

An Optimized Machine Learning Framework for Multiclass IoT Attack Detection with Emphasis on Spoofing Attacks

Kukkala Durga Prameela¹, Y. Sudheer Kumar²

¹M. Tech, Department of CS & SE, Andhra University College of Engineering (A), Visakhapatnam, India

²Research Scholar, Department of CS & SE, Andhra University College of Engineering (A),
Visakhapatnam, India

Abstract—The rapid growth of Internet of Things (IoT) networks has increased their exposure to cyberattacks, particularly spoofing attacks that exploit device and communication identities. Machine-learning-based intrusion detection can identify malicious traffic, however, class imbalance and high-dimensional features can limit minority-class detection. This paper proposes a class-aware machine-learning framework for multiclass IoT attack detection using the CIC IoT 2023 dataset. The framework performs preprocessing, label encoding, stratified train-test partitioning, and feature selection using Random Forest feature importance and Mutual Information. Complete and selected feature sets are evaluated using Random Forest, XGBoost, LightGBM, and CatBoost, with XGBoost selected for further analysis. A class-aware training strategy limits highly represented classes while retaining available minority-class samples. The final model is evaluated using classification metrics, confusion-matrix and per-class analysis, with specific evaluation of DNS Spoofing and MITM-ArpSpoofing. The results show that the class-aware strategy improves minority-class detection while maintaining strong overall multiclass performance.

Index Terms—Internet of Things (IoT), Intrusion Detection System (IDS), Spoofing Attack Detection, CIC IoT 2023, Machine Learning, XGBoost, Feature Selection, Class Imbalance.

I. INTRODUCTION

The Internet of Things (IoT) has become an important part of modern computing infrastructure by connecting heterogeneous devices and enabling continuous communication among devices, networks, and cloud services. However, the growing scale and diversity of IoT environments introduce significant

security challenges. IoT devices and communication channels are vulnerable to cyberattacks that can affect confidentiality, integrity, availability, and network trust. Among these threats, spoofing attacks are particularly important because attackers can impersonate legitimate devices or network entities by manipulating identity-related information.

DNS spoofing and Address Resolution Protocol (ARP) spoofing are examples in which malicious traffic may appear to originate from trusted sources, making reliable identification of the actual source difficult.

Machine-learning based intrusion detection systems (IDSs) can identify malicious traffic by learning patterns from network-flow characteristics. Previous studies have evaluated algorithms such as Naive Bayes, Logistic Regression, Support Vector Machine, and Random Forest for IoT spoofing detection. One such study reported Random Forest as the best-performing model with approximately 98% accuracy [1]. However, feature-selection methods and class-aware training for highly imbalanced multiclass attack detection were not investigated.

IoT traffic datasets often exhibit significant class imbalance, where some attack categories contain substantially fewer samples than dominant classes. Therefore, high overall accuracy may not indicate reliable minority-class detection. Class-sensitive metrics and detailed per-class analysis are consequently important for evaluating IoT intrusion detection systems.

This study proposes a class-aware machine-learning framework for multiclass IoT cyberattack detection using the CIC IoT 2023 dataset. The framework

integrates preprocessing, stratified data partitioning, feature-selection analysis, comparative ensemble-model evaluation, and class-aware training. Random Forest feature importance and Mutual Information are used for feature selection, while Random Forest, XGBoost, LightGBM, and CatBoost are comparatively evaluated. XGBoost is subsequently selected for class-aware training, where highly represented classes are limited while available minority-class samples are retained.

The final model is evaluated using accuracy, precision, recall, weighted F1-score, macro F1-score, MCC, confusion-matrix analysis, and spoofing-specific analysis of DNS Spoofing and MITM-Arpspoofing.

