Evaluating Tree-based Ensemble Strategies for Imbalanced Network Attack Classification

Hui Fern Soon[1], Amiza Amir[2], Hiromitsu Nishizaki[3], Nik Adilah Hanin Zahri[4],
Latifah Munirah Kamarudin[5], Saidatul Norlyana Azemi[6]
Faculty of Electronic Engineering & Technology
Universiti Malaysia Perlis Arau, Perlis, Malaysia
University of Yamanashi, Kofu, Yamanashi, Japan[1]
Faculty of Electronic Engineering & Technology
Centre of Excellence for Advanced Computing (ADVCOMP)
Universiti Malaysia Perlis Arau, 02600, Perlis, Malaysia[2,4]
Integrated Graduate School of Medicine, Engineering & Agricultural Science
University of Yamanashi, Kofu, Yamanashi, Japan[3]
Faculty of Electronic Engineering & Technology
Centre of Excellence for Advanced Sensor Technology (CEASTECH)
Universiti Malaysia Perlis Arau, 02600, Perlis, Malaysia[5]
Faculty of Electronic Engineering & Technology
Centre of Excellence for Advanced Communication Engineering (ACE)
Universiti Malaysia Perlis Arau, 02600, Perlis, Malaysia[6]

***Abstract*—With the continual evolution of cybersecurity threats, the development of effective intrusion detection systems is increasingly crucial and challenging. This study tackles these challenges by exploring imbalanced multiclass classification, a common situation in network intrusion datasets mirroring real-world scenarios. The paper aims to empirically assess the performance of diverse classification algorithms in managing imbalanced class distributions. Experiments were conducted using the UNSW-NB15 network intrusion detection benchmark dataset, comprising ten highly imbalanced classes. The evaluation includes basic, traditional algorithms like the Decision Tree, K-Nearest Neighbor, and Gaussian Naive Bayes, as well as advanced ensemble methods such as Gradient Boosted Decision Trees (GraBoost) and AdaBoost. Our findings reveal that the Decision Tree surpassed the Multi-Layer Perceptron, K-Nearest Neighbor, and Naive Bayes in terms of overall F1-score. Furthermore, thorough evaluations of nine tree-based ensemble algorithms were performed, showcasing their varying efficacy. Bagging, Random Forest, ExtraTrees, and XGBoost achieved the highest F1-scores. However, in individual class analysis, XGBoost demonstrated exceptional performance relative to the other algorithms. This is confirmed by achieving the highest F1-scores in eight out of the ten classes within the dataset. These results establish XGBoost as a predominant method for handling multiclass imbalance classification with Bagging being the closest feasible alternative, as Bagging gains an almost similar accuracy and F1-score as XGBoost.**

***Keywords*—Multiclass imbalanced classification; ensemble algorithm; network attack; UNSW-NB15 dataset; F1-score**

I. Introduction

Following the COVID-19 pandemic, accelerated advancements in information technology have reshaped organizational operations, interpersonal interactions, and service delivery methods. The Internet and cyber technology have facilitated a highly interconnected global society, significantly influencing almost every facet of the modern world. This revolutionizes human lifestyles, transforms various industries, and promotes global innovation. The shift towards remote work and virtual platforms has surged, prompting the development of new tools and technologies to accommodate these changes. Additionally, the healthcare sector has experienced a growth in telemedicine and digital health solutions, enabling remote patient consultations and monitoring. However, these advancements also increase vulnerability to cybersecurity attacks, as cybercriminals view the rapid expansion of IT applications, especially in e-commerce, as lucrative targets. The European Union Agency for Cybersecurity (ENISA) noted a notable rise in cybersecurity incidents during the latter part of 2022 and the first half of 2023, as referenced in [1]. These developments underscore the urgent need for effective, reliable, and robust defense systems against such attacks. Concurrently, with the proliferation of AI, machine learning and deep learning algorithms have emerged as powerful tools for network security.

The effectiveness of machine learning and deep learning models in detecting network attacks hinges on the quality and relevance of the training data. Inadequate or irrelevant training data can yield inaccurate or unreliable outcomes. Therefore, it is essential to ensure the training data for these models is high-quality and representative of actual network attack scenarios. Typically, network traffic remains normal until a cyberattack or network failure occurs, causing a deviation from usual patterns. Machine learning and deep learning models are capable of identifying and learning these anomalies, thereby precisely detecting and classifying network attacks.

Consequently, most of the training data will consist of normal network traffic. The abnormal network traffic dataset, representing potential network attacks, includes various categories of network assaults. Rare or novel attack types might have limited sample sizes, potentially smaller than those found

in common attack types or normal traffic data. This leads to a significant imbalance in class composition, potentially introducing model bias, rendering predictions unreliable, and hindering the detection of rare or new attacks. As noted by [2], most network intrusion datasets are inherently multiclass imbalanced, reflecting real-world conditions. In such datasets, class distribution is uneven (as depicted in Fig. 1), with some classes being minority and others majority.

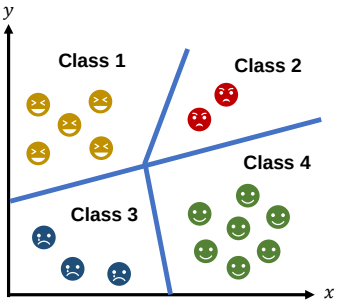

Fig. 1. Example of multiclass imbalanced classification.

Imbalanced datasets can skew classifiers, biasing them toward the majority class [3]. This presents a significant challenge, as most classifiers are inherently designed for balanced scenarios. One simplistic approach could be to exclude minority classes with insufficiently sized samples from the dataset. However, this could result in models that are outdated with respect to the latest cyberattacks. We continuously update and retrain these models with fresh data to enhance their adaptability to evolving attack techniques and ensure sustained effectiveness.

Addressing the imbalance often involves sampling solutions. Techniques such as Random Oversampling [4] or Synthetic Minority Oversampling Technique (SMOTE) [5] augment infrequent cases, while methods like Random Undersampling [6] or Tomek links [7] reduce redundancies in the dataset by decreasing majority samples. Hybrid techniques like SMOTEENN [8], which combine oversampling and undersampling, and ROSE (Random OverSampling Examples) [9], which create synthetic spaces between classes, are also utilized. However, oversampling risks overfitting, and undersampling may lead to information loss. SMOTEENN and ROSE, while versatile, are also prone to overfitting. Moreover, the continually changing nature of new attacks complicates the use of these methods, given the dynamic class distributions. Thus, these methods can temporarily achieve balance but have limitations in long-term applicability and robustness. Consequently, this paper does not focus on sampling solutions but rather on the inherent capabilities of classification algorithms to address imbalanced class problems effectively.

While various machine learning approaches have been proposed for network attack classification, a predominant focus remains on enhancing overall accuracy—a metric poorly suited for imbalanced multiclass datasets. Accuracy measures the proportion of correct predictions made by the model, but it fails to adequately represent minority classes, particularly those with low sample sizes. An accuracy-centric model might disregard minority classes, classifying all instances as the majority class, thereby achieving high overall accuracy but poor detection of rare, yet critical, cases. This oversight necessitates a more nuanced, class-specific evaluation. Additionally, there is a notable research gap concerning the effectiveness of different algorithms in addressing multiclass imbalances.

