Advances in Artificial Intelligence and Machine Learning; Research 6 (1) 4891-4907 Received 5-10-2025; Accepted 17-01-2026; Published online 24-01-2026

Enhancing IoT Security Using Ensemble Based Feature Selection

Tahani Alkhudaydi talkhudaydi@ut.edu.sa
Department of Computer Science
Faculty of Computers and Information Technology
University of Tabuk
Tabuk 47512, Saudi Arabia

Wedad O. Alahamade woahamade@taibahu.edu.sa
Department of Computer Science and Information
Applied College, Taibah University
Madinah 42353, Saudi Arabia

Corresponding Author: Tahani Alkhudaydi

Copyright © 2026 Tahani Alkhudaydi and Wedad O. Alahamade. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

The massive heterogeneity of Internet-of-Things (IoT) traffic makes intrusion-detection systems (IDS) especially vulnerable to the severe class imbalance that dominates real world attack logs. Using the recent CICIoT2023 benchmark, which captures 47 flow features for 33 attack types launched by 105 physical devices, we first show that two widely adopted baselines (XGBoost and RF trained on the full data) attain high overall accuracy over 99% yet fail on minority attacks. This failure drives the macro-averaged F1 score down to 79 %. To remedy this, we introduce the Attack-aware Feature Aggregation Model (AFAM). AFAM (i) partitions the training data by high-level attack domain, (ii) applies homogeneous feature selection within each partition with RF and XGBoost, and (iii) aggregates the per-domain rankings via a maximum operator before retraining a global classifier. With the top-30 aggregated features and an XGBoost decision engine, AFAM boosts macro precision/recall/F1 to $96.4\%, 87.3\%, 90.8\%$, while preserving $\geq$ 99.9 % accuracy on majority classes. Minority classes benefit most: F1 rises from 11.4 % to 67.2 % on Uploading_Attack and from 19.8 % to 71.4 % on Recon-PingSweep. A feature-importance analysis further reveals how imbalance skews conventional rankings and how AFAM recovers the critical predictors for rare threats. These results show that aggregating attack based features can lead to improved performance in heavily imbalanced IoT datasets. Therefore, the proposed method aids to build more resilient IDS deployments.

Keywords: Intrusion detection systems, IDS, CICIoT 2023, IoT, Machine learning, Feature selection.

Citation: Tahani Alkhudaydi and Wedad O. Alahamade. Enhancing IoT Security Using Ensemble Based Feature Selection. **Advances in Artificial Intelligence and Machine Learning.** 2026;6(1):271. https://dx.doi.org/10.54364/AAIML.2026.61271

1. INTRODUCTION

The Internet of Things (IoT) is transforming many sectors such as healthcare, industrial systems, multimedia, and consumer markets. It utilizes advanced technologies such as 6G networks to build intelligent, and autonomous systems [1]. Theses advances have led to improvements in efficiency, monitoring, and data analysis in these sectors [1–5]. By 2030, the number of IoT devices is estimated to grow from 15.1 billion in 2020 to 29 billion, with China expected to lead with 8 billion devices. Over the next decade, the consumer segment is projected to maintain about 60% of all IoT devices in 2020[6].

Nonetheless, despite these advances, there are many concerns about the security of IoT systems. This is due to the wide diversity of IoT devices, protocols, and platforms, where it is hard to design unified security standards. Thus, it is crucial to design tailored intrusion detection systems (IDS) to detect and eliminate any unauthorized access.

Historically, Anderson's 1972 report established the intrusion detection research by stating the need for automated security breach detection, and it later reached its peak in Denning and Neumann's first real time IDS in 1985 [7]. In modern systems, IDS systems consist of three key components: data collection (system calls, network traffic), feature selection, and a decision engine for classifying threats [8].

The aim of the feature selection phase is to remove redundant or irrelevant attributes to reduce the number of dimensions in the captured dataset. Therefore, the computational overhead will be minimized and classifier accuracy will be increased. Then, the decision engine classify network activity using one of three methods: misuse detection,anomaly detection and hybrid models. The first matches traffic against known attack patterns. The second constructs models to distinguish between normal and harmful behavior. Lastly, hybrid models combine both strategies where they employ supervised classifiers to enhance detection performance while managing false positives. [8].

A robust IDS must depend on high quality datasets that capture balanced variety and complexity of network attacks. This will lead the models to learn and adapt to evolving threats. However, most IoT attack datasets are skewed and lack coverage of diverse attack types and realistic device topologies. Therefore, IDS models will lack generalization where they will favor common attacks and miss infrequent ones. Thus, it is important to develop diverse dataset that capture real IoT devices attacks to build more effective intrusion detection [9].

CICIoT2023 [9] is one of the most comprehensive IoT attack corpora. It was built from executing 33 distinct exploits against 105 real devices and organized into seven classes. It introduces novel attack types not present in prior IoT datasets but has highly skewed class distribution that might results in training biased models that overfit frequent and miss infrequent but critical attacks. Moreover, this imbalance can also distort feature selection phase and causes IDS to miss indicator of low occurrence attacks. Therefore, correcting these biases is essential to ensure accurate, comprehensive threat analysis on real world IoT deployments.

The existing research [10–14] on the CICIoT2023 dataset shows significant advancements in intrusion detection systems for IoT networks. However, further investigation is needed to enhance the

performance of these systems. Studies should focus on utilizing the entire dataset, including all 33 attack classes, and exploring techniques to handle imbalanced data within deep learning models.

Although benchmarks are important, it is also vital to build and deploy secure IoT systems with comprehensive and robust protection on real world environments such as smart campuses and industrial infrastructures. These systems should detect infrequent and rare attacks that may lead to high risk threat if left undetected.

In this paper, we address these issues by developing a more comprehensive and efficient intrusion detection framework for IoT networks. We explore the impact of unbalanced data on threat landscape analysis and propose a multi-step homogeneous ensemble pipeline for feature selection and classification on multi-attack datasets to mitigate class bias and improve detection accuracy. The main contributions of this research are as follows:

- Conduct a comprehensive analysis of the CICIoT2023 dataset and propose an Attack-aware Feature Aggregation Model (AFAM) that directly addresses severe class imbalance in IoT intrusion detection.
- Demonstrate that AFAM substantially improves macro-averaged precision, recall, and F1-score for minority attack classes while maintaining or slightly enhancing performance on majority classes, which indicates balanced gains without the typical accuracy trade-off.
- Reveal how class imbalance distorts per-class feature importance and show that AFAM's aggregation strategy recovers the critical features needed to recognize rare attacks.