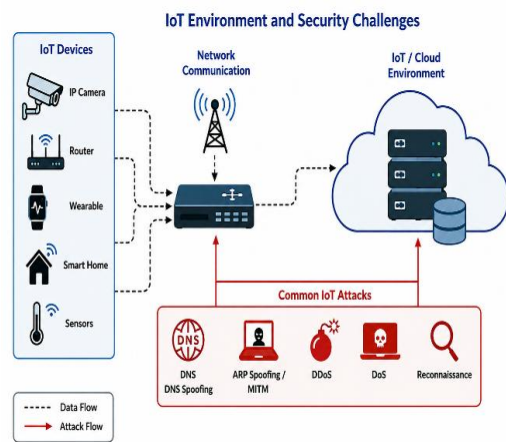


Fig 1. Security Challenges and Spoofing threats in IoT Networks

A. Background and Motivation

The effectiveness of an IoT intrusion detection system depends not only on overall classification performance but also on its ability to detect under represented attack classes. Severe class imbalance can cause machine-learning models to favor dominant classes, resulting in poor detection of minority attacks. Spoofing attacks are particularly important because they involve impersonation of legitimate network entities. This motivates the investigation of feature selection and class-aware training to improve multiclass IoT attack detection.

B. Problem Statement

Existing machine-learning-based IoT intrusion detection approaches may achieve high overall

performance while providing limited detection of minority attack categories. In addition, limited investigation has been performed on the combined effect of feature selection, ensemble-model comparison, and class-aware training for multiclass IoT attack detection. Therefore, this work addresses the problem of developing a machine-learning framework that can maintain strong overall detection while improving the representation and identification of minority attack classes, particularly DNS Spoofing and MITM-Arpspoofing.

Main Contributions

The main contributions of this work are:

- A multiclass IoT attack-detection framework is developed using the CIC IoT 2023 dataset with 46 numerical features and 34 attack classes.
- Random Forest feature importance and Mutual Information are investigated for feature selection and compared with the complete feature set.
- Random Forest, XGBoost, LightGBM, and CatBoost are comparatively evaluated.
- A class-aware training strategy is introduced to reduce the dominance of highly represented classes while retaining minority-class samples.
- Class-Aware XGBoost is evaluated using overall and class-sensitive metrics, with emphasis on DNS Spoofing and MITM-Arpspoofing.
- Confusion-matrix and spoofing misclassification analyses are performed to identify difficult attack categories.

II. RELATED WORK

Machine-learning-based intrusion detection has been widely studied for identifying malicious activities in IoT environments. Existing studies have explored different attack types, datasets, and learning algorithms; however, their effectiveness depends on dataset characteristics, feature selection, and minority-class detection.

Aldabbas et al. [1] proposed an SDN-based approach for ARP spoofing detection in IoT networks by analyzing topology information and ARP queries to improve attack identification and mitigation.

Aldosari [2] investigated location spoofing using LSTM and MLP models for spoofing detection and estimation of the time of arrival of spoofed signals.

Kumar et al. [3] developed EDIMA, a machine-learning-based botnet detection system for IoT edge gateways that combined supervised traffic analysis with autocorrelation-based bot identification.

Venu et al. [4] addressed poor cross-dataset generalization in botnet detection by proposing a consistent feature set for machine-learning models.

Tuan et al. [5] evaluated Naive Bayes, Decision Tree, Artificial Neural Network, and Support Vector Machine for botnet-based DDoS detection using the UNSW-NB15 and KDD99 datasets.

Sharma and Babbar [6] investigated IoT spoofing detection using Random Forest, Naive Bayes, Logistic Regression, and Support Vector Machine. Random Forest achieved the highest reported accuracy of 98%. Overall, these studies demonstrate the applicability of machine learning to IoT intrusion, botnet, and spoofing detection. However, further investigation is needed into feature selection, ensemble-model comparison, class imbalance, and spoofing-specific analysis. This work addresses these aspects using the CIC IoT 2023 dataset, Random Forest feature importance, Mutual Information, ensemble-model comparison, and class-aware training, with specific analysis of DNS_Spoofing and MITM-ArpSpoofing.

A. Research Gaps

The reviewed literature reveals the following gaps:

- **Feature-selection gap:** The reviewed studies do not explicitly compare Random Forest feature importance and Mutual Information for multiclass IoT attack detection.
- **Model-comparison gap:** The base study [6] evaluates Random Forest, Naive Bayes, Logistic Regression, and Support Vector Machine, whereas this study compares Random Forest, XGBoost, LightGBM, and CatBoost.
- **Class-imbalance gap:** Class-aware training for highly imbalanced multiclass attack categories is not explicitly investigated.
- **Multiclass evaluation gap:** Greater emphasis is placed on weighted and macro metrics, MCC, and class-level performance rather than overall accuracy alone.