Therefore, this research has a dual focus. First, it seeks to identify which conventional machine learning algorithms are best suited for addressing the unique challenges of multiclass imbalanced classification, specifically in the context of network attack classification. Second, it explores which ensemble algorithms are most effective in these scenarios. Following guidance from [10], potential solutions include sampling techniques, ensemble methods, cost-sensitive learning, and deep learning methods. This paper, however, concentrates on the application of ensemble approaches to manage imbalanced data scenarios. We compare and experiment with a range of machine learning algorithms, from simpler ones like decision trees and K-nearest neighbors to more complex ensemble algorithms such as Gradient Boosted Decision Trees (GraBoost) and AdaBoost. The objective is to ascertain the most effective algorithm for addressing the complexities of imbalanced datasets in network intrusion detection.

The experimental evaluation utilizes the publicly available UNSW-NB15 dataset [11], characterized by a highly imbalanced class distribution. Initial experiments compared the performance of a single Decision Tree (DT) against instance-based methods like K-Nearest Neighbor (KNN), function-based models including Multilayer Perceptron (MLP), and Bayesian-based approaches exemplified by Naive Bayes (NB).

Despite the initial success of the Decision Tree, there is a need for more precision, particularly in identifying tree-based ensemble algorithms that excel in multiclass imbalance classification. This research thus focuses on discovering the most effective tree-based ensemble algorithms for managing the challenges posed by imbalanced multiclass datasets. In addition to a single Decision Tree, we conducted experiments comparing nine tree-based ensemble learning algorithms: Bagging with a Decision Tree as the base classifier, Random Forest (RF), Extremely Randomized Trees (ExtraTree), Adaptive Boosting (AdaBoost), Gradient Boosting (GraBoost), Histogram-based Gradient Boosting (HistGraBoost), Extreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), and Categorical Gradient Boosting (CatBoost).

To summarize, the paper's primary contributions are as follows. First, preliminary results indicate the superiority of the Decision Tree over other traditional machine learning algorithms. Second, XGBoost has been determined as the optimal tree-based ensemble method for multi-class imbalanced classification with Bagging being the closest feasible alternative. Third, this paper offers practitioners a powerful approach to address the issues often encountered with imbalanced multi-class datasets effectively. Consequently, this improves the overall efficacy of cybersecurity protocols.

The structure of this paper is as follows. Section II explains the related work. Section II describes the methodology. Section IV illustrates the dataset, algorithms and performance metrics used in this research, while Section V describes results of the algorithms. Finally, in Section VI the conclusions and the future works are being discuss.

II. Literature Review

Network attack detection datasets are often multiclass imbalanced [2]. Nevertheless, despite this observed pattern, many research efforts continue to pay attention to tackling the issue of imbalanced classification problems. Typical solutions to dealing with imbalanced dataset issues include utilising sampling approaches [12], [13]. The first sampling approach is to apply the oversampling technique to address the imbalance in minority classes. Random Over-sampling involves the random duplication of cases from the minority class [4]. SMOTE [5], which stands for Synthetic Minority Over-sampling Technique, is a method used to create synthetic samples comparable to the minority data cluster. The second sampling approach is by under-sampling majority classes. For example, the Random Under-sampling [6] randomly removes the majority of class examples, and Tomek links [7] work by removing overlap between class sample distributions. Finally, hybrid/ensemble sampling refers to a technique that combines multiple sampling methods or models to improve the accuracy and reliability of the sampling process. For instance, SMOTEENN [8] is a technique that combines SMOTE over-sampling with edited closest neighbour under-sampling. ROSE(Random OverSampling Examples) sampling [9] generates smooth distributions by creating synthetic spaces between minority and majority examples.

Nevertheless, the deliberate process of oversampling minority classes can lead to over fitting of the model due to the replication and noise. On the contrary, by undersampling the majority classes, there is a risk of losing valuable information crucial for precise classification. Hybrid or ensemble sampling techniques, such as SMOTEEN and Rose sampling, prove helpful in generating more balanced sample distributions. However, class distribution patterns in complex real-world contexts are rarely uniform or evenly distributed. In addition, they inherit the overfitting and losing valuable issues from oversampling and undersampling, respectively. Another key challenge is class distribution concept drift in the dynamic network traffic data. It is possible that novel network attacks will emerge, each with a limited sample size. As relative class frequencies change over time, a previously balanced data set may become outdated.

Hence, this paper aims to identify the best algorithm without considering any sampling approaches. In many studies [14], [15], [16], the classification of network attacks from an imbalanced binary class distribution has been looked at without taking sampling methods into account. Binary classification refers to classification problems where there are only two target classes. Imbalanced binary classification problems occur when one class has many more training examples than the other. Typically, the normal traffic class has a majority, while the abnormal (under attack) traffic class has a significantly smaller minority. The research by [14] aimed to build a classifier to determine whether a Distributed Denial of Service (DDoS) attack occurs on the network. The study employs a range of classifiers, including Extreme Gradient Boosting (XGB), Support Vector Machine (SVM), Logistic Regression (LR), K-Nearest Neighbor (KNN), and Decision Tree (DT). Evaluation metrics such as F1-score, Precision, Recall, and Accuracy indicate XGBoost's strength as the top-performing classifier, achieving an accuracy of 98.24%. In another study for DDoS attack detection, the authors of [15] applied Logistic Regression, K-Nearest Neighbour, Multi-layer Perceptron, and Decision Tree to investigate the best detection model. Notably, KNN and DT demonstrate superior accuracy, especially for TCP and ICMP flooding attacks, while for UDP, DT exhibits a better accuracy of 77.23% with an almost equivalent F1-score.

Concurrently, there exists a group of researchers actively addressing the challenges associated with multiclass imbalanced scenarios in network attack classification. Examples of instances include [17], where the F1-score remains suboptimal, indicating the model is not achieving adequate performance on the minority class, even though overall accuracy appears high. In a study employing the UNSW-NB15 dataset [11], even though the dataset is multiclass imbalanced, the primary emphasis lies on presenting overall performance rather than individual class results. The findings demonstrate that Random Forest attains the best Area Under the Curve (AUC) and F2 scores. Additionally, [18] utilizes the NSL-KDD dataset, comparing the performance of Naive Bayes and SVM. Despite SVM's accuracy exceeding 90%, the F1-score remains around 0.69.

An additional study in [17] developed a model to classify benign network traffic versus malicious attack categories like Distributed Denial of Service (DDoS) attacks that leverage malicious TCP ACK or PSH-ACK packet flows.The results highlight the superiority of logistic regression over other classifiers used in the paper. The study in [19] applied the CICIDS2017 network intrusion detection benchmark to assess an array of both classical (Decision Tree, K-nearest Neighbours, and Support Vector Machine) and ensemble classifiers (Random Forest, GraBoost, and AdaBoost) for identifying malicious network behaviours within realistic traffic. The study reported that GraBoost outperformed other classifiers in terms of accuracy, precision, recall, and F1-score. Meanwhile, AdaBoost struggles with dataset complexity, lagging other classifiers significantly across all metrics.

