STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **12**, March 2024, pp 547–570.
Published online in International Academic Press (www.IAPress.org)

Comparative Evaluation of Imbalanced Data Management Techniques for Solving Classification Problems on Imbalanced Datasets

Tanawan Watthaisong [1], Khamron Sunat [1,*], Nipotepat Muangkote [2]

[1] *Department of Computer Science, College of Computing, Khon Kaen University, 40000, Thailand*
[2] *Department of Smart City Management and Digital Innovation, Mahasarakham Business School, Mahasarakham University, Mahasarakham, 44150, Thailand*

Abstract Dealing with imbalanced data is crucial and challenging when developing effective machine-learning models for data classification purposes. It significantly impacts the classification model's performance without proper data management, leading to suboptimal results. Many methods for managing imbalanced data have been studied and developed to improve data balance. In this paper, we conduct a comparative study to assess the influence of a ranking technique on the evaluation of the effectiveness of 66 traditional methods for addressing imbalanced data. The three classification models, i.e., Decision Tree, Random Forest, and XGBoost, act as classification models. The experimental settings have been divided into two segments. The first part evaluates the performance of various imbalanced dataset handling methods, while the second part compares the performance of the top 4 oversampling methods. The study encompasses 50 separate datasets: 20 retrieved from the UCI repository and 30 sourced from the OpenML repository. The evaluation is based on F-Measure and statistical methods, including the Kruskal-Wallis test and Borda Count, to rank the data imbalance handling capabilities of the 66 methods. The SMOTE technique is the benchmark for comparison due to its popularity in handling imbalanced data. Based on the experimental results, the MCT, Polynom-fit-SMOTE, and CBSO methods were identified as the top three performers, demonstrating superior effectiveness in managing imbalanced datasets. This research could be beneficial and serve as a practical guide for practitioners to apply suitable techniques for data management.

Keywords Imbalanced data handling, Oversampling Technique, Machine Learning, Classification, Synthetic Minority Over-sampling Technique (SMOTE)

DOI: 10.19139/soic-2310-5070-1890

1. Introduction

*I*mbalanced data denotes a dataset in which certain classes possess a considerably larger instance than others. The class with the highest frequency is recognized as the majority class, while the class with the lowest occurrence is denoted as the minority class [1, 2, 3, 4, 5, 6, 7]. The literature emphasizes the significant data imbalance that arises when dealing with a substantially smaller proportion of specific cases, in contrast to the usual cases [8, 9, 10, 11, 12]. In the existing literature, numerous empirical comparisons of sampling methods provide valuable guidance to researchers and practitioners for enhancing classification or regression models in real-world applications [1, 2, 5, 7, 12, 13, 14, 15, 16, 17]. We classified the sampling methods into three distinct ways: 1) undersampling, 2) oversampling, and 3) hybrid approaches. Undersampling techniques involve reducing the number of instances in the majority class [18, 19, 20]. On the other hand, oversampling methods include increasing the number of minority samples through random resampling of the original minority class [21, 22, 23, 24]. In

*Correspondence to: Khamron Sunat (Email: skhamron@kku.ac.th (K. Sunat)). Department of Computer Science, College of Computing, Khon Kaen University, 40000, Thailand.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

comparison, hybrid methodologies combine oversampling and undersampling techniques [1, 25, 26, 27, 28]. For example, in 2014, Sandhan and Choi [26] introduced a hybrid method that combines the oversampling of the minority class with the undersampling of the majority class to create a classifier ensemble. In 2019, Eleedy and Atiya [27] presented a comprehensive analysis of SMOTE for handling class imbalance by hybridizing both undersampling and oversampling techniques. Additionally, other hybrids have been introduced, e.g., in 2020, Raghuwanshi and Shukla [28] presented SMOTECSELM, an acronym for Synthetic Minority Over-Sampling TEchnique based on Class-Specific Extreme Learning Machine, which integrates ELM with SMOTE. Finally, in 2024, Carla Vairetti et al. [25] introduced SMOTENN, a novel hybrid undersampling–oversampling technique for imbalanced classification in Big Data. This method uniquely combines intelligent undersampling and oversampling within a MapReduce framework, performing both tasks in a single pass over the data.

Over the past two decades, extensive research has been conducted into the broad spectrum of methods addressing the challenge of imbalanced data management. These approaches can be broadly categorized into two major groups: 1) the first group related to the SMOTE technique and 2) the second group unrelated to SMOTE. The first group includes techniques related to the SMOTE method. An important example is the Synthetic Minority Over-sampling Technique (SMOTE), introduced by Chawla et al. [29]. In SMOTE, synthetic samples are generated instead of using oversampling with replacement.