2. RELATED WORK

Nowadays, IDS are significant for enhancing the security and protecting IoT networks from unauthorised access and security threats. On the other hand, machine learning (ML) techniques have been heavily involved in IDS to detect an anomaly or an intrusion within large, or even real time data. ML-based IDS for IoT networks typically involve data collection, feature extraction, model training, anomaly detection, real time monitoring, adaptive learning, and alerting and/or response mechanisms [15].

Apostol et al. [16] developed an unsupervised deep learning model to detect anomalies in IoT botnets, while Kumar et al. [17], developed a machine learning-based early detection system for IoT botnets using network-edge traffic. Churcher et al. [18] classified attacks in IoT networks using machine learning, and Qaddoura et al. [19] proposed a multi-stage classification technique based on clustering for IoT intrusion detection. Alkadi et al. [20] built a collaborative intrusion detection system for cloud and IoT networks that is supported by a deep blockchain framework. The efficacy of these models is mainly affected by the distribution of classes within the dataset [21]. The incorporation of ML approaches into IDS for IoT networks enhances the ability to proactively identify and respond to security threats, hence contributing to the overall security and resilience of IoT systems.

Several datasets for IoT attacks exist, including Bot-IoT [22], IoT-23 [23], N-BaIoT [24], CICIDS2017 [25], and CICIoT2023 [9]. These datasets contain various attacks that affect IoT net-

works and can be used with ML to build a strong detection system. While these datasets have been widely utilized in research and considered as reliable resources for testing IoT-based intrusion detection systems, they do have significant drawbacks. The drawbacks include: a lack of diversity of attack types, limited sample size, and other limitation during data collection such as not considering the diversity of network topology of real time IoT devices.

One of the largest datasets is the CICIoT2023 [9], which is a relatively recent dataset in 2023, in terms of both the number of devices engaged in constructing the network topology and the number of attacks. This dataset covers a wide scope of specific and general cyber threats. Researchers in [9], also conducted experiments using different machine learning models, that are logistic regression, AdaBoost, Deep Neural Network (DNN), and Random Forest (RF) with different classification scenarios. They found that RF achieved the highest performance among these models. The paper primarily emphasizes the analysis of the dataset rather than the classification system. Unlike other existing IoT security datasets, the CICIoT2023 dataset [9] is unique and has several features that make it better such as: comprehensive and extensive topology, realistic attack scenarios, availability, accessibility, and data formats making it a valuable resource for developing and testing new security analytics solutions for IoT environments. The extensive size of this dataset poses a rich resource for studying and designing an intrusion system due to its comprehensive features and the inclusion of real world IoT devices.

Recently, there has been a lot of current research focused on studying part of this data, however, none of them studies the whole dataset. Raghavendra et al. [10], propose a framework to detect intrusion in IoT based on CICIoT2023 dataset. They used different feature selection and class-balancing techniques to solve data imbalance problem. They found that the combination of CfsSubsetEval (CFS) for feature selection with the Balanced RF Classifier sampling (BRFC) technique improves the classification accuracy. The results showed high accuracy (greater than 99%) and an extremely low F1 score (less than 76%) in the proposed models, which could indicate the effect of a class imbalance issue, where one class is much more frequent than the others.

Wang et al. [12], propose a new model, called DL-BiLSTM to detect intrusion in the network. This model is based on deep neural networks (DNNs) with bidirectional long short-term memory networks (BiLSTMs). In addition to applying incremental principal component analysis (IPCA) algorithm for reducing feature dimensionality. The model is tested on three datasets CICIoT2023, CIC IDS2017, and N-BaIoT. The proposed model outperformed the traditional deep learning model techniques in terms of detection performance, while maintaining a lower model complexity. This research focuses only on reducing the complexity of the model alone by relying on two primary classes: attack, not attack, rather than improving the detection performance across all 33 distinct classes of attacks in the CICIoT2023 dataset.

Sahin et al. [11], applied multiple ML such as RF, K-Nearest Neighbors (KNN), Decision Tree (DT), Weighted K-Nearest Neighbors (WKNN), and Multi-Layer Perceptron (MLP). Their research showed that extracting only the inter-packet arrival time (IAT) feature enhanced detection rates.

Although prior research on the CICIoT2023 dataset reports good contributions, it does not address the issue of the imbalanced distribution of attack classes. Studies should focus on using the entire dataset, including all 33 attack classes, and investigating techniques for handling imbalanced data within deep learning models.

Table 1: Summary of Feature Categories and Their Corresponding Attributes

Basic Information	Packet Flags	Protocol Types	Statistical Measures	Advanced Metrics
Timestamp	SYN	IP	Total Sum	Magnitude
Flow Duration	ACK	TCP	Min	Radius
Header Length	RST	UDP	Max	Covariance
	FIN	ICMP	Avg	Variance
	PSH	HTTP	Std Dev	Weight
	URG	HTTPS	Packet Count	
	ECE	DNS		
	CWR Flags	SSH		
		ARP		

3. CICIoT2023 DATASET

Our experiments use the CICIoT2023 dataset [9]. Each input data is represented by 47 network traffic features. TABLE 1 summarizes these features, which comprise a range of categories from basic network attributes to advanced traffic metrics. The dataset is divided into seven attack domains. Each attack can be further divided into subattacks, with a total of 33 attack classes. The dataset has a benign class for the traffic that does not contain malicious activity or intrusion attempts.

FIGURE 1 shows the distribution of instances under each attack domain and the number of instances across all 33 classes. As shown, the dataset is highly imbalanced, with DDoS being the most frequent class, followed by DoS and Mirai. The remaining categories: Spoofing, Recon, Web-Based, and Brute Force represent minority classes. In terms of sub-attack distribution, some classes have significantly fewer instances than others, such as 'Uploading Attack,' compared to DDoS-ICMP Flood. This class imbalance poses significant challenges for machine learning models, as they tend to favour majority classes while struggling to classify minority attacks accurately. The original dataset contains a single 'Brute Force' class that has no finer grained subclasses, whereas every other domain is represented by multiple subclasses (e.g., twelve distinct DDoS variants). To keep the hierarchy consistent, we exclude the Brute Force class and report all results on the remaining 32 attack classes plus the Benign Traffic class (33 classes in total)

In machine learning, most algorithms prioritize majority classes, often overlooking or misclassifying the minority samples. These minority samples, although infrequent, carry significant importance in the dataset [26]. Class imbalance in a dataset can introduce several challenges and disadvantages when applying data mining classification techniques such as bias towards majority class and limited feature importance for minority class [27].