Network security operates in a dynamic realm where cybersecurity threats continually evolve in complexity and diversity. The deployed classifier must constantly adapt to new attacks. However, a notable proportion of cybersecurity research concentrates on the development of machine learning models without considering the accurate detection of new attacks with a small sample size (minority classes). While studies like [14] and [15] offer insights into algorithmic performance in binary contexts, there exists a significant gap in understanding whether these algorithms retain their effectiveness amid the complexities of multiclass imbalanced datasets. Moreover, the interaction between different algorithms and metrics, such as the F1-score, remains underexplored. Therefore, a comprehensive investigation is needed to identify the high-performance algorithm that overcomes the imbalanced class distribution in the absence of sampling methods to rebalance the distribution. Additionally, the limited reporting of individual class results, as observed in [11], poses a gap in our understanding of algorithmic vulnerabilities and strengths across diverse attack types. Lastly, despite extensive algorithm testing, a systematic exploration of the suitability of different machine learning algorithm families for multiclass imbalanced datasets is lacking. Addressing these research gaps is imperative for advancing

the field, guiding algorithm selection, and advancing network intrusion detection in complex, real-world scenarios.

III. Methodology

A series of experiments followed the procedure outlined in Fig. 2. Initially, the dataset was partitioned into two segments for training and testing purposes. Subsequent experiments evaluated the performance of four distinct traditional machine learning algorithms to identify the optimal base algorithm for the ensemble. Upon determining the optimal conventional algorithm, further tests were conducted to ascertain the most effective ensemble method, utilizing the previously selected traditional algorithm.

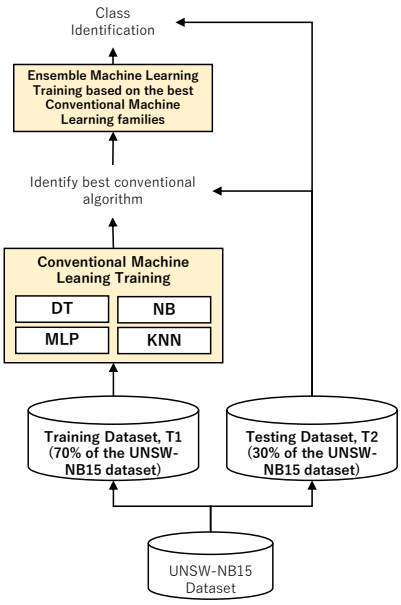

Fig. 2. Experimental evaluation flow.

A. Dataset

In this research, we strategically utilize a highly imbalanced network intrusion dataset, reflective of real-world network anomaly scenarios, as our primary training resource. The dataset selected for this study is the publicly accessible and extensively recognized UNSW-NB15 dataset. It comprises ten different attack categories, represented by 43 features as detailed in Table I. This dataset includes a total of 257,673 instances, categorized into ten distinct classes, as delineated in Table II.

Table II highlights a notable characteristic of the UNSW-NB15 dataset: its classification as a multiclass imbalanced dataset. There are substantial variations in the frequency of different attack categories. These differences mirror the complexity of real-world scenarios, where certain network attacks, although less frequent, may be of higher significance. Categories such as Analysis, Backdoor, Reconnaissance, Shellcode, and Worms, each accounting for less than 6% of the total instances, are thus identified as minority classes in this study.

In order to demonstrate the skewed and highly imbalanced class scenario that exists within this dataset, we present the

TABLE I. Features of the UNSW-NB15 Dataset

No.	Features	Data types	No.	Features	Data types
1	id	int64	23	dtrcpb	int64
2	dur	float64	24	dwin	int64
3	proto	object	25	tcprtt	float64
4	service	object	26	synack	float64
5	state	object	27	ackdat	float64
6	spkts	int64	28	smean	int64
7	dpkts	int64	29	dmean	int64
8	sbytes	int64	30	trans-depth	int64
9	dbytes	int64	31	response-body-len	int64
10	rate	float64	32	ct-srv-src	int64
11	sttl	int64	33	ct-state-ttl	int64
12	dttl	int64	34	ct-dst-ltm	int64
13	sload	float64	35	ct-src-dport-ltm	int64
14	dload	float64	36	ct-dst-sport-ltm	int64
15	sloss	int64	37	ct-dst-src-ltm	int64
16	dloss	int64	38	is-ftp-login	int64
17	sinpkt	float64	39	ct-ftp-cmd	int64
18	dinkpt	float64	40	ct-flw-http-mthd	int64
19	sjit	float64	41	ct-src-ltm	int64
20	djit	float64	42	ct-src-dst	int64
21	swin	int64	43	is-sm-ips-ports	int64
22	stcpb	int64	44	attack-cat	object

disparity between classes by utilising two metrics that are distinct from one another but interconnected metrics. Firstly, the Fraction to Majority Class was calculated using Eq. (1) as shown below:

$$\text{Fraction to Majority Class } (\%) = \frac{TNIPC}{TNIMMC} \times 100 \quad (1)$$

This metric aligns with the challenges identified in the practical scenario of network intrusion detection. Under these circumstances, some classes may have a low occurrence rate yet present a substantial risk. Eq. (2) was applied to calculate the Fraction to Total Instances is shown below:

$$\text{Fraction to Total Instances } (\%) = \frac{TNIPC}{TNIWD} \times 100 \quad (2)$$

For both Eq. (1) and Eq. (2), $TNIPC$ represents the total number of instances for a specific class, $TNIMMC$ is the total number of instances for the most majority class (the class with the highest number of instances), and $TNIWD$ indicates the total number of instances for the whole dataset. Instead of being mere mathematical equations, these equations also provide a clear understanding of the complex, imbalanced distribution of the dataset, which is also the problem found in the real-world situation.

By closely analyzing the imbalance and complexity of the dataset, a strong understanding of the complications of the dataset is established. This is crucial as it will ensure the techniques to be used are able to be utilized accurately and correctly when facing the imbalanced problem.

B. Data Preparation

Before initiating the model training process, several preparatory steps are essential for the UNSW-NB15 dataset to ready the classifiers for subsequent stages. As indicated in Table I, the datatypes of the attack classes were initially in an object format. Consequently, the initial step in this research was to assign a numerical value to each class. This transformation is crucial as it not only standardizes representations but

TABLE II. NUMBER OF INSTANCES IN EACH ATTACK CLASS IN THE UNSW-NB15 DATASET

Classes (attack-cat)	Total number of instances	Fraction to Majority class (Percentage,%)	Fraction to Total instances (Percentage,%)
Analysis	2,677	2.9	1.0
Backdoor	2,329	2.5	0.9
DoS	16,353	17.6	6.3
Exploits	44,525	48.9	17.3
Fuzzers	24,246	26.1	9.4
Generic	58,871	63.3	22.8
Normal	93,000	100	36.1
Reconnaissance	13,987	15.0	5.4
Shellcode	1,511	1.6	0.6
Worms	174	0.2	0.1
Total	**257673**		

TABLE III. DATASET DISTRIBUTION

Classes (attack-cat)	Assigned Number	Total number of Instances in UNSW-NB15 dataset	Number of Instances in Training dataset	Number of Instances in Testing dataset
Analysis	0	2,677	1,874	803
Backdoor	1	2,329	1,630	699
DoS	2	16,353	11,447	4,906
Exploits	3	44,525	31,167	13,358
Fuzzers	4	24,246	16,972	7,274
Generic	5	58,871	41,210	17,661
Normal	6	93,000	65,100	27,900
Reconnaissance	7	13,987	9,791	4,196
Shellcode	8	1511	1,058	453
Worms	9	174	122	52
Total		257,673	180,371	77,302

also ensures compatibility with machine learning algorithms. Furthermore, features such as proto, service, and state, which are initially in an object format, have also been encoded.