- **Spoofing-specific gap:** DNS_Spoofing and MITM-ArpSpoofing are separately analyzed using classification and misclassification patterns.

B. Research Objectives

The main objective is to develop a class-aware machine-learning framework for multiclass IoT attack detection, with emphasis on spoofing attacks and imbalanced classes. The specific objectives are:

- To develop a multiclass IoT attack-detection framework using the CIC IoT 2023 dataset.
- To identify informative features using Random Forest feature importance and Mutual Information and compare them with the complete feature set.
- To compare Random Forest, XGBoost, LightGBM, and CatBoost for multiclass attack detection.
- To investigate the effect of feature reduction on classification performance and training cost.
- To develop a class-aware training strategy that reduces majority-class dominance while retaining minority-class samples.
- To evaluate Class-Aware XGBoost using accuracy, precision, recall, weighted F1, macro F1, and MCC.
- To analyze DNS_Spoofing and MITM-ArpSpoofing using class-level metrics, confusion-matrix analysis, and misclassification patterns.

III. METHODOLOGY

The methodology used to develop and evaluate the proposed class-aware multiclass IoT intrusion detection framework. The methodology consists of dataset preparation, preprocessing, stratified data partitioning, feature selection, feature-set evaluation, ensemble-model comparison, class-aware training, and final evaluation.

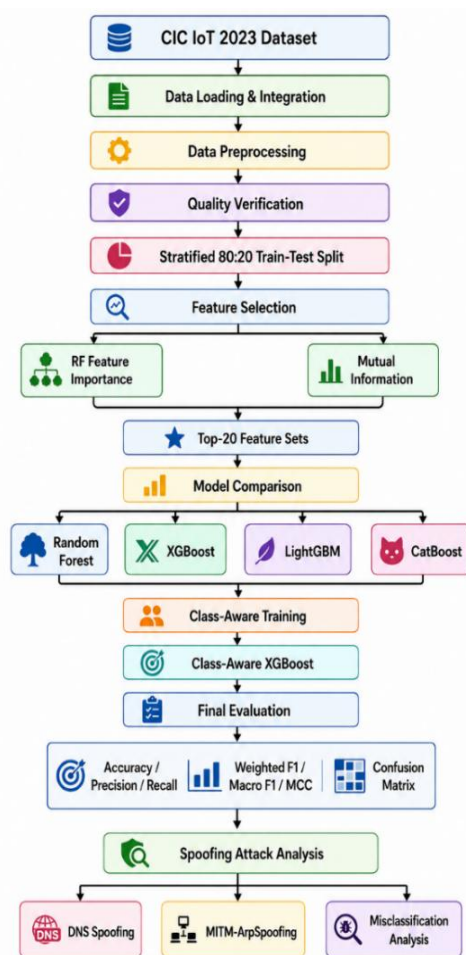


Fig 2. Proposed system architecture for multiclass IoT attack detection

Step 1: Dataset Preparation

The CIC IoT 2023 dataset [7] is used for the experiments. The integrated dataset contains 3,380,000 records, 46 numerical network-flow features, and 34 attack classes. The attack categories include benign traffic, DDoS, DoS, Mirai, reconnaissance, vulnerability scanning, and spoofing attacks.

Step 2: Data Preprocessing

The `source_file` attribute is removed because it identifies the originating file rather than traffic behavior. The label attribute is separated as the target variable and encoded into numerical class identifiers using Label Encoding. The resulting feature matrix contains 46 input features and the target contains 34 classes. Missing values and duplicate records are checked during preprocessing.

Step 3: Stratified Train-Test Partitioning

The dataset is divided into training and testing sets using an 80:20 stratified split with `random_state = 42`. Stratification preserves the relative representation of all 34 classes in both subsets. The test set remains unchanged throughout the experiments to provide consistent evaluation.