4. METHODOLOGY

The primary challenge of CICIoT2023 lies in the unbalanced distribution of attack categories, which poses a risk of bias in machine learning models. To mitigate the effects of imbalanced data, we implemented an attack-aware feature selection approach to improve classification accuracy across all classes and reduce bias caused by imbalanced data. In order to implement our proposed

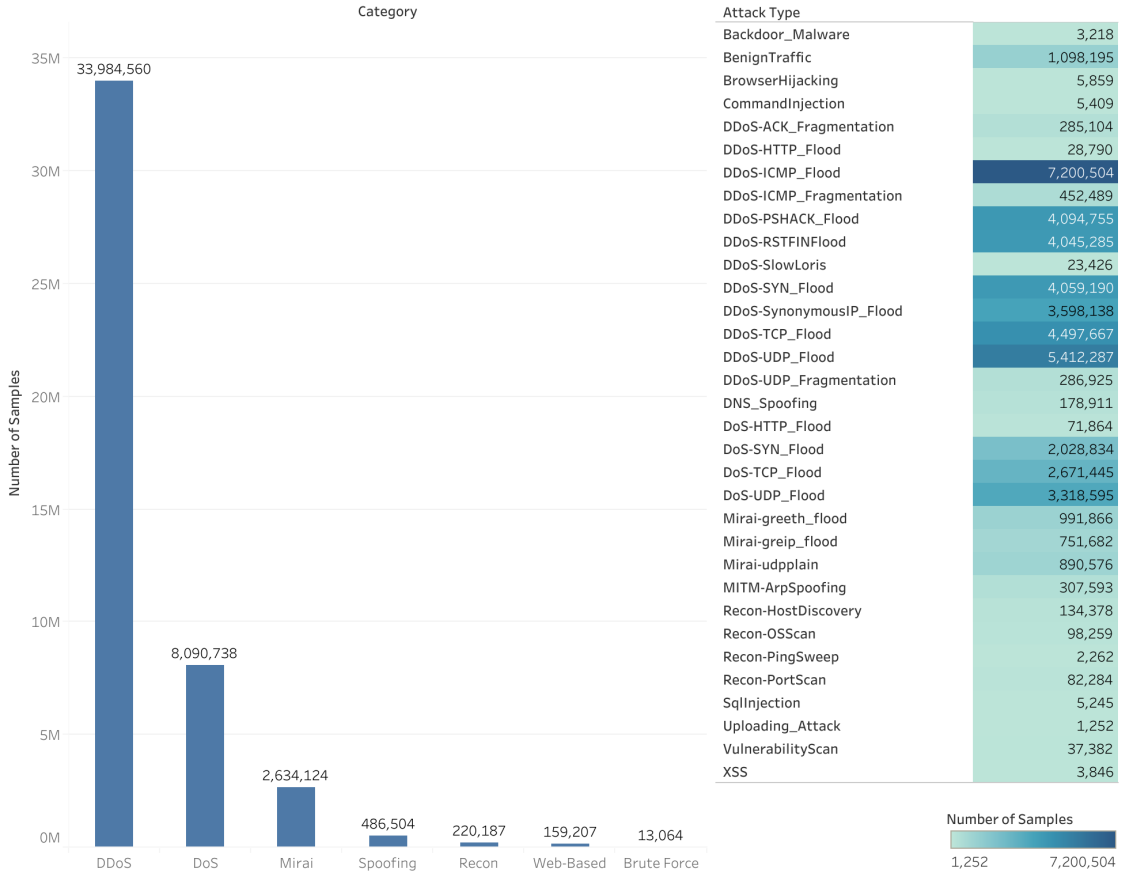

Figure 1: Class distribution of CICIoT 2023 dataset. The right panel shows the number of instances in individual class, and the left panel shows the number of instances after data division.

method, we utilized the concept of a homogeneous ensemble for feature selection [28]. FIGURE 2 illustrates the paradigm of this concept. It consists of training data subsets, a feature selector, and an aggregation method. It is called homogeneous because the same type of feature selector is applied to each subset.

This method ensures that feature selection is adapted to the unique characteristics of different data categories. In our case, it takes into account that different attack types may have distinct important features. Otherwise, if we relied on a global feature selector, the extracted important features will be more unique to the majority classes and irrelevant and might be even misleading to the minority classes and this leads to skewed performance.

The process involves analyzing different subsets of data, where each subset contains uniform characteristics, to identify the most relevant features within that group. Features are ordered based on their

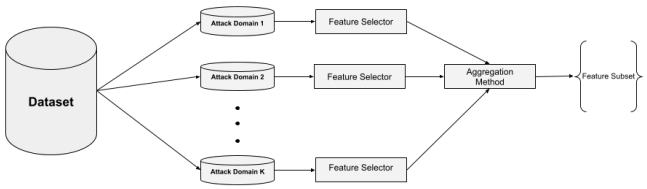

Figure 2: Homogeneous Ensemble Feature Selection Paradigm. The method involves dividing the dataset into subsets. Then, it applies a uniform feature selector to each subset, and aggregates the selected features to improve model performance.

contribution to prediction accuracy, and the selected features from all subsets are then aggregated for use in a global model.

For example, in an IoT attack dataset, features like packet size variance might be highly relevant for detecting DDoS attacks, whereas protocol type could be a key indicator for reconnaissance attacks. This approach computes feature rankings within homogeneous attack subsets and then aggregates them. This reduces the dominance of majority classes patterns and lead to improved detection rate across all classes.