After assigning numerical values to the classes, the dataset underwent a stratified 70:30 split. Seventy percent of the data was allocated as the training dataset, enabling the classifier to learn patterns and relationships within the data. The remaining 30% served as the testing dataset, used to evaluate the performance of the trained classifiers in this research. The stratified split ensures equitable representation of all classes in both training and testing datasets, preventing any class from being overrepresented and potentially misleading classifier performance.

The detailed composition of the dataset split is presented in Table III. Employing the aforementioned stratified 70:30 split, instances for each class were proportionately divided between the training and testing datasets. This approach provides a more equitable and accurate assessment of the performance of the algorithms used in this research, particularly in addressing multiclass imbalanced classification challenges.

C. Conventional Machine Learning Algorithms

This paper evaluates four distinct conventional machine learning algorithms, each representing a different family of algorithms: tree-based, instance-based, function-based, and Bayesian-based. These algorithms were chosen for their simplicity and computational efficiency, a desirable trait given the need for rapid training in scenarios involving frequently updated network attacks. The assessed algorithms are: Decision Tree (DT) from the tree-based family, K-nearest neighbor (KNN) from the instance-based family, Multilayer Perceptron (MLP) from the function-based family, and Naive Bayes (NB) from the Bayesian-based family. Initially, the performance of these algorithms is evaluated to identify the most effective family-based classifier for addressing the multiclass imbalanced problem. A brief description of these algorithms is as follows:

1) Decision Tree (DT): A well-known approach used in the field of network intrusion detection. It constructs a hierarchical tree with decision leaves and data element nodes to solve the classification problem. Although [20] has raised concerns about the necessity for numerous splits in a skewed distribution dataset, some researchers [21], [22], [23] have proved the efficiency of DT in this field.
2) K-nearest neighbor (KNN): An instance-based algorithm, KNN classifies dataset instances using Euclidean distance to measure the proximity between training and testing instances [24]. It is simple and robust against noisy data [25], albeit with some efficiency drawbacks, particularly in selecting the optimal "k" value [26]. In our experiment, $k = 10$ was chosen as the most suitable value after fine-tuning.
3) Multilayer Perceptron (MLP): As a neural network, or function-based algorithm, MLP consists of multiple interconnected neuron layers [27], [26]. The number of hidden and output layers determines its structure [28]. In our experiments, we configured the MLP with 100 hidden layers, using the Rectified Linear Unit (ReLU) as the activation function and Adam as the optimizer with a learning rate of 0.001. The maximum number of iterations was set to 200.
4) Naive Bayes (NB): Naive Bayes classifier is a family of simple probabilistic classification algorithms based on Bayes' theorem. In contrast to Bayes theorem, it is designed based on naive assumption that features are independent from each other to simplify the algorithm. In this experiment, we implemented Gaussian variant which uses Gaussian Distribution for the feature values of each class [29]. Instead of solely relying on the Euclidean distance from the class mean, this algorithm takes both into account. Yet, it does have the drawback of only modeling each dimension independently, as it neglects the joint distribution of weight and height [30].

D. Tree-based Ensemble Algorithms

The research employed a selected set of ensemble algorithms, with a specific focus on tree-based families in which the decision tree serves as the primary classifier for these methods. The choice was made due to the decision tree's ability to handle the imbalanced dataset, as was discussed in Section V-A.

The following nine tree-based ensemble algorithms were applied in this study:

1) Bagging: Bagging (Bootstrap Aggregating) is an effective technique that is able to solve the high

variance problem faced by some algorithms, such as decision trees. It involves constructing several trees without prunning and is able to show reliable results [31].

2) Random Forest (RF): An improved version of Bagging that is able to reduce noise, solve outliers, and overfit problems, which are common challenges found in a dataset. By reducing correlations between individual classifiers, RF effectively eliminates and deals with these difficulties, creating a robust and reliable model [31], [32].
3) Extremely Randomized Tree (ExtraTree): As compared to RF, this algorithm, which is also an evolution of Bagging, constructs random trees by using the instances of the dataset [31]. An enhanced robustness and increased resilience were able to be guaranteed with this intentional injection of diversity, which also strengthened the overall ensemble.
4) Adaptive Boosting (AdaBoost): It is a weighted-assigned ensemble algorithm that modifies the weight of the instances of the dataset dynamically. By doing so, the algorithm is able to allocate attention strategically during the construction of subsequent models, which enhances its capabilities in handling the different complexities present in the network intrusion data.
5) Gradient Boosting (GraBoost): GraBoost is a very complex and sophisticated ensemble algorithm. Despite its complexity, GraBoost stands as one of the most formidable ensemble methods, particularly distinguished for its efficacy in elevating classification performance amidst the challenges posed by imbalanced datasets [31].
6) Histogram-based Gradient Boosting (HistGraBoost): HistGraBoost, an innovative boosting algorithm, addresses a key limitation of the GB algorithm—lengthy training times on large datasets. This is remedied by discretizing continuous input variables, optimizing efficiency. The critical hyperparameter is the learning rate, with extensive optimization through multiple rounds of tuning [33].
7) Extreme Gradient Boosting (XGBoost): XGBoost, a highly scalable tree boosting system, is renowned for state-of-the-art performance in machine learning challenges. Leveraging sparsity-aware techniques and insights into cache access patterns, data compression, and sharding, XGBoost excels in efficiency. It outperforms comparable systems on large datasets while optimizing resource utilization [34].
8) Light Gradient Boosting Machine (LightGBM): LightGBM, a robust framework implementing Gradient Boosted Decison Tree (GBDT), emphasizes efficient parallel training. With features like accelerated training speed, reduced memory consumption, and support for distribution, LightGBM excels in accuracy and swift processing of massive datasets [35].
9) Category Gradient Boosting (CatBoost): CatBoost, an innovative algorithm, automatically treats categorical features as numerical characteristics. Utilizing a combination of category features enriches feature dimensions, while a perfectly symmetrical tree model reduces overfitting, enhancing accuracy and generalizability [36]. This categorical-centric approach positions CatBoost as a sophisticated solution for handling categorical features within gradient boosting algorithms.

The strategic evaluation of these ensemble algorithms is necessary to tackle the intricate challenges posed by imbalanced datasets. Section V-A will showcase the preliminary outcomes that demonstrate the proficiency of each traditional machine learning algorithm. This will provide an understanding of the factor influences the selection of algorithms in this research project.

