	99.67%	99.70%	99.68%
RF	99.51%	99.49%	99.52%	99.50%
Proposed Ensemble	99.80%	99.70%	99.81%	99.80%

Table 2. Comparison table for the set 2 performance metrics (in %) of each model for binary classification

Model	TPR (%)	TNR (%)	MCC	NPV (%)	FDR (%)	FNR (%)	FOR (%)	FPR (%)
XGBoost	99.83	93.13	0.929	93.04	0.16	0.17	6.96	6.87
DT	99.78	89.06	0.896	90.53	0.26	0.22	9.47	10.94
RF	99.83	93.13	0.929	93.04	0.16	0.17	6.96	6.87
Proposed Ensemble	99.89	96.47	0.959	95.52	0.08	0.11	4.48	3.53

The evaluation results demonstrate that the proposed ensemble model outperforms individual classifiers such as XGBoost, Decision Tree, and Random Forest across key metrics. It achieves the highest accuracy, precision, and recall, along with the lowest error rates. Notably, the ensemble shows a superior True Negative Rate and a significantly lower False Positive Rate, indicating its robustness in distinguishing between normal and malicious traffic. These improvements suggest that the ensemble approach provides a more reliable and efficient solution for intrusion detection in IoT networks.

2) Multi-Class Classification Results

In multi-class classification results, multiple classes of labels are used to examine the proposed and existing works already done. Those labels include normal, DoS attacks, reconnaissance attacks, malware attacks, Botnet attacks, Brute force attacks, etc. The full clarification of the assessment results in multi-class classification problems is provided below.

Table 3. Comparison table for the performance metrics (in %) of each model for Multi-Class Classification

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
XGBoost	99.02	98.96	99.02	98.98
Random Forest	99.07	99.00	99.07	99.00
Decision Tree	99.11	99.12	99.11	99.11
Ensemble	**99.25**	**99.22**	**99.25**	**99.22**

The performance metrics highlight that all classifiers demonstrate strong accuracy, precision, recall, and F1-scores. Among them, the proposed ensemble model achieves the best results across all metrics, with an accuracy of 99.25% and an F1-score of 99.22%**,** outperforming individual models like XGBoost, Random Forest, and Decision Tree. This confirms the ensemble's ability to effectively combine multiple learners for enhanced prediction accuracy and robustness in intrusion detection scenarios.

The evaluation results indicate that the proposed ensemble model performs better than individual models such as XGBoost, Random Forest, and Decision Tree. In the binary classification task, it achieved the highest accuracy (99.80%), as well as top scores in precision, recall, and F1-score, with minimal error rates. The ROC-AUC score of 0.98 reflects strong classification capability.

In the multi-class classification, the ensemble model again delivered the best results, with 99.25% accuracy and consistently high performance across all metrics. This demonstrates that the ensemble method provides a more accurate and reliable solution for identifying and categorising network traffic.

CONCLUSION AND FUTURE WORK

This study evaluated multiple classification models for their ability to detect network intrusions using the CIC-IoT2023 dataset. Among the evaluated models, the proposed ensemble approach exhibited the highest effectiveness in both binary and multi-class classification scenarios. It consistently outperformed individual models, such as XGBoost, Decision Trees, and Random Forests, on key performance indicators like accuracy, precision, recall, and F1-score. The results demonstrate the ensemble method's reliability and strength in accurately identifying and classifying network traffic. In the future, this work can be expanded by implementing the system in real-time environments for continuous monitoring. Further improvements may also involve integrating more advanced learning techniques, refining the feature selection process, and validating the model on broader, more varied datasets to enhance its flexibility and applicability to different network conditions and threat types.

REFERENCES

[1] S. Kumar, P. Tiwari, and M. Zymbler, "Internet of Things is a revolutionary approach for future technology enhancement: A review," J. Big Data, vol. 6, p. 111, Dec. 2019.

[2] J. Shahid, R. Ahmad, A. K. Kiani, T. Ahmad, S. Saeed, and A. M. Almuhaideb, "Data protection and privacy of the Internet of Healthcare Things (IoHTs)," Appl. Sci., vol. 12, no. 4, p. 1927, Feb. 2022.

[3] R. Das and E. Ozdogan, "Layered management approach to cyber security of IoT solutions," Int. J. Grid Utility Comput., vol. 14, no. 5, pp. 493–504, 2023.

[4] R. Kumar and N. Agrawal, "Analysis of multi-dimensional industrial IoT (IIoT) data in edge–fog–cloud based architectural frameworks: A survey on current state and research challenges," J. Ind. Inf. Integr., vol. 35, Oct. 2023, Art. no. 100504.

[5] K. Shaukat, T. M. Alam, I. A. Hameed, W. A. Khan, N. Abbas, and S. Luo, "A review on security challenges in Internet of Things (IoT)," in Proc. 26th Int. Conf. Autom. Comput. (ICAC), Sep. 2021, pp. 1–6.

[6] L. L. Dhirani, E. Armstrong, and T. Newe, "Industrial IoT, cyber threats, and standards landscape: Evaluation and roadmap," Sensors, vol. 21, no. 11, p. 3901, Jun. 2021

[7] M. Serror, S. Hack, M. Henze, M. Schuba, and K. Wehrle, “Challenges and opportunities in securing the industrial Internet of Things,” IEEE Trans. Ind. Informat., vol. 17, no. 5, pp. 2985–2996, May 2021.

[8] R. D. McLeod, K. Ferens, and M. R. Friesen, “The IoT: Examples and trends,” in Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI), Dec. 2015, pp. 336–339.