4.1 Attack-Aware Feature Aggregation Model Framework (AFAM)

We propose AFAM (FIGURE 3), an attack-aware Feature Aggregation Model. AFAM extends the homogeneous ensemble feature selection approach [28]. It begins with partitioning the training data by attack domain. Then, it performs feature selection independently on each subset using two selectors (XGBoost and Random Forest). Each selector produces an importance score per feature within a subset to select the most important features across all attacks. Then, the scores are combined using the Maximum Aggregation Method. As illustrated in FIGURE 3, AFAM comprises four steps: (1) attack-based partitioning, (2) Attack-Based feature selection, (3) feature aggregation, and (4) multi-class detection. The following subsections detail each stage of AFAM.

4.1.1 Attack-based dataset partitioning

The dataset is divided into six subsets based on attack domains: DDoS, DoS, Mirai, Spoofing, Recon, and Web. The aim of this step is to isolate the sub attacks within each domain. This allow extracting the unique characteristics and feature importance of each attack domain.

4.1.2 Attack-based feature selection

In the second step, we employ two powerful feature selection algorithms: XGBoost and RF. The goal is to evaluate the importance of various features within each subset. As a result, this step generates feature importance scores for each attack after applying XGBoost and RF. These algorithms were

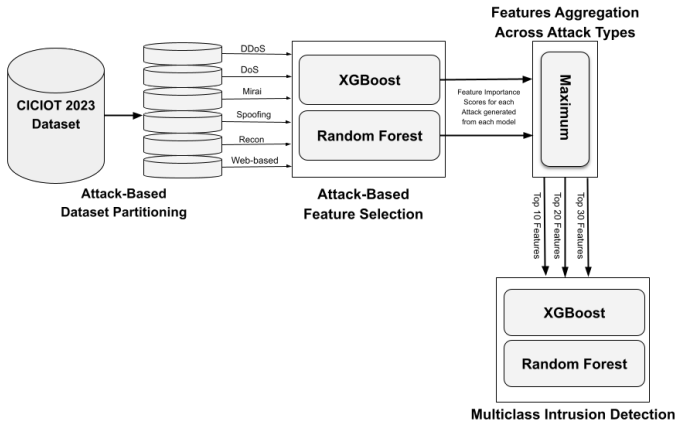

Figure 3: Proposed Attack-Aware Feature Aggregation Model Framework using XGBoost and RF on the CICIoT 2023 Dataset. (1) The dataset is divided according to attack types. (2) XGBoost and RF derive the feature importance scores. (3) The most important features from each domain are aggregated using a maximum function and the top 10, 20, and 30 features are selected for classification. (4) XGBoost and RF models perform multiclass intrusion detection on selected features.

chosen due to their ability to capture complex relationships and produce stable feature rankings on tabular traffic features. The feature selection process is performed independently for each attack subset and for each model. In XGBoost, we compute feature importance scores according to gain. The gain measures the impact of each feature in the decision trees. Features with higher gain values are considered more important [29]. In RF, we use impurity based importance such as Gini impurity or information gain to determine feature importance scores. The importance score is computed as the average impurity decrease for each feature across all trees in the forest [30].

4.1.3 Features aggregation across attack types

After identifying the most important features in each subset, we apply the Maximum Aggregation Method (Algorithm 1) to aggregate feature importance scores across all attack domains.

This approach selects features with the highest individual importance score across any of the attack subsets. The goal is to ensure that features that are particularly impactful for at least one specific attack type are not overlooked in the overall selection.

4.1.4 Multi-class intrusion detection

For each selected feature set (top 10, 20, and 30 features), we train XGBoost and RF models on the entire CICIoT2023 dataset. This allows us to compare model performance across different feature sets.

Algorithm 1: Homogeneous Feature Selection with Maximum Aggregation

Input: $feature\_importances$ (dictionary mapping attack categories to DataFrames with `Feature` and `Importance` columns)
Output: $selected\_features, top\_features$
$max\_scores \leftarrow \{\}$;
foreach (c, df) *in* $feature\_importances$ **do**
  foreach $(feature, score)$ *in* df **do**
    if $feature \notin max\_scores$ ***or*** $score > max\_scores[feature]$ **then**
      $max\_scores[feature] \leftarrow score$;
Sort max_scores in descending order by score;
$top\_features \leftarrow$ top 10, 20, 30 features from sorted max_scores;
$selected\_features \leftarrow$ keys of $top\_features$;
return $(selected\_features, top\_features)$;

Training XGBoost and RF models on the entire dataset using the selected features from the small models enhances the generalization capability of the models. This is because the features are chosen based on their importance across multiple subsets where they are relevant for a wide range of attack types.

4.2 Training and Evaluation Scenarios

We compared two classification scenarios to evaluate the performance of our framework.

- **Base Model Scenario:** In this scenario, we trained XGBoost classifier, which is called **XGBoost-base-model**, and RF classifier that is called **RF-base-model** on the entire dataset without division or feature selection. This gives us the baseline reference for model performance.

- **Attack-Aware Feature Aggregation Model Scenario:** In contrast, we applied the proposed AFAM framework on both XGBoost and RF. In this case, feature selection are applied on data subdivision using the Maximum Aggregation Method before training the models. The models in this scenario are called **AFAM-XGBoost** and **AFAM-RF**.

To ensure an appropriate evaluation, all models run on the same environment and under the same preprocessing steps, starting by splitting the data using an 80:20 train-test split, then data normalization and feature scaling for data standardization. All experiments run on a system with NVIDIA GTX GPUs, Python, and the scikit-learn library for efficient training and evaluation.

4.2.1 Evaluation metrics

To compare the AFAM model against the base models (XGBoost and RF), we use the following evaluation metrics:

1. **Accuracy** measures the proportion of total predictions that were correct, including both positive and negative classifications.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{1}$$

2. **Precision** assesses the percentage of real positives among all identified positives.

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{2}$$

3. **Recall (Sensitivity)** evaluates the model's ability to correctly identify all actual positives and reflects the proportion of true positives out of the positives that should have been identified.

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{3}$$

4. **F1 Score** provides a balance between precision and recall and it is the harmonic mean of these two metrics. This offers a clear measure of the model's accuracy in classifying data.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{4}$$

In model evaluation, we highlight the use of the Macro F1-Score for different reasons: (i) the dataset is highly imbalanced. (ii) This measure assigns equal importance to all classes, regardless of size. (iii) The Macro F1-Score evaluates each class independently and then averages the results, unlike the accuracy measure, which can be biased toward majority classes.