IV. Evaluation Metrics

The F1-score is crucial in evaluating the effectiveness of the tree-based ensemble methods used in this work. The F1-score is instrumental in situations where imbalances are common. It offers a balanced evaluation that considers the constraints of accuracy, which can often give too much weight to classes with a high number of instances or overlook differences within classes. The decision to prioritize the F1-score as the primary evaluation metric is based on its inherent insensitivity to class imbalance. It is a suitable tool for assuring an unbiased and impartial assessment [37].

The F1-score is mathematically defined in Eq.(3).

$$\text{F1-score} = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{3}$$

where precision refers to the measure of how accurate positive predictions are. It is calculated by dividing the number of true positive (TP) by the sum of true positive and false positive (FP) predictions. Recall, also known as sensitivity or the true positive rate, gauges the ability to accurately identify positive instances, measured as the ratio of true positive (TP) predictions to the sum of true positive and false negative (FN) predictions.

The F1-score ranges between 0 and 1, with 1 denoting optimal performance. A higher F1-score signifies superior performance, achieving an equilibrium between precision and recall [38]. This paper also reports the F1-score for each class to provide insights into class-specific performance. Understanding how well the model performs for each class is essential in real-world applications. Beyond complementing the F1-score per class, we also provided the Weighted F1-score and the Macro Average F1-score to analyze the overall performance of the algorithms.

The macro average F1-score assigns equal importance to each class, so preventing the dominance of larger classes from overshadowing the performance of smaller ones. Additionally, it offers valuable insights into the performance of each class separately, which is particularly useful in situations when the performance of each class is of utmost importance. The weighted F1-score enables the allocation of distinct weights to classes according to their significance, so effectively addressing imbalances in a manner better to macro averaging. In this experiment, the weights are allocated according to the sizes of the classes. Note that this research excludes the use of micro average F1-score due to its susceptibility to being

influenced by classes with bigger sizes, which may result in the performance of smaller classes being disregarded.

The equation for the Weighted F1-score is provided in Eq. (4), whereas the equation for the Macro Average F1-score is given in Eq. (5).

$$\text{Weighted F1-score} = \frac{\sum_i \text{F1-score}_i \times \text{Weight}_i}{\sum_i \text{Weight}_i} \quad (4)$$

where F1-score_i is the F1-score for class i, and Weight_i is the weight assigned to class i which refers to the proportion of instances in class i in the dataset.

$$\text{Macro Average F1-score} = \frac{\sum_{i=1}^{N} \text{F1-score}_i}{N} \quad (5)$$

where F1-score_i is the F1-score for class i and N is the number of classes.

In addition to the F1-score, we also deliver the results based on the accuracy value.

$$Accuracy(\%) = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \quad (6)$$

where $TP + TN$ denotes the total number of instances correctly classified in that class, and $TP + TN + FP + FN$ represent the total number of instances in that class in the testing dataset.

In summary, the chosen evaluation measures, with the F1-score as the leading indicator, provide a thorough and informative insight for evaluating the effectiveness of the measured algorithms on imbalanced multiclass datasets. The F1-score enables us to assess the efficacy better. The method aimed to improve classification performance, especially for rare classes in real-world situations.

V. Results and Discussion

A. Preliminary Results

Four machine learning methods from different families were utilized to train the classifier on the training dataset and then test it on the testing dataset. The algorithmic selection consisted of representatives from various families, including Decision Tree (DT) from the tree-based family, K-nearest neighbour (KNN) from the instance-based family, Multilayer Perceptron (MLP) from the functions-based family, and Naive Bayes (NB) from the Bayesian-based family. The purpose of this selection process was to identify the most suitable machine learning algorithm families for tackling the complex task of multiclass imbalanced classification.

The thorough assessment, as depicted in Tables IV and V, showcases the results of our study. The performance of the machine learning algorithms varies considerably across different attack categories. The Decision Tree (DT) approach demonstrated the maximum accuracy in the "Generic" class with 98.31% and the "Normal" class with 91.26%. Nevertheless, the Multilayer Perceptron (MLP) exhibited higher accuracy in the "Normal" class with 99.60%. When working with classes that have a small number of instances, like "Worms" and "Shellcode," even a single misclassification might have a large impact on the accuracy results due to the low size of the sample. The results show that in overall, the MLP exhibits inferior accuracy compared to other algorithms, indicating that it may encounter difficulties handling the complex nature of specific attack patterns. The mean accuracy for DT is 48.55%; for KNN, it is 31.57%; for MLP, it is 3.08%; and for NB, it is 16.76% across all attack classes. These results provide a perspective on the performance of algorithms. Still, the interpretation should be done carefully, considering the presence of class imbalances.

The tree-based classifiers, specifically the Decision Tree (DT) with the highest Weighted F1-score of 0.80, clearly outperformed the algorithms from other families regarding overall F1-scores and accuracy. The Weighted F1-score of KNN is 0.65, which is the second highest among the models. Naive Bayes is entirely ineffective in detecting Analysis and Denial of Service (DoS) threats. The KNN, MLP and NB were facing difficulties in accurately detecting and categorizing threats such as Analysis, Backdoors, Shellcode, and Worms as the F1-score for each class is below 0.15. The MLP exhibited poor results with all classes except for the "Normal" class, achieving an F1-score of 0.1 or lower. This demonstrates that the MLP is only capable of recognizing regular network traffic and lacks the ability to identify network attacks.

The experiment strongly suggests a greater efficacy of the decision tree, as evidenced by the substantial findings. Furthermore, a per-class analysis reveals that it surpassed other traditional algorithms in performance for all classes. This discovery yields a vital inference: tree-based algorithms demonstrate superior performance when addressing multiclass imbalanced classification issues compared to conventional techniques. This analysis clarifies the reasoning for choosing tree-based ensemble techniques and explores the further findings.

B. The Evaluation of Tree-based Ensemble Algorithms Performance

In this part, we will further investigate the most appropriate tree-based technique for practical application in the problem of multiclass imbalanced classification. This analysis is based on the findings presented in Section V-A and focuses on comparing different tree-based ensemble algorithms. This section offers an extended analysis of the tree-based ensemble algorithms employed in this study. Similar to the previous section (Section V-A), the selected tree-based ensemble algorithms were evaluated based on the accuracy, F1-score per class, Weighted F1-score and Macro Average F1-score as explained in Section IV.

The tables labelled as VI and VII include valuable information about how well these ensemble approaches, built on trees, perform in classifying instances for each category. XGBoost outperforms other algorithms, demonstrating exceptional results across the majority of classes with a classification accuracy of 50%. However, it is essential to note that there are outliers within the Analysis, Backdoor, and Denial of Service (DoS) attack categories. All the algorithms used in this study exhibit reduced accuracy in those instances.