Dataset	Samples	Features
Training	2,704,000	46
Testing	676,000	46
Total	3,380,000	46

Step 4: Feature Selection

Two feature-selection methods are independently applied to identify informative network-flow features:

A. Random Forest Feature Importance

Random Forest feature importance [9] is used to rank the input features according to their contribution to multiclass classification. A Random Forest model is trained on a stratified subset of 100,000 training samples. Features are ranked according to their importance scores, and the top 20 features are selected. Top 20 RF importance Feature: IAT, Tot size, syn_count, Min, Magnitude, Protocol Type, Header Length, AVG, fin_count, ack_flag_number, psh_flag_number, UDP, fin_flag_number, syn_flag_number, rst_count, flow_duration, rst_flag_number, Tot sum, urg_count, Max

B. Mutual Information

Mutual Information is used as a complementary feature-selection method to measure the dependency between individual features and the target classes. Mutual Information is calculated using the same 100,000-sample subset to measure the relationship between each feature and the attack label. The top 20 features are selected based on their scores.

The two selected feature sets are compared to identify common and method-specific features.

Top 20 MI Features: IAT, Tot size, Header_Length, Magnitude, AVG, Tot sum, Min, Max, Protocol Type, TCP, syn_count, flow_duration, Srate, Rate, rst_count, syn_flag_number, UDP, urg_count, ICMP, Covariance

Step 5: Feature-Set Evaluation

Three feature configurations are considered:

- Complete set of 46 features

- Random Forest Top-20 features
- Mutual Information Top-20 features

A controlled 100,000-sample training subset is used for this experiment. XGBoost is applied to the three configurations under the same experimental conditions to examine the trade-off between predictive performance and feature reduction. Based on this comparison, the feature representation for subsequent experiments is selected.

Feature-Set Comparison RF+MI: AVG, Header_Length, IAT, Magnitude, Max, Min, Protocol Type, Tot size, Tot sum, UDP, flow_duration, rst_count, syn_count, syn_flag_number, urg_count

Step 6: Comparative Model Evaluation

Four ensemble-learning models XGBoost [8], Random Forest [9], LightGBM, and CatBoost are comparatively evaluated under a common experimental setup. Their performance is compared using accuracy, precision, recall, weighted F1-score, macro F1-score, and MCC. The best-performing model is selected for the final class-aware experiment.

Step 7: Class-Aware Training

Based on the comparative evaluation, XGBoost was selected for class-aware training. To reduce the dominance of highly represented classes, classes exceeding 20,000 samples were limited to 20,000 samples, while classes below this threshold retained all available samples.

This produced a training set of 391,636 samples across 34 classes, while the original 676,000-sample test set was retained without sampling.

Step 8: Final Class-Aware XGBoost

Based on the comparative evaluation, XGBoost is selected for the final experiment. The model is trained using the class-aware training dataset and the selected feature representation, and is evaluated on the unchanged test set.

Step 9: Performance Evaluation

The final model is evaluated using accuracy, precision, recall, weighted F1-score, macro F1-score, and Matthews Correlation Coefficient (MCC). A 34×34 confusion matrix is generated for class-level analysis. Particular attention is given to DNS_Spoofing and MITM-ArpSpoofing, including their class-level performance and major misclassification patterns.

IV. EXPERIMENTAL RESULTS

This section evaluates the proposed framework in four stages: feature-selection analysis, comparative model evaluation, class-aware training, and spoofing-specific analysis.

A. Feature-Selection Results

Random Forest feature importance and Mutual Information were applied to the same 100,000-sample subset. Each method selected the top 20 features from the original 46 features, with 15 common features (75% overlap).

RF Top 20: 20 features, MI Top 20: 20 features

Common: 15 features

RF-only: 5 features, MI-only: 5 features

Feature overlap: 75%

The feature sets were evaluated using XGBoost under the same experimental conditions. The complete 46-feature representation achieved the highest Accuracy (0.9905), Macro F1 (0.7034), and MCC (0.9896). Although the reduced feature sets produced slightly lower performance, RF Top 20 reduced training cost by 37.30%.

Table I. Comparison of classification performance for the evaluated feature sets

Feature Set	Accuracy	Weighted F1	Macro F1	MCC
All 46	0.99	0.98	0.70	0.98
RF Top 20	0.99	0.98	0.69	0.98
MI Top 20	0.98	0.98	0.69	0.98

These results indicate that feature reduction can provide computational benefits, but retaining all 46 features provides the best overall classification performance. Therefore, the complete feature set was retained for the final model.

B. Comparative Model Evaluation

Random Forest, XGBoost, LightGBM, and CatBoost were evaluated using the same 100,000 training samples, 46 features, and 676,000 test samples.