This will ensure fair model evaluation, especially for minority attack classes, which are often critical in real world applications, and are dominated by majority classes during evaluation.

Additionally, we conduct a comparative analysis to examine the impact of our approach in dataset partitioning to enhance feature selection on the performance of XGBoost and RF models in the intrusion detection task.

5. RESULTS AND DISCUSSION

To evaluate the efficacy of our framework (AFAM), we compare it with baseline models (XGBoost-base-model and RF-base-model). Then, we analyze its impact on different attack classes. We present results across three major evaluation aspects: Overall Model Performance, Detection Performance on Majority vs. Minority Attacks, and Analysis of Selected Features.

5.1 Overall Model Performance

TABLE 2 presents a comparative analysis of the baseline models and AFAM variants trained with different feature selection configurations. As shown in TABLE 2, AFAM-RF (Top 30) achieved

Table 2: Performance Comparison of Baseline and Attack-Aware Feature Aggregation Model Framework (AFAM)

Model	Accuracy	Precision	Recall	F1-Score
Baseline Models				
XGBoost-base-model	99.40%	76.88%	87.38%	79.11%
RF-base-model	99.56%	94.25%	80.67%	83.81%
AFAM-XGBoost				
Top 10 with XGBoost	98.81%	70.50%	79.86%	71.41%
Top 20 with XGBoost	99.34%	76.25%	87.74%	78.42%
Top 30 with XGBoost	99.39%	88.25%	76.89%	79.17%
Top 10 with RF	99.15%	83.57%	91.48%	86.68%
Top 20 with RF	**99.60%**	85.76%	**96.26%**	89.39%
Top 30 with RF	99.59%	95.82%	83.23%	86.81%
AFAM-RF				
Top 10 with XGBoost	99.53%	**96.54%**	86.59%	90.32%
Top 20 with XGBoost	99.57%	96.42%	87.26%	90.79%
Top 30 with XGBoost	99.59%	96.44%	87.32%	**90.83%**
Top 10 with RF	99.55%	96.08%	86.14%	89.78%
Top 20 with RF	99.58%	96.29%	84.72%	88.46%
Top 30 with RF	99.58%	95.71%	83.05%	86.60%

the highest F1-score of 90.83% when trained with XGBoost. It outperformed all baseline models (XGBoost-base-model: 79.11%, RF-base-model: 83.81%) as well as all other AFAM configurations. Moreover, all other AFAM-RF variations (Top 10 and Top 20) also achieved their highest F1-scores when trained with XGBoost.

On the other hand, training the AFAM-RF features with RF led to weaker results compared to XGBoost. However, its performance remains comparable to all AFAM-XGBoost feature variations when trained with RF. Notably, all AFAM-XGBoost variations resulted in the lowest performance when trained with XGBoost among both the baseline models and AFAM-RF configurations. This suggests that AFAM-RF, when combined with XGBoost, is the most effective framework for this task.

The experimental results at this level demonstrate that the AFAM-RF framework, when integrated with XGBoost, achieves superior detection performance compared to all other configurations. In contrast, AFAM-XGBoost exhibits limited efficacy when trained on its own feature representations. These results show that variation in model performance can be affected by the feature aggregation strategy and selection, and not only by the model architecture.

5.2 Detection Performance for Minority vs. Majority Classes

TABLE 3 presents a comparative analysis of minority classes in CICIoT2023, which compares the baseline models with the AFAM-RF (Top 30) variant trained with XGBoost.

Table 3: Performance Metrics (Precision, Recall, F1-Score) for Minority Classes with Highlighted Best Model

Class	XGBoost-base-model			RF-base-model			(AFAM) Top 30 with XGBoost		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Backdoor_Malware	51.69%	9.12%	15.5%	87.72%	22.42%	35.71%	**90.14%**	**56.05%**	**69.12%**
BrowserHijacking	84.16%	27.3%	41.22%	95.42%	43.89%	60.13%	**97.19%**	**58.3%**	**72.88%**
CommandInjection	79.23%	27.05%	40.33%	84.51%	42.26%	56.34%	**95.49%**	**63.15%**	**76.02%**
DNS_Spoofing	74.41%	69.71%	71.98%	**83.52%**	**77.25%**	**80.27%**	84.77%	74.59%	79.36%
DoS-HTTP_Flood	99.59%	99.88%	99.73%	99.78%	99.7%	99.74%	**99.94%**	**99.94%**	**99.94%**
MITM-ArpSpoofing	90.22%	82.14%	85.99%	**93.61%**	**86.11%**	**89.7%**	92.68%	85.57%	88.98%
Recon-HostDiscovery	80.55%	81.62%	81.08%	87.66%	87.84%	87.75%	**88.02%**	**88.61%**	**88.31%**
Recon-OSScan	69.65%	40.48%	51.2%	85.5%	63.83%	73.1%	**88.15%**	**64.09%**	**74.22%**
Recon-PingSweep	66.32%	13.85%	22.91%	73.24%	11.43%	19.77%	**93.91%**	**57.58%**	**71.39%**
Recon-PortScan	68.44%	62.33%	65.24%	85.15%	68.11%	75.68%	**81.37%**	**71.58%**	**76.16%**
SqlInjection	55.52%	18.77%	28.05%	94.69%	33.33%	49.31%	**91.86%**	**57.98%**	**71.09%**
Uploading_Attack	41.46%	6.94%	11.89%	78.95%	6.12%	11.36%	**95.49%**	**51.84%**	**67.2%**
VulnerabilityScan	98.91%	99.41%	99.16%	98.66%	99.38%	99.02%	**99.87%**	**99.93%**	**99.9%**
XSS	36.92%	3.17%	5.84%	75.1%	24.31%	36.73%	**92.99%**	**54.29%**	**68.56%**

We focus our analysis on the RF baseline in comparison with the proposed AFAM model. Although the RF-based model generally outperformed the XGBoost baseline for most minority attack types, its performance was still inadequate as it failed to achieve reliable precision, recall, or F1-scores.

The results show that the proposed model (AFAM, Top 30 with XGBoost) achieved higher F1-scores across almost all minority classes,especially for previously under represented attacks, such as ”Uploading_Attack, Recon-PingSweep”. This shows the effectiveness and strength of the proposed model in recognizing uncommon and difficult attack types that biased models frequently miss.