Furthermore, SMOTE has been subject to further investigation to enhance its performance. For instance, Cluster-SMOTE, introduced by Cieslak et al. in 2006 [30], utilizes a clustering-based approach for oversampling by generating synthetic samples. Polynom-fit-SMOTE, proposed by Gazzah and Amara in 2008 [31], operates within the feature space to populate the minority feature subspace. This method generates an appropriate number of synthetic instances using various methods aligned with the oversampling rate. G-SMOTE, presented by Sandhan and Choi in 2014 [26], introduces a hybrid sampling technique. A noteworthy strategy employed in this approach is bootstrapping, which effectively addresses the complex challenge posed by highly skewed data distributions. Three enhancements of SMOTE denoted as SMOTE-Out, SMOTE-Cosine, and Selected-SMOTE, proposed by Koto [32], have been developed to address scenarios not comprehensively covered by the original SMOTE technique. In 2023, Ruijuan Liu [18] introduced the SMOTE-RD technique, incorporating a noise filter based on relative density (RD) into SMOTE. This enhancement aims to eliminate noise generation and address the issue of class imbalance. Lastly, in 2023, Ruizhi Zhang et al. [22] introduced KDENDS_SMOTE, an enhanced synthetic minority oversampling technique using kernel density estimation and neighbor density selection. This method tackles challenges in traditional oversampling by mitigating issues with uncontrollable synthetic sample positions that may worsen data overlap and degrade classification performance. The second group, outside SMOTE, encompasses approaches like the Adaptive Synthetic Sampling Approach (ADASYN), as introduced by He and Garcia in 2009 [13], which dynamically generates synthetic data samples based on their inherent learning complexities. This approach effectively addresses challenges arising from imbalanced datasets. The Cluster-Based Synthetic Oversampling (CBSO) method, proposed by Barua and Murase in 2011 [21], builds upon ADASYN, integrating established synthetic oversampling methodologies with a clustering-based data generation approach. In 2011, Francisco Fernández-Navarro et al. [33] introduced a dynamic oversampling procedure integrated into a memetic algorithm (MA) to enhance the classification of imbalanced datasets with more than two classes. This procedure optimizes radial basis functions neural networks (RBFNNs). The SVM-balance method, presented by Farquad and Bose in 2012 [34], tackles class imbalance by utilizing the support vector machine (SVM). In this approach, SVM is integrated as a preprocessing step, where the original target values of training data are replaced with predictions from the trained SVM model. The Minority Cloning Technique (MCT) was proposed by Jiang et al. in 2015 [35]. This technique seeks to rebalance the class distribution in training data by replicating each instance from the minority class. In 2022, Moghadam and Ahmadi [9] introduced an innovative clustering technique that utilized the Red Deer Algorithm (RDA). This approach was employed to develop a three-stage clustering-based undersampling method to address the class imbalance challenge. In 2024, Zeyu Teng et al. [24] introduced MLBOTE (Multi-Label Borderline Oversampling Technique), a novel approach for resampling multi-label datasets in the context of imbalanced learning. These methodologies collectively exemplify the diverse strategies to handle class imbalance in machine learning, each offering distinct perspectives and techniques.

As previously discussed, the Synthetic Minority Over-sampling Technique (SMOTE) remains a widely favored approach for addressing imbalanced data. It is highly relevant and extensively applied in various real-world domains due to its simplicity and classifier-independence [18, 36, 37, 38, 39, 40, 41]. Its application spans diverse disciplines, such as predicting nutrient and chlorophyll levels in oil palm (Elaeis guineensis) [39]. To diagnose transformer faults, Rahman Azis Prasojo et al. [40] proposed a precise machine learning-based fault identification model that utilizes the Random Forest algorithm with SMOTE preprocessing. Akira Imakura et al. [41] proposed an anchor data construction technique, an extension of SMOTE, to enhance recognition performance without data leakage risk. They then evaluated its performance in real-world binary and multi-class classification problems. In a previous study, Kovács [14] compared and evaluated various techniques for minority oversampling. However, even though more than a hundred datasets were included in the study's experimental phase, it's crucial to acknowledge that applying these findings to real-world datasets may reveal markedly different characteristics. This discrepancy represents a limitation. Therefore, it is advisable to select an oversampling model carefully. To the best of our knowledge, our research differs from previously published studies. In this paper, we conduct a comparative study to evaluate how a ranking technique influences the effectiveness of traditional methods for addressing imbalanced data. We expanded our study to investigate comparing techniques for handling imbalanced data. We also assessed 66 distinct variations of minority oversampling techniques, employing three diverse classifier types. The main contributions of this paper are summarized as follows:

1. Our objective is to perform a comparative study to investigate the impact of a ranking technique on the evaluation of the effectiveness of 66 traditional methods for addressing imbalanced data. We employ statistical techniques such as the Kruskal-Wallis test proposed by Kruskal and Wallis [42] and Borda Count proposed by Jean-Charles de Borda [43], to compare the rankings of these 66 oversampling methods, aiming to assess their respective abilities.
2. In this paper, we apply the novel metric, Likelihood Ratio Imbalance Degree (LRID), proposed by Rui Zhu [44], to our research. LRID is a robust metric for quantifying class imbalance levels in multi-class data, surpassing the existing Imbalance Ratio (IR) [45, 46] and Imbalance Degree (ID) [47] by accurately measuring the level of class imbalance.
3. The oversampling techniques are evaluated in various classification models. This paper utilizes three well-known classifiers: Decision Tree, Random Forest, and XGBoost. These are versatile supervised learning methods effective in both regression and classification tasks.
4. This paper uses the conventional SMOTE technique as the baseline for comparison. We select and recommend data management techniques that outperform SMOTE by 10% or more in achieving superior results.

The rest of the paper is organized as follows. Section 2 provides a brief overview of relevant prior research. Section 3 offers a short description of the Classifier Models. Section 4 details the performance measures, datasets, and the experimental methodology employed in this study. The experimental outcomes are presented in Section 5, followed by detailed discussions in Section 6. Finally, Section 7 serves as the conclusion of the paper.

2. Related Works

2.1. The Nature of Imbalanced Data

Haibo He and Edwardo A. Garcia [13] mentioned that "*any data set that exhibits an unequal distribution between its classes can be considered imbalanced*." This quote means that a dataset is considered imbalanced if at least one of its classes contains significantly fewer patterns (minority) than the other classes (majority), also known as the class imbalance problem. In literature, numerous suggestions are provided to researchers and practitioners for enhancing machine learning models in real-world applications [1, 13, 14, 48, 49, 50, 51]. For example, in medical datasets, there may be information about individuals at risk of Non-Communicable Diseases (NCDs), such as hypertension, kidney, and heart diseases, with a smaller proportion compared to the overall data of non-afflicted individuals. A notable example in literature [8, 9] is the prediction of failed kidney transplants, where this data imbalance is commonly encountered. As Rui Zhu [44] demonstrated, the class imbalance problem formulation can be represented as a multinomial distribution. In classification modeling, it is learned from the joint distribution, $p(\mathbf{x}, y) = p(y)\, p(\mathbf{x} \mid y)$, where the data vector $x \in \mathbb{R}^{p\times 1}$, y is the data label, and $p(y)$ is the prior knowledge of the probability of label y. Let $\mathbf{y} = [y_1, y_2, \ldots, y_C]$, where C represents the possible outcomes for y. Each outcome y_C is associated with a probability p_C, and $\sum_{c=1}^{C} p_c = 1$. The frequencies of the possible labels, represented as $\mathbf{n} = [n_1, n_2, \ldots, n_C]$, can be modeled using a multinomial distribution denoted as Multinomial$(N, \mathbf{p})$, with parameters N and $\mathbf{p} = [p_1, p_2, \ldots, p_C]$.

2.1.1. Imbalance ratio Imbalance ratio (IR) is widely used to measure the imbalance degree, which is defined as the sample number of the majority class divided by that of the minority class. It can be calculated as follows [44]:

$$\text{IR} = \frac{\hat{p}_{\text{max}}}{\hat{p}_{\text{min}}}, \tag{1}$$

where $\hat{p}_{\text{min}}, \hat{p}_{\text{max}}$ are the minimum and maximum values in $\hat{p}$, respectively.

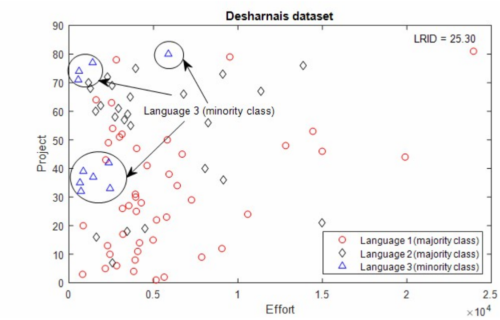
Let's assume there are 1,000 cases in the dataset under study. It's found that there are only ten patients with kidney disease, while the non-afflicted individuals total 990. This situation leads to an imbalance ratio (IR) of about 100. These imbalanced datasets can directly impact the performance of classification models, resulting in learning and prediction errors, particularly as the imbalance ratio (IR) increases. This limitation arises because the model might struggle to accurately predict classes with fewer instances [13, 14, 52]. Imbalanced learning has gained substantial attention in machine learning and artificial intelligence. Traditional Machine Learning models typically perform effectively with balanced distributions. Managing imbalanced data impacts the training of Machine Learning models, enabling them to learn from comprehensive data coverage. This improvement significantly enhances model efficiency, especially in classification tasks that require high accuracy, e.g., it aids medical professionals in making informed decisions when screening individuals at risk of various NCDs [53, 54]. In fact, in medical classification problems, datasets are often imbalanced. Using accuracy as a measurement may not be suitable. In this paper, the F-measure is utilized for imbalanced datasets, as it is more ideal than accuracy.