XGBoost achieved the best performance with 99.05% Accuracy, 98.97% Weighted F1, 70.34% Macro F1, and 98.96% MCC. It was therefore selected for the subsequent class-aware training experiment.

Table II. Comparative performance of the evaluated machine-learning models.

Model	Accuracy	Weighted F1	Macro F1	MCC
LightGBM	0.50	0.51	0.28	0.48
Random Forest	0.97	0.97	0.59	0.97
CatBoost	0.98	0.98	0.64	0.98
XGBoost	0.99	0.98	0.70	0.98

XGBoost achieved the highest performance across all reported metrics and was therefore selected as the base classifier for the class-aware training experiment.

Accuracy Comparison of Machine-Learning Models

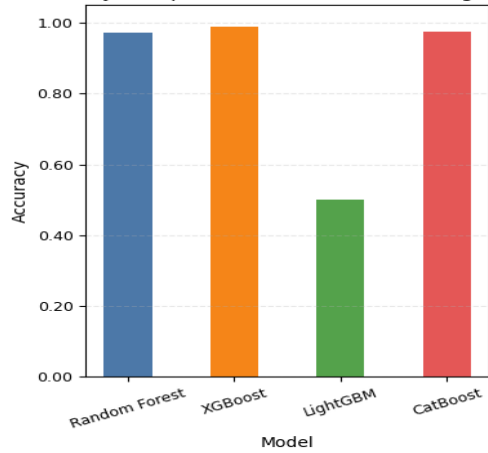


Fig 3. Comparative Accuracy of RF, XGBoost, LightGBM, and CatBoost

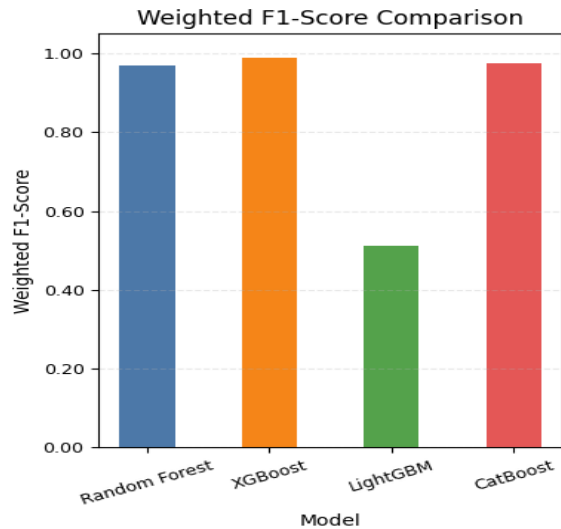


Fig. 4 Comparative Weighted F1-score of RF, XGBoost, LightGBM, and CatBoost

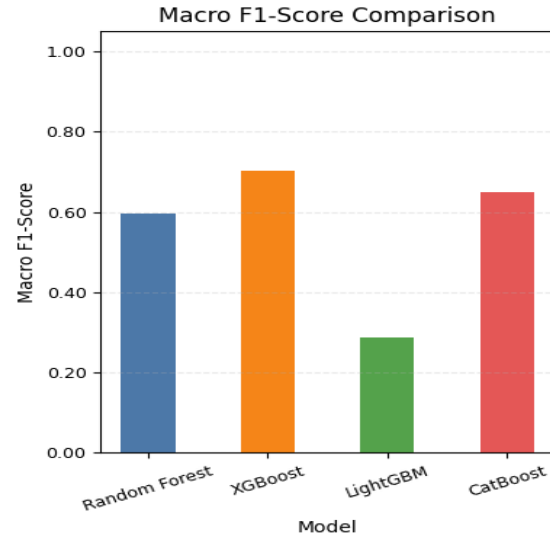


Fig.5 Comparative Macro F1-score of RF, XGBoost, LightGBM, and CatBoost

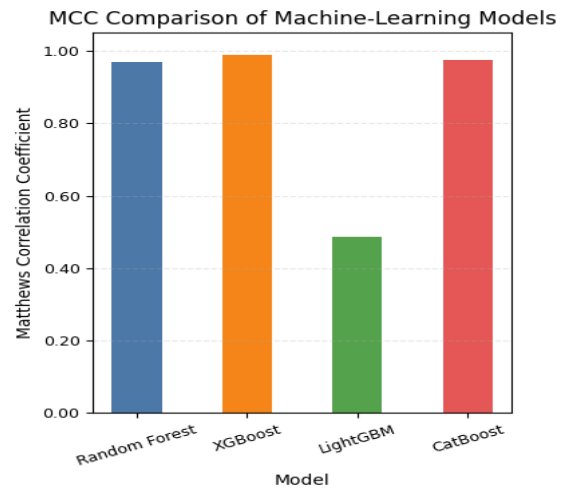


Fig. 6 Comparative MCC of RF, XGBoost, LightGBM, and CatBoost

C. Class-Aware Training Results

To reduce the dominance of highly represented classes, classes exceeding 20,000 samples were limited to 20,000, while smaller classes retained all available samples. This produced 391,636 training samples across 34 classes, while the original 676,000-sample test set was retained.