For example, the most substantial improvement was observed in the detection of Uploading_Attack, where the F1-score increased from 11.36% (RF-base) to 67.2% using AFAM with an impressive relative improvement of 491.55%. Similarly, large gains were seen in Recon-PingSweep (from 19.77% to 71.39%, +261.10%), Backdoor_Malware (+93.56%), XSS (+86.66%), and SQL Injection (+44.17%).

These enhancements in model performance demonstrate that the proposed model not only improves detection in majority classes but is also capable at tackling the imbalance issue and significantly improving performance for minority attacks.

It is worth noting that for attack types where the RF-base model already performed strongly such as ”DNS Spoofing” and ”MITM-ArpSpoofing” the gains were more modest. For example, ”DNS Spoofing” saw only a slight decrease in F1-score (from 80.27% to 79.36%), likely because the base model had already learned sufficiently discriminative features for that class.

In contrast, TABLE 4 shows the performance of the baseline models compared to the proposed AFAM model on the majority classes. Even though the baseline models were able to detect and classify these classes, the AFAM model slightly improved performance across all metrics.

This indicates that the gains achieved for minority classes did not compromise the model’s ability to identify the more frequent attacks off in imbalanced learning scenarios. For example, in the ”DDoS-SlowLoris” class, the F1-score increased from 98.70% (XGBoost) and 98.31% (RF) to 99.86% with the proposed approach.

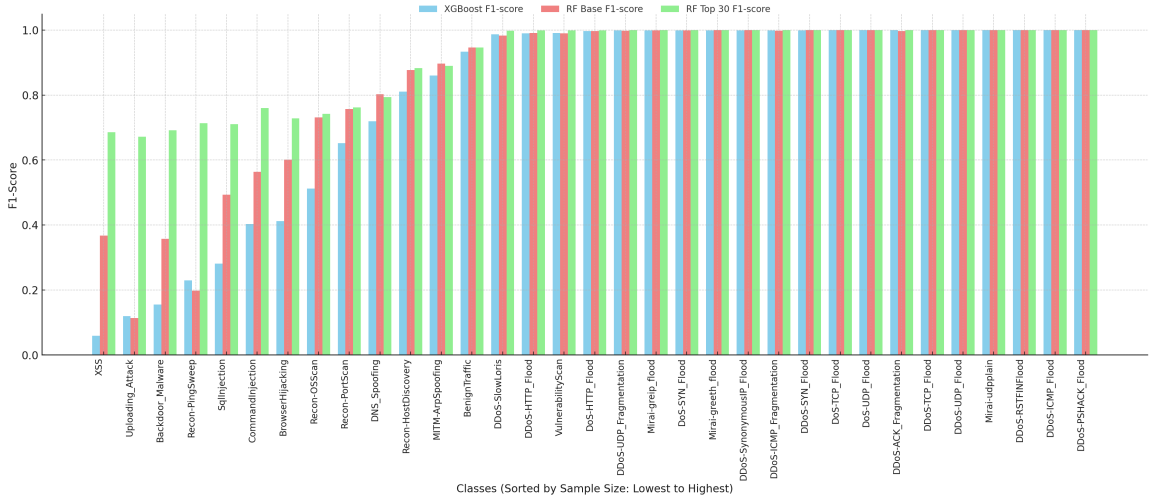

Figure 4: F1-Score Comparison Across Majority and Minority Attack Classes in CICIoT2023 of three models: XGBoost-base-model (blue), RF-base-model (red), and the proposed model AFAM model (green). The Attack classes are ordered by sample size from Minority to Majority. The AFAM achieved the highest F1 score for the Minority Attack Classes while maintaining the performance of the Majority Attack Classes.

Table 4: Performance Metrics (Precision, Recall, F1-Score) for Majority Classes with Highlighted Best Model

Class	**XGBoost-base-model**			**RF-base-model**			**proposed model (AFAM) Top 30 with XGBoost**		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
BenignTraffic	89.32%	97.87%	93.40%	**90.86%**	**98.82%**	**94.67%**	91.00%	98.54%	94.62%
DDoS-ACK_Fragmentation	99.96%	99.98%	99.97%	99.71%	99.77%	99.74%	**99.99%**	**99.99%**	**99.99%**
DDoS-HTTP_Flood	99.42%	98.71%	99.06%	99.44%	98.74%	99.09%	**99.86%**	**99.88%**	**99.87%**
DDoS-ICMP_Flood	**100.00%**	**100.00%**	**100.00%**	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
DDoS-ICMP_Fragmentation	99.94%	99.97%	99.95%	99.76%	99.93%	99.85%	**99.99%**	**99.99%**	**99.99%**
DDoS-PSHACK_Flood	**100.00%**	**100.00%**	**100.00%**	99.99%	99.98%	99.99%	100.00%	100.00%	100.00%
DDoS-RSTFINFlood	**100.00%**	**100.00%**	**100.00%**	100.00%	99.98%	99.99%	100.00%	100.00%	100.00%
DDoS-SYN_Flood	99.97%	99.94%	99.96%	99.96%	99.98%	99.97%	**100.00%**	**100.00%**	**100.00%**
DDoS-SlowLoris	98.16%	99.24%	98.70%	97.35%	99.29%	98.31%	**99.87%**	**99.85%**	**99.86%**
DDoS-SynonymousIP_Flood	99.92%	99.97%	99.95%	99.98%	99.97%	99.98%	**100.00%**	**100.00%**	**100.00%**
DDoS-TCP_Flood	99.96%	100.00%	99.98%	100.00%	100.00%	100.00%	**100.00%**	**100.00%**	**100.00%**
DDoS-UDP_Flood	99.98%	99.98%	99.98%	99.98%	99.98%	99.98%	**100.00%**	**100.00%**	**100.00%**
DDoS-UDP_Fragmentation	99.93%	99.92%	99.92%	99.77%	99.81%	99.79%	**99.99%**	**100.00%**	**99.99%**
DoS-SYN_Flood	99.96%	99.92%	99.94%	99.96%	99.93%	99.95%	**99.98%**	**99.99%**	**99.99%**
DoS-TCP_Flood	99.99%	99.94%	99.97%	99.99%	99.99%	99.99%	**100.00%**	**100.00%**	**100.00%**
DoS-UDP_Flood	99.96%	99.96%	99.96%	99.98%	99.97%	99.98%	**99.99%**	**99.98%**	**99.99%**
Mirai-greeth_flood	99.96%	99.92%	99.94%	99.97%	99.97%	99.97%	**99.99%**	**99.99%**	**99.99%**
Mirai-greip_flood	99.90%	99.96%	99.93%	99.92%	99.95%	99.94%	**99.99%**	**99.98%**	**99.98%**
Mirai-udpplain	99.99%	99.98%	99.99%	99.98%	99.98%	99.98%	**100.00%**	**100.00%**	**100.00%**