TABLE IV. ACCURACY RESULTS FOR FOUR CONVENTIONAL MACHINE LEARNING ALGORITHMS FOR EACH ATTACK CLASS IN UNSW-NB15 DATASET

Attack class	Number of instances per classes in test dataset	Accuracy (%)			
		Decision Tree(DT)	K-nearest neighbor(KNN)	Multilayer Perceptron(MLP)	Naive Bayes(NB)
Analysis	803	**12.07**	2.74	0.62	0.00
Backdoor	699	**9.16**	0.14	0.29	0.29
DoS	4,906	**33.87**	30.54	6.38	0.18
Exploits	13,358	**73.87**	50.94	0.91	2.00
Fuzzers	7,274	**58.62**	26.23	1.94	21.37
Generic	17,661	**98.31**	97.42	0.05	97.99
Normal	27,900	91.26	79.73	**99.60**	45.67
Reconnaissance	4,196	**75.92**	33.92	0.00	43.07
Shellcode	453	**59.02**	7.95	0.00	1.33
Worms	52	**53.85**	0.00	0.00	3.85
Average		**48.55**	31.57	3.08	16.76

TABLE V. F1-SCORE RESULTS FOR FOUR CONVENTIONAL MACHINE LEARNING ALGORITHMS FOR EACH ATTACK CLASS IN UNSW-NB15 DATASET

Attack class	Number of instances per classes in test dataset	F1-scores			
		Decision Tree(DT)	K-nearest neighbor(KNN)	Multilayer Perceptron(MLP)	Naive Bayes(NB)
Analysis	803	**0**.17	0.05	0.01	0.00
Backdoor	699	**0.15**	0.00	0.01	0.01
DoS	4,906	**0.33**	0.30	0.10	0.00
Exploits	13,358	**0.69**	0.43	0.02	0.04
Fuzzers	7,274	**0.61**	0.30	0.04	0.25
Generic	17,661	**0.99**	0.98	0	0.64
Normal	27,900	**0.91**	0.78	0.54	0.61
Reconnaissance	4,196	**0.82**	0.49	0.00	0.15
Shellcode	453	**0.59**	0.12	0.00	0.02
Worms	52	**0.47**	0.00	0.00	0.01
Macro Average F1-score		**0.63**	0.31	0.06	0.16
Weighted F1-score		**0.80**	0.65	0.21	0.40

For the Analysis class, RF, XGBoost, and Bagging perform better than other models, attaining the highest F1-scores of 0.19. The F1-scores for DT and ExtraTree are both 0.17. XGBoost outperforms other models in terms of F1-score with a value of 0.17 for the Backdoor class. Bagging, Decision Trees (DT), and Random Forest (RF) exhibit similar performance, with F1-scores almost equal to 0.16. ExtraTree and DT demonstrate the most robust performance in terms of F1-score (0.33) for the DoS attack. Bagging and Random Forest (RF) perform strongly, achieving F1-scores ranging from 0.30 to 0.33. We found that XGBoost and CatBoost achieved the highest F1-score of 0.74 in the Exploit class. Other methods such as Bagging, RF, ExtraTree, GraBoost, and HistGraBoost produce comparable scores ranging from 0.72 to 0.73. For Fuzzer attacks, Bagging demonstrates the most significant F1-scores, precisely 0.66. Other algorithms, such as Random Forest (RF) and ExtraTree, attain scores about equal to 0.65. The results also show that most algorithms perform exceptionally in categorizing generic traffic, as evidenced by their high F1-scores of approximately 0.99. For Normal traffic, the results show that XGBoost, Bagging, RF, and ExtraTrees exhibit the most outstanding F1-scores of 0.93. Bagging and XGBoost achieve the highest F1-scores for the Reconnaissance and Shellcode classes, with 0.84 and 0.69, respectively. Regarding the class with the smallest number of samples, Worms, it is observed that XGBoost attains the highest F1-scores, precisely 0.63.

The F1-score data shown in Table VI demonstrates that advanced ensemble approaches, namely Bagging, Random Forest, XGBoost, and ExtraTrees, exhibited superior performance compared to the conventional Decision Tree. Bagging, Random Forest, XGBoost, and ExtraTree obtained the highest Weighted F1-score. Bagging attains its highest macro average F1-score of 0.63, denoting outstanding overall performance. Additional algorithms, such as XGBoost and DT, exhibit similar performance with macro F1-scores ranging from 0.60 to 0.63. Bagging, Random Forest, ExtraTrees, and XGBoost demonstrate superior performance in addressing imbalanced classes, as evidenced by their highest weighted F1-scores of 0.82. Several other algorithms, such as DT, GraBoost, HistGraBoost, LightGBM, and CatBoost, achieve weighted F1-scores in the range of $0.79-0.80$. The weighted F1-score offers a more accurate evaluation by taking into account both the performance of each class and the distribution of classes.

The results suggest that the algorithm's efficacy is significantly influenced by the distinctive properties of each class, thereby necessitating an understanding of attack characteristics. After analyzing Table VII, it is evident that XGBoost emerges as the most robust choice, outperforming all other tree-based ensemble algorithms by attaining the highest F1-score for eight out of ten classes. While Bagging demonstrates comparable Weighted F1-scores and Macro Average F1-scores, an in-depth analysis indicates that XGBoost surpasses in particular categories (except for DoS and Fuzzers). Furthermore, the accuracy results presented in Table VIII conclusively indicate that XGBoost exceeded other algorithms in terms of accuracy, with Bagging being an equally strong candidate.

The data presented in this paper suggest that XGBoost and Bagging is the best tree-based ensemble method for multiclass imbalanced classification in the particular scenario of network attack detection. The study's results emphasize the algorithm's effectiveness in tackling the difficulties posed by imbalanced datasets, making it a highly appropriate choice for practical

TABLE VI. F1-SCORE RESULTS FOR DECISION TREE AND NINE TREE-BASED ENSEMBLE MACHINE LEARNING ALGORITHMS FOR EACH ATTACK CLASS IN THE UNSW-NB15 DATASET

Attack class	Number of instances per classes in test dataset	F1-scores									
		DT	Bagging	RF	ExtraTree	AdaBoost	GraBoost	HistGraBoost	XGBoost	LightGBM	CatBoost
Analysis	803	0.17	**0.19**	**0.19**	0.17	0.11	0.08	0.13	**0.19**	0.13	0.09
Backdoor	699	0.16	0.16	0.16	0.15	0.01	0.12	0.14	**0.17**	0.09	0.12
DoS	4,906	**0.33**	0.32	0.30	**0.33**	0.08	0.12	0.08	0.20	0.25	0.17
Exploits	13,358	0.69	0.72	0.72	0.72	0.22	0.72	0.73	**0.74**	0.70	**0.74**
Fuzzers	7,274	0.61	**0.66**	0.65	0.65	0.29	0.58	0.52	0.64	0.59	0.61
Generic	17,661	**0.99**	**0.99**	**0.99**	**0.99**	0.93	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
Normal	27,900	0.91	**0.93**	**0.93**	**0.93**	0.60	0.91	0.91	**0.93**	0.90	0.92
Reconnaissance	4,196	0.82	**0.84**	0.83	0.82	0.51	0.83	0.83	**0.84**	0.80	0.83
Shellcode	453	0.61	**0.69**	0.67	0.64	0.39	0.65	0.53	**0.69**	0.52	0.61
Worms	52	0.47	0.62	0.20	0.20	0.01	0.17	0.14	**0.63**	0.13	0.17
Macro Average F1-score		0.60	**0.63**	0.59	0.58	0.33	0.54	0.53	0.62	0.54	0.55
Weighted F1-score		0.80	**0.82**	**0.82**	**0.82**	0.53	0.79	0.79	**0.82**	0.79	0.80