Compared with Natural XGBoost, Class-Aware XGBoost improved Accuracy from 0.9905 to 0.9933 and Macro F1 from 0.7034 to 0.7740. Macro Precision increased from 0.7579 to 0.8398, Macro Recall from 0.6963 to 0.7615, and MCC from 0.9896 to 0.9926.

Table III. Performance comparison of Natural and Class-Aware XGBoost.

Metric	Natural XGBoost	Class-Aware XGBoost
Accuracy	0.9905	0.9933
Weighted Precision	0.9896	0.9932
Weighted Recall	0.9905	0.9933
Weighted F1	0.9897	0.9931
Macro Precision	0.7579	0.8398
Macro Recall	0.6963	0.7615
Macro F1	0.7034	0.7740
MCC	0.9896	0.9926

The improvement in macro-level metrics indicates better overall class-level performance under the imbalanced distribution.

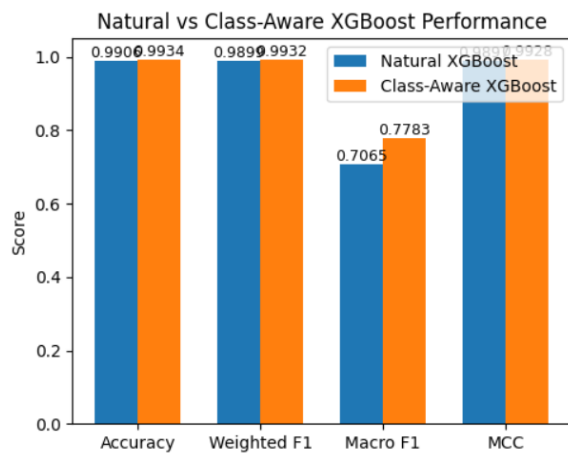


Fig 7 Performance comparison of Natural and Class-Aware XGBoost

D. Spoofing Attack Detection

The final Class-Aware XGBoost model was evaluated for DNS_Spoofing and MITM-ArpSpoofing.

Table IV. Class-level performance for spoofing attacks

Attack Type	Precision	Recall	F1-Score
DNS_Spoofing	0.6701	0.7624	0.7133
MITM-ArpSpoofing	0.8313	0.8512	0.8411

MITM-ArpSpoofing achieved higher precision, recall, and F1-score than DNS_Spoofing, indicating different classification difficulty between the two spoofing categories.

E. Confusion-Matrix Analysis

A 34×34 confusion matrix was used to analyze class-level predictions. Strong diagonal concentration was

observed for many classes, while off-diagonal predictions indicated more difficult categories.