FIGURE 4 shows F1-scores of the three models in our experiments: XGBoost-base-model, RF-base-model, and the proposed AFAM model. All classes in this figure are arranged from the smallest (minority) to the largest (majority). Overall, the AFAM model perform higher across most classes, especially the minority classes. This highlighting how our proposed model deals with class imbalance more effectively.

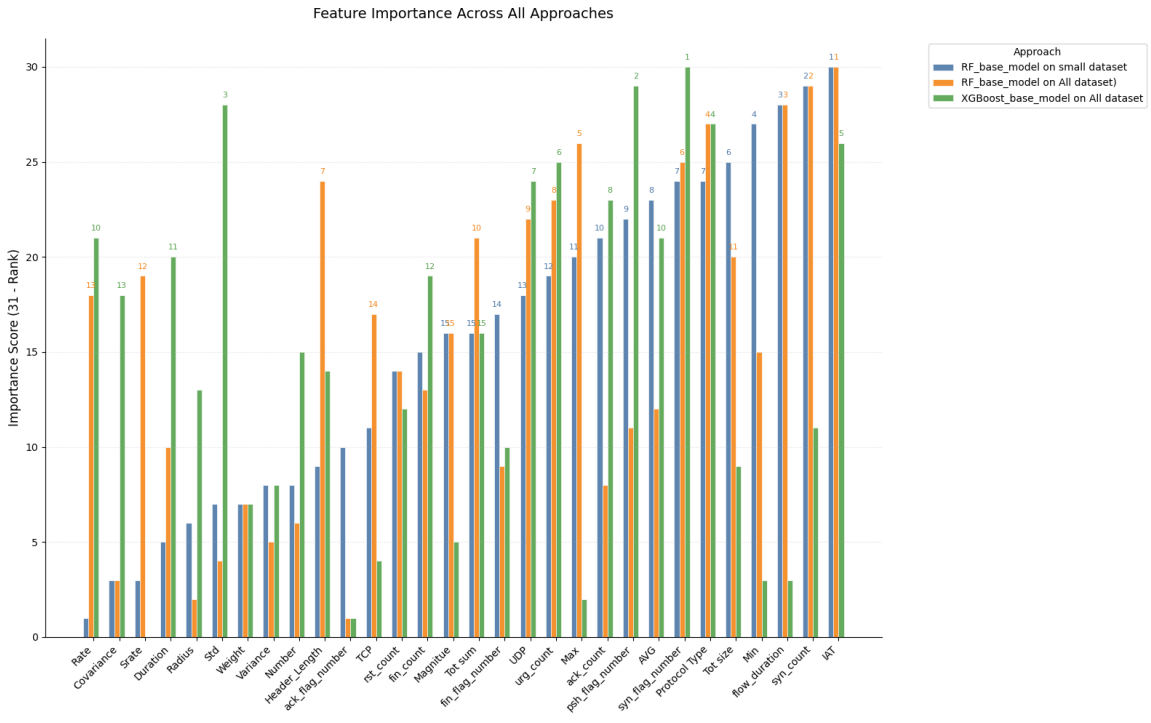

Figure 5: Consensus ranking of the top 30 features across multiple models: Random Forest model on small dataset (blue), Base Random Forest model on All dataset (orange) and Base XGBoost model on All dataset (green). Y axis represents the consumed ranking score.

The results show that the proposed model maintains a consistent performance across both majority and minority classes. By selecting the most informative features, the model improves sensitivity to minority classes while maintaining the detection accuracy for majority attack types. This balance is essential for real world intrusion detection systems, which need to identify both rare and frequent attacks in imbalanced settings.

5.3 Analysis of Selected Features

FIGURE 5 shows a comparison between features consensus ranking of the top 30 across multiple models in case if small or all dataset is used with RF-base-model and XGBoost-base-model . As we can see RF-base-model on small datasets (blue bars) demonstrate consistent ranking on all features, while other models (XGBoost-base-model and RF-base-model) on the full dataset show greater fluctuation in importance scores.

Some features like Syn-flag-number, Std, Duration, and Rate are ranked very high in XGBoost-base-model but receive lower rank in RF models. Other features like UDP, IAT, Protocal Type show moderate importance in both RF and XGBoost models on all dataset, but they are less emphasized by the RF-base-model on the small datasets.

From this comparison, RF-base-model on small datasets has led to a balanced perspective on feature relevance and it has avoided the strong bias toward patterns present in larger datasets. For this reason, we select this strategy as feature selection before training the model using XGBoost classifier on the full dataset.

The integration between RF during feature selection and XGBoost during classification gave us the highest interpretability of the features before classifying attacks. As shown in the experimental results that XGBoost with RF for feature selection achieved the highest accuracy across both majority and minority classes.

6. CONCLUSION AND FUTURE WORK

This study discusses the issue of handling class imbalance in the CICIoT2023 dataset, a common challenge in IoT intrusion detection. It is a comprehensive dataset consisting of 47 features and 33 different attack classes. However, it is highly skewed, with one attack having 7.2 million samples (DDoS-ICMP_Flood) whereas another attack (Uploading_Attack) has around 1.2 thousand.

To tackle this issue, the study proposes an Attack-Aware Feature Aggregation Model Framework (AFAM). AFAM employs a homogeneous ensemble feature selection approach [28] to select features based on their influence within each attack class using XGBoost and Random Forest as selectors. This leads to a more representative feature set that does not overly favor the majority classes (attacks with higher sample sizes). Then, XGBoost and Random Forest are used as global classifiers on the representative feature set.