TABLE VII. ALGORITHMS WITH HIGHEST F1-SCORE PER CLASS

Class	DT	Bagging	RF	ExtraTree	AdaBoost	GraBoost	HistGraBoost	XGBoost	LightGBM	CatBoost
Analysis		✓(0.19)	✓(0.19)					✓(0.19)		
Backdoor								✓(0.17)		
DoS	✓(0.33)			✓(0.33)						
Exploits								✓(0.74)		✓(0.74)
Fuzzers		✓(0.66)								
Generic	✓(0.99)	✓(0.99)	✓(0.99)	✓(0.99)		✓(0.99)	✓(0.99)	✓(0.99)		✓(0.99)
Normal		✓(0.93)	✓(0.93)	✓(0.93)				✓(0.93)		
Reconnaissance		✓(0.84)						✓(0.84)		✓(0.83)
Shellcode		✓(0.69)						✓(0.69)		
Worms								✓(0.63)		

implementation in cybersecurity and network intrusion detection.

VI. CONCLUSION AND FUTURE WORKS

The findings indicate that tree-based ensemble methods, including Bagging, Random Forest, XGBoost, and ExtraTrees, have achieved a high Weighted F1-score, despite the constraint of an imbalanced training dataset. These qualities make them very suitable for identifying network intrusions in the UNSW-NB15 dataset. XGBoost surpassses other tree-based algorithms in terms of per-class F1-scores, which is a useful performance measure for addressing multiclass imbalance problems. Nevertheless, the overall accuracy of XGBoost is about equivalent to that of Bagging. These findings confirm that XGBoost is the most effective approach for addressing multiclass imbalance classification, with Bagging being the most viable option. In summary, the results highlight the effectiveness of Decision Tree (DT) and tree-based ensemble algorithms in handling the problem of imbalanced multi-class datasets.

This study has offered valuable insights into the efficacy of tree-based ensemble algorithms for multiclass imbalanced classification in network intrusion detection. However, it is crucial to recognise the underlying constraints and difficulties. Although ensemble strategies have been used to address class imbalance, the problem persists. The disproportionate allocation of classes, specifically pertaining to minority categories such as Analysis, Backdoor, and Denial of Service (DoS), still poses substantial difficulties in detecting these classes.

Beyond the difficulties and constraints, the outcomes provide a solid groundwork for future studies in this domain. Further investigations into feature engineering, advanced sampling approaches, or algorithmic adaptations that can effectively improve the identification of minority class occurrences should be conducted. More advanced algorithms with adaptive sampling capable of dealing with data changes over time are likely needed. These efforts are necessary for creating resilient and flexible solutions to address the constantly evolving cyber-attack scenario.

Future research must consider developing and using domain-specific evaluation metrics that improve the interpretation of algorithmic performance in situations with imbalances across many classes. This evaluation metric should surpass conventional performance metrics such as the F1-score. It should provide a comprehensive assessment considering the trade-off between false positives and false negatives in various domains. This will result in a more thorough evaluation of performance.

ACKNOWLEDGMENT

The authors express their gratitude for the support received from the Ministry of Higher Education Malaysia through the Fundamental Research Grant Scheme (FRGS), awarded under grant number FRGS/1/2018/ICT02/UNIMAP/02/6.

REFERENCES

[1] I. Lella, E. Tsekmezoglou, M. Theocharidou, E. Magonara, A. Malatras, R. S. Naydenov, and C. Ciobanu, "Enisa threat landscape 2023," 2023.

[2] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Computers & Security*, vol. 86, pp. 147–167, 2019.

[3] K. M. Hasib, M. S. Iqbal, F. M. Shah, J. Al Mahmud, M. H. Popel, M. I. H. Showrov, S. Ahmed, and O. Rahman, "A survey of methods for managing the classification and solution of data imbalance problem," *Journal of Computer Science*, vol. 16, p. 1546–1557, Nov. 2020.

[4] S. K. M, Devidas, S. N. Pai, S. Kolekar, V. Pai, and B. R, "Use of machine learning and random oversampling in stroke prediction," in *2022 International Conference on Artificial Intelligence and Data Engineering (AIDE)*, pp. 331–337, 2022.

TABLE VIII. ACCURACY RESULTS FOR DECISION TREE AND NINE TREE-BASED ENSEMBLE MACHINE LEARNING ALGORITHMS FOR EACH ATTACK CLASS IN THE UNSW-NB15 DATASET

Attack class	Number of instances per classes in test dataset	Accuracy (%)									
		DT	Bagging	RF	ExtraTree	AdaBoost	GraBoost	HistGraBoost	XGBoost	LightGBM	CatBoost
Analysis	803	12.08	11.08	10.96	10.84	**62.15**	4.36	7.11	10.35	8.48	4.99
Backdoor	699	**9.73**	9.30	9.01	8.87	4.44	6.29	8.01	9.16	5.58	6.72
DoS	4,906	**33.93**	28.52	26.17	31.60	4.84	6.84	4.33	12.87	18.23	10.43
Exploits	13,358	73.84	81.15	81.51	79.46	14.59	90.38	**92.17**	90.70	83.88	90.65
Fuzzers	7,274	58.48	**61.48**	60.69	60.16	21.10	50.67	40.17	58.09	52.95	57.85
Generic	17,661	98.23	**98.33**	97.97	97.94	89.78	97.67	97.93	98.28	97.98	97.94
Normal	27,900	91.32	94.87	94.44	94.55	51.80	92.66	**95.75**	95.10	90.20	94.14
Reconnaissance	4,196	76.03	76.34	76.23	75.13	**79.62**	76.77	75.89	76.94	76.44	76.11
Shellcode	453	61.16	70.87	66.46	60.27	29.15	65.79	54.98	**73.08**	56.08	59.83
Worms	52	57.70	**67.32**	13.46	13.46	59.63	48.09	38.47	**67.32**	32.70	9.63
Correctly Identified Instances		62264	64251	63889	63793	38056	62812	63015	**64598**	61998	63959
Accuracy (%)		80.55%	83.12%	82.65%	82.52%	49.23%	81.25%	81.52%	**83.57%**	80.20%	82.48%

[5] F. Kamalov, H.-H. Leung, and A. K. Cherukuri, "Keep it simple: random oversampling for imbalanced data," in *2023 Advances in Science and Engineering Technology International Conferences (ASET)*, pp. 1–4, 2023.

[6] J. Hancock, T. M. Khoshgoftaar, and J. M. Johnson, "The effects of random undersampling for big data medicare fraud detection," in *2022 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, pp. 141–146, 2022.