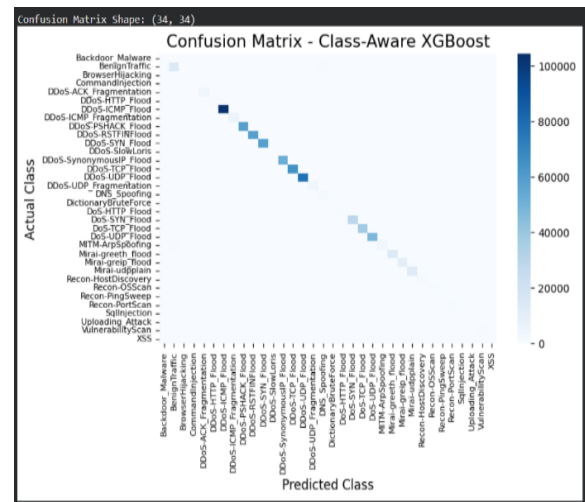


Fig 8. Confusion matrix of the final Class-Aware XGBoost model

Table V. Major spoofing-related misclassifications

Actual Class	Predicted Class	Samples
MITM-ArpSpoofing	BenignTraffic	419
DNS_Spoofing	BenignTraffic	247
DNS_Spoofing	MITM-ArpSpoofing	217
MITM-ArpSpoofing	DNS_Spoofing	171
DNS_Spoofing	Recon-HostDiscovery	116
MITM-ArpSpoofing	Recon-HostDiscovery	44

Major spoofing-related errors included MITM-ArpSpoofing misclassified as BenignTraffic (419 samples) and DNS_Spoofing as BenignTraffic (247 samples). Confusion between the two spoofing categories was also observed, highlighting the remaining challenges in spoofing-specific multiclass detection.

V. CONCLUSION

This study presented a class-aware machine-learning framework for multiclass IoT attack detection using the CIC IoT 2023 dataset. The framework combined data preprocessing, feature-selection analysis, comparative ensemble-model evaluation, and class-aware training to address the challenges of highly imbalanced IoT traffic.

Random Forest feature importance and Mutual Information were used to identify informative features, while Random Forest, XGBoost, LightGBM,

and CatBoost were comparatively evaluated. XGBoost achieved the best overall performance and was therefore selected for class-aware training. The proposed class-aware strategy reduced the dominance of highly represented classes while retaining available minority-class samples.

The final Class-Aware XGBoost model achieved 99.33% accuracy, 99.31% weighted F1-score, 77.40% Macro F1-score, and 99.26% MCC. Macro F1 improved from 70.34% to 77.40% compared with the natural XGBoost training approach, indicating improved class-level performance under class imbalance. Further analysis of DNS_Spoofing and MITM-ArpSpoofing demonstrated the model's ability to detect spoofing attacks while highlighting remaining classification challenges through confusion-matrix analysis.

Overall, the proposed framework provides an effective approach for multiclass IoT attack detection with emphasis on minority-class performance. Future work will focus on hyperparameter optimization, real-time or edge-based deployment, and improved detection of challenging minority attack classes.

REFERENCES

- [1] H. Aldabbas and R. Amin, "A novel mechanism to handle address spoofing attacks in SDN based IoT," *Cluster Computing*, vol. 24, no. 4, pp. 3011–3026, 2021, doi: 10.1007/s10586-021-03309-0.
- [2] W. Aldosari, "Deep learning-based location spoofing attack detection and time-of-arrival estimation through power received in IoT networks," *Sensors*, vol. 23, no. 23, Art. no. 9606, 2023, doi: 10.3390/s23239606.
- [3] Kumar, M. Shridhar, S. Swaminathan, and T. J. Lim, "Machine learning-based early detection of IoT botnets using network-edge traffic," *Computers & Security*, vol. 117, Art. no. 102693, 2022, doi: 10.1016/j.cose.2022.102693.
- [4] N. Venu, A. Kumar, and A. S. Rao, "Botnet attacks detection in Internet of Things using machine learning," *NeuroQuantology*, vol. 20, no. 4, pp. 743–754, 2022, doi: 10.14704/NQ.2022.20.4. NQ22298.
- [5] T. A. Tuan, H. V. Long, L. H. Son, R. Kumar, I. Priyadarshini, and N. T. K. Son, "Performance evaluation of Botnet DDoS attack detection using machine learning," *Evolutionary Intelligence*, vol. 13, no. 2, pp. 283–294, 2020, doi: 10.1007/s12065-019-00310-w.
- [6] Sharma and H. Babbar, "Preventing spoofing threats in IoT: Machine learning approaches for intrusion detection," in *Proc. 2024 IEEE 3rd World Conf. Applied Intelligence and Computing (AIC)*, 2024, doi: 10.1109/AIC61668.2024.10730888.
- [7] C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," *Sensors*, vol. 23, no. 13, Art. no. 5941, 2023, doi: 10.3390/s23135941.
- [8] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.
- [9] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.