In terms of results, AFAM has improved performance metrics across all minority attack classes while maintaining the performance of majority attacks. This indicates that attack-class feature selection, followed by feature aggregation, forms a representative feature set that enhances detection accuracy.

As for future work, one direction is to test AFAM using other machine learning methods such as deep learning. Also, it would be useful to investigate how various attack classes affect the framework's performance and explore other feature selection methods designed to handle imbalanced datasets. Another important direction is to test and deploy our framework in real world IoT environments, such as smart campuses , and help improve the protection of such systems in practice.

References

[1] Nguyen DC, Ding M, Pathirana PN, Seneviratne A, Li J, et al. 6G Internet of Things: A Comprehensive Survey. IEEE Internet Things J. IEEE. 2021;9:359-383.

[2] Castiglione A, Umer M, Sadiq S, Obaidat MS, Vijayakumar P. The Role of Internet of Things to Control the Outbreak of COVID-19 Pandemic. In IEEE Internet Things J. 2021;8:16072-16082.

[3] Da Xu LD, He W, Li S. Internet of Things in Industries: A Survey. IEEE Trans Ind Inform. 2014;10:2233-2243.

[4] Nauman A, Qadri YA, Amjad M, Zikria YB, Afzal MK, et al. Multimedia Internet of Things: A Comprehensive Survey. IEEE Access. 2020;8:8202-8250.

[5] Habibzadeh H, Dinesh K, Shishvan OR, Boggio-Dandry A, Sharma G, et al. A Survey of Healthcare Internet of Things (HIoT): a clinical perspective. In IEEE Internet Things Journal. 2019;7:53-71.

[6] https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

[7] Denning DE. An Intrusion-Detection Model. IEEE Trans Softw Eng. 1987;SE-13:222-232.

[8] Bridges RA, Glass-Vanderlan TR, Iannacone MD, Vincent MS, Chen Q. A Survey of Intrusion Detection Systems Leveraging Host Data. ACM Comput Surv. 2019;52:1-35.

[9] Neto PEC, Dadkhah S, Ferreira R, Zohourian A, Lu R, et al. Ciciot2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment. Sensors. 2023;23:5941.

[10] Narayan KR, Mookherji S, Odelu V, Prasath R, Turlapaty AC, et al. Iids: Design of Intelligent Intrusion Detection System for Internet-Of-Things Applications. 2023. ArXiv pre-print: https://arxiv.org/pdf/2308.00943

[11] https://www.researchgate.net/publication/374563932_Advancing_Intrusion_Detection_Efficiency_A_'Less_is_More'_Approach_via_Feature_Selection

[12] Wang Z, Chen H, Yang S, Luo X, Li D, et al. A Lightweight Intrusion Detection Method for IoT Based on Deep Learning and Dynamic Quantization. PeerJ Comput Sci. 2023;9:e1569.

[13] Hasan K, Hossain KS, Apurbo GM, Islam MD, Alam MS. Real-Time Ddos Detection in Software-Defined Networks Using Machine Learning [Doctoral dissertation, Brac University]. 2024. Available at: https://dspace.bracu.ac.bd/xmlui/bitstream/handle/10361/24344/20101332%2C%2020101321%2C%2020301100%2C%20201013 22%2C%2020301286_CSE.pdf?sequence=1&isAllowed=y

[14] Behera A, Sagar Sahoo KS, Kumara Mishra T, Nayyar A, Bilal M. Enhancing DDoS Detection in SDIoT Through Effective Feature Selection With SMOTE-ENN. PLOS One. 2024;19:e0309682.

[15] Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. IEEE Commun Surv Tutor. 2015;17:2347-2376.

[16] Apostol I, Preda M, Nila C, Bica I. IoT Botnet Anomaly Detection Using Unsupervised Deep Learning. Electronics. 2021;10:1876.

[17] Kumar A, Shridhar M, Swaminathan S, Lim TJ. Machine Learning-Based Early Detection of IoT Botnets Using Network-Edge Traffic. Comput Secur. 2022;117:102693.

[18] Churcher A, Ullah R, Ahmad J, Ur Rehman S, Masood F, et al. An Experimental Analysis of Attack Classification Using Machine Learning in IoT Networks. Sensors. 2021;21:446.

[19] Qaddoura R, Al-Zoubi AM, Almomani I, Faris H. A Multi-Stage Classification Approach for IoT Intrusion Detection Based on Clustering With Oversampling. Appl Sci. 2021;11:3022.

[20] Alkadi O, Moustafa N, Turnbull B, Choo KK. A Deep Blockchain Framework Enabled Collaborative Intrusion Detection for Protecting IoT and Cloud Networks. IEEE Internet Things J. 2020;8:9463-9472.

[21] Fernando KR, Tsokos CP. Dynamically Weighted Balanced Loss: Class Imbalanced Learning and Confidence Calibration of Deep Neural Networks. IEEE Trans Neural Netw Learn Syst. IEEE. 2022;33:2940-2951.

[22] https://iEEE-dataport.org/documents/bot-iot-dataset

[23] https://www.stratosphereips.org/datasets-iot23

[24] Meidan Y, Bohadana M, Mathov Y, Mirsky Y, Shabtai A, et al. N-Baiot Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders. IEEE Pervasive Comput. 2018;17:12-22.

[25] Sharafaldin I, Lashkari AH, Ghorbani AA. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. ICISSp. 2018;1:108-116.

[26] Longadge R, Dongre S. Class Imbalance Problem in Data Mining Review. 2013. ArXiv preprint: https://arxiv.org/pdf/1305.1707

[27] https://www.researchgate.net/profile/Vaishali-Ganganwar/publication/292018027_An_overview_of_classification_algorithms_for_imbalanced_datasets/links/58c7707a458515478dc4c68b/An-overview-of-classification-algorithms-for-imbalanced-datasets.pdf

[28] Bolón-Canedo V, Alonso-Betanzos A. Ensembles for Feature Selection: A Review and Future Trends. Inf Fusion. 2019;52:1-12.

[29] Oppelt MP, Foltyn A, Deuschel J, Lang NR, Holzer N, et al. ADABase: A Multimodal Dataset for Cognitive Load Estimation. Sensors. 2023;23:340.

[30] Breiman L. Random Forests. Mach Learn. Springer Nature. 2001;45:5-32.