[7] Q. Dai, J.-w. Liu, and Y. Liu, "Multi-granularity relabeled undersampling algorithm for imbalanced data," *Applied Soft Computing*, vol. 124, p. 109083, 2022.

[8] M. Muntasir Nishat, F. Faisal, I. Jahan Ratul, A. Al-Monsur, A. M. Ar-Rafi, S. M. Nasrullah, M. T. Reza, and M. R. H. Khan, "A comprehensive investigation of the performances of different machine learning classifiers with smote-enn oversampling technique and hyperparameter optimization for imbalanced heart failure dataset," *Scientific Programming*, vol. 2022, pp. 1–17, 2022.

[9] S. Demir and E. K. Şahin, "Evaluation of oversampling methods (over, smote, and rose) in classifying soil liquefaction dataset based on svm, rf, and naïve bayes," *Avrupa Bilim ve Teknoloji Dergisi*, no. 34, pp. 142–147, 2022.

[10] S. Abokadr, A. Azman, H. Hamdan, and N. Amelina, "Handling imbalanced data for improved classification performance: Methods and challenges," in *2023 3rd International Conference on Emerging Smart Technologies and Applications (eSmarTA)*, pp. 1–8, 2023.

[11] I. Fosić, D. Žagar, and K. Grgić, "Network traffic verification based on a public dataset for ids systems and machine learning classification algorithms," in *2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)*, pp. 1037–1041, 2022.

[12] N. Abedzadeh and M. Jacobs, "A survey in techniques for imbalanced intrusion detection system datasets," *International Journal of Computer and Systems Engineering*, vol. 17, no. 1, pp. 9 – 18, 2023.

[13] M. Kim and K.-B. Hwang, "An empirical evaluation of sampling methods for the classification of imbalanced data," *PLoS One*, vol. 17, no. 7, p. e0271260, 2022.

[14] R. Raj and S. Singh Kang, "Mitigating ddos attack using machine learning approach in sdn," in *2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, pp. 462–467, 2022.

[15] P. S. Patil, S. L. Deshpande, G. S. Hukkeri, R. H. Goudar, and P. Siddarkar, "Prediction of ddos flooding attack using machine learning models," in *2022 Third International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*, pp. 1–6, 2022.

[16] M. A. Talukder, K. F. Hasan, M. M. Islam, M. A. Uddin, A. Akhter, M. A. Yousuf, F. Alharbi, and M. A. Moni, "A dependable hybrid machine learning model for network intrusion detection," *Journal of Information Security and Applications*, vol. 72, p. 103405, 2023.

[17] S. K. Naing and T. T. Thwel, "A study of ddos attack classification using machine learning classifiers," in *2023 IEEE Conference on Computer Applications (ICCA)*, pp. 108–112, 2023.

[18] V. Santhi, J. Priyadharshini, M. Swetha, and K. Dhanavandhana, "A hybrid feature extraction method with machine learning for detecting the presence of network attacks," in *2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS)*, pp. 454–459, 2023.

[19] R. Wen and K. Zhang, "Research on automated classification method of network attacking based on gradient boosting decision tree," in *2022 International Conference on Machine Learning and Knowledge Engineering (MLKE)*, pp. 72–76, 2022.

[20] F. Shakeel, A. S. Sabhitha, and S. Sharma, "Exploratory review on class imbalance problem: An overview," in *8th International Conference on Computer Communications and Networks Technologies (ICCCNT)*, 2017.

[21] N. Elmrabit, F. Zhou, F. Li, and H. Zhou, "Evaluation of machine learning algorithms for anomaly detection," in *International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, pp. 1–6, 2020.

[22] D. Kurniabudi, D. Stiawan, M. Y. Bin Bin Idris, A. M. Bamhdi, and R. Budiarto, "Improving the anomaly detection by combining pso search methods and j48 algorithm," *IEEE Explore for Emerging Cyber Security and Information Systems*, pp. 119–126, 2020.

[23] D. Kurniabudi, D. Stiawan, M. Y. Bin Bin Idris, A. M. Bamhdi, and R. Budiarto, "Cicids-2017 dataset feature analysis with information gain for anomaly detection," *IEEE Access*, vol. 8, pp. 132911–132921, 2020.

[24] N. Ali, D. Neagu, and P. Trundle, "Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets," *SN Applied Sciences*, vol. 1, pp. 1–15, 2019.

[25] P. K. Syriopoulos, N. G. Kalampalikis, S. B. Kotsiantis, and M. N. Vrahatis, "k nn classification: a review," *Annals of Mathematics and Artificial Intelligence*, pp. 1–33, 2023.

[26] E. Y. Boateng, J. Otoo, and D. A. Abaye, "Basic tenets of classification algorithms k-nearest-neighbor, support vector machine, random forest and neural network: a review," *Journal of Data Analysis and Information Processing*, vol. 8, no. 4, pp. 341–357, 2020.

[27] H. Taud and J. Mas, "Multilayer perceptron (mlp)," *Geomatic approaches for modeling land change scenarios*, pp. 451–455, 2018.

[28] Q. Jiang, L. Zhu, C. Shu, and V. Sekar, "Multilayer perceptron neural network activated by adaptive gaussian radial basis function and its application to predict lid-driven cavity flow," *Acta Mechanica Sinica*, pp. 1–16, 2021.

[29] A. H. Jahromi and M. Taheri, "A non-parametric mixture of gaussian naive bayes classifiers based on local independent features," in *2017 Artificial Intelligence and Signal Processing Conference (AISP)*, pp. 209–212, 2017.

[30] R. D. Raizada and Y.-S. Lee, "Smoothness without smoothing: why gaussian naive bayes is not naive for multi-subject searchlight studies," *PloS one*, vol. 8, no. 7, p. e69566, 2013.

[31] J. Brownlee, *Machine Learning Mastery with Python: Understand Your Data, Create Accurate Models and Work Projects End-To-End*. v1.19 ed., 2020.

[32] R. Saravanan and P. Sujatha, "A state of art techniques on machine learning algorithms: A perspective of supervised learning approaches in data classification," in *2019 2nd International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 945–949, 2019.

[33] G. Marvin, L. Grbčić, S. Družeta, and L. Kranjčević, "Water distribution network leak localization with histogram-based gradient boosting," *Journal of Hydroinformatics*, vol. 25, no. 3, pp. 663–684, 2023.

[34] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, ACM, Aug. 2016.

[35] C. Ma, Y. Chi, D. Hao, and X. Ji, "A new approach based on feature selection of light gradient boosting machine and transformer to predict circrna-disease associations," *IEEE Access*, vol. 11, pp. 47187–47201, 2023.

[36] M. Luo, Y. Wang, Y. Xie, L. Zhou, J. Qiao, S. Qiu, and Y. Sun, "Combination of feature selection and catboost for prediction: The first application to the estimation of aboveground biomass," *Forests*, vol. 12, no. 2, p. 216, 2021.

[37] J. Wu, Z. Zhao, C. Sun, R. Yan, and X. Chen, "Learning from class-imbalanced data with a model-agnostic framework for machine intelligent diagnosis," *Reliability Engineering & System Safety*, vol. 216, p. 107934, December 2021.

[38] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc.", 2022.