[9] D. Stiawan, A. H. Abdullah, and M. Y. Idris, “Characterising network intrusion prevention system,” Int. J. Comput. Appl., vol. 14, no. 1, pp. 11–18, Jan. 2011.

[10] S. Mukkamala, A. H. Sung, and A. Abraham, “Intrusion detection using an ensemble of intelligent paradigms,” J. Netw. Comput. Appl., vol. 28, no. 2, pp. 167–182, Apr. 2005

[11] R. M. Gomathi, G. H. S. Krishna, E. Brumancia, and Y. M. Dhas, “A survey on IoT technologies, evolution and architecture,” in Proc. Int. Conf. Comput., Commun., Signal Process. (ICCCSP), Feb. 2018, pp. 1–5.

[12] B. Sharma, L. Sharma, and C. Lal, “Anomaly detection techniques using deep learning in IoT: A survey,” in Proc. Int. Conf. Comput. Intell. Knowl. Economy (ICCIKE), Dec. 2019, pp. 146–149, doi: 10.1109/ICCIKE47802.2019.9004362.

[13] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez, and E. V ´ azquez, “Anomaly-based network ´ intrusion detection: Techniques, systems and challenges,” computers & security, vol. 28, no. 1-2, pp. 18–28, 2009

[14] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, “Network intrusion detection system: A systematic study of machine learning and deep learning approaches,” Trans. Emerg. Telecommun. Technol., vol. 32, no. 1, Jan. 2021, Art. no. e4150

[15] R. Ahmad, I. Alsmadi, W. Alhamdani, and L. Tawalbeh, “Zero-day attack detection: A systematic literature review,” Artif. Intell. Rev., vol. 56, no. 10, pp. 10733–10811, Oct. 2023.

[16] S. M. Tahsien, H. Karimipour, and P. Spachos, “Machine learning based solutions for security of Internet of Things (IoT): A survey,” J. Netw. Comput. Appl., vol. 161, Jul. 2020, Art. no. 102630.

[17] M. M. Inuwa and R. Das, “A comparative analysis of various machine learning methods for anomaly detection in cyber attacks on IoT networks,” Internet Things, vol. 26, Jul. 2024, Art. no. 101162.

[18] S. Bacha, A. Aljuhani, K. B. Abdellafou, O. Taouali, N. Liouane, and M. Alazab, “Anomaly-based intrusion detection system in IoT using kernel extreme learning machine,” J. Ambient Intell. Humanized Comput., vol. 15, no. 1, pp. 231–242, Jan. 2024.

[19] Yali Yuan, LiuweiHuo and D. Hogrefe, "Two Layers Multi-class Detection method for network Intrusion Detection System," *2017 IEEE Symposium on Computers and Communications (ISCC)*, Heraklion, Greece, 2017, pp. 767-772, doi: 10.1109/ISCC.2017.8024620.

[20] F. Rehman, F. Mushtaq and H. Zaman, "A Host-based Intrusion Detection: Using Signature-based and AI-driven Anomaly Detection for Enhanced Cybersecurity*," *2024 4th International Conference on*

Digital Futures and Transformative Technologies (ICoDT2), Islamabad, Pakistan, 2024, pp. 1-7, doi: 10.1109/ICoDT262145.2024.10740248.

[21] R. Zhang, Y. Song and X. Wang, "Network Intrusion Detection Scheme Based on IPSO-SVM Algorithm," *2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, Dalian, China, 2022, pp. 1011-1014, doi: 10.1109/IPEC54454.2022.9777568.

[22] S. Zheng, "Network Intrusion Detection Model Based on Convolutional Neural Network," *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Chongqing, China, 2021, pp. 634-637, doi: 10.1109/IAEAC50856.2021.9390930.

[23] S. Saif et al., ‘‘HIIDS: Hybrid intelligent intrusion detection system empowered with machine learning and metaheuristic algorithms for application in IoT based healthcare,’’ Microprocess. Microsyst., 2022, Art. no. 104622.

[24] M. Ahmad, Q. Riaz, M. Zeeshan, H. Tahir, S. A. Haider, and M. S. Khan, ‘‘Intrusion detection in Internet of Things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set,’’ EURASIP J. Wireless Commun. Netw., vol. 2021, no. 1, pp. 1–23, Dec. 2021

[25] S. Yaras and M. Dener, ‘‘IoT-based intrusion detection system using new hybrid deep learning algorithm,’’ Electronics, vol. 13, no. 6, p. 1053, Mar. 2024.

[26] H. Mankodiya, M. S. Obaidat, R. Gupta, and S. Tanwar, ‘‘XAI-AV: Explainable artificial intelligence for trust management in autonomous vehicles,’’ in Proc. Int. Conf. Commun., Comput., Cybersecurity, Informat. (CCCI), Oct. 2021, pp. 1–5.

[27] F. Fleuret, "Fast Binary Feature Selection with Conditional Mutual Information," *Journal of Machine Learning Research*, vol. 5, pp. 1531–1555, 2004.

[28] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[29] Quinlan, J. R. (1986). *Induction of decision trees*. Machine Learning, 1(1), 81–106.

[30] Breiman, L. (2001). *Random forests*. Machine Learning, 45(1), 5–32.

[31] Chen, T., & Guestrin, C. (2016). *XGBoost: A scalable tree boosting system*. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794).

[32] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, A. A. Ghorbani. "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," Sensor (2023) – (submitted to Journal of Sensors).