2.1.2. Likelihood ratio imbalance degree Although the Imbalanced Ratio (IR) is suitable for binary-class data, it may not provide a comprehensive representation of multi-class imbalance because it disregards information related to between-class distributions. The likelihood ratio imbalance degree (LRID), introduced by Rui Zhu [44], is a reliable measure of class-imbalance that surpasses the imbalance ratio (IR) [45, 46] and imbalance degree (ID) [47] in quantifying the extent of class-imbalance in multi-class data. In LRID, the distance metric used for assessing the extent of imbalance is excluded due to its significant influence on the ID metric—instead, the statistical inference technique known as the likelihood-ratio (LR) test is employed.

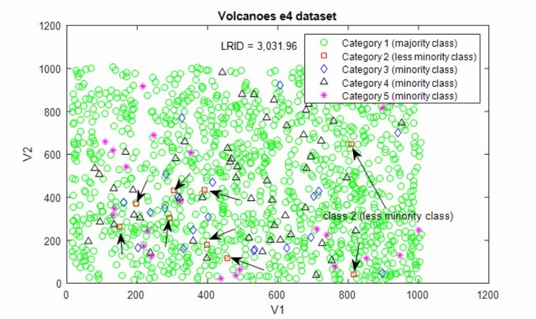
Assuming a dataset with C classes and $\mathbf{n} = [n_1 + n_2 + \ldots + n_C]$, the LR test is employed for the multinomial distribution Multinomial$(N, \mathbf{p})$ to test the null hypothesis that the parameters p are equal to predefined values, where N represents the total number of observations $\sum_{c=1}^{C} n_c$. In this context, the objective is to examine whether the parameter p can be effectively fitted by b, indicating a balanced class distribution. The hypothesis testing is $H_0 : \mathbf{p} = \mathbf{b}$ against $H_0 : \mathbf{p} = \hat{\mathbf{p}}$. The LR test statistic is $-2\ln\left[L(\mathbf{b}|\mathbf{n})/L(\hat{\mathbf{p}}|\mathbf{n})\right]$, where $L(\cdot)$ is the likelihood function. Consequently, in the case of balanced data, the ratio $\frac{L(\mathbf{b}|\mathbf{n})}{L(\hat{\mathbf{p}}|\mathbf{n})}$ equals 1, resulting in a test statistic value of 0. However, when dealing with imbalanced data, where $L(\mathbf{b}|\mathbf{n}) < L(\hat{\mathbf{p}}|\mathbf{n})$, the test statistic takes on a value greater than 0. As the difference between the estimated class distribution $\hat{\mathbf{p}}$ and the balanced class distribution $\mathbf{b}$ increases, so does the value of the test statistic. Hence, the test statistic's value can serve as a measure of the difference between $\hat{\mathbf{p}}$ and $\mathbf{b}$, representing the extent of class imbalance. When $\hat{p}_c = \frac{n_c}{N}$, LRID can be written as:

$$LRID = -2\sum_{c=1}^{C} n_c \ln\frac{b_c}{\hat{p}_c} = -2\sum_{c=1}^{C} n_c \ln\frac{N}{Cn_c} \tag{2}$$

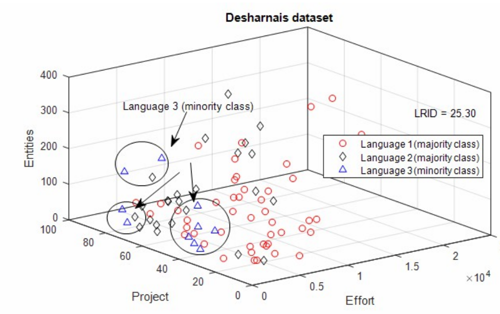
Note that comprehensive methodological overviews of LRID and the limitations of IR and ID can be found in the work of Rui Zhu [44]. The nature of imbalanced data exhibits complex characteristics, including small disjuncts, class overlap, rare instances, and outliers within the minority class space [55]. An example of the nature of imbalanced data is illustrated in Figure. 1.



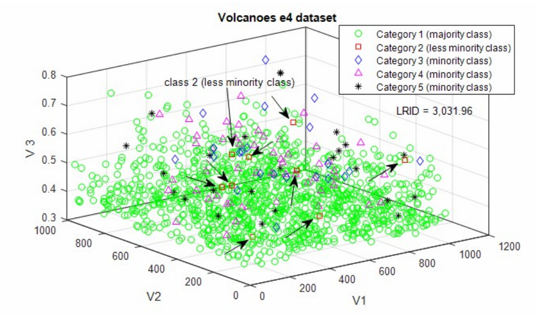
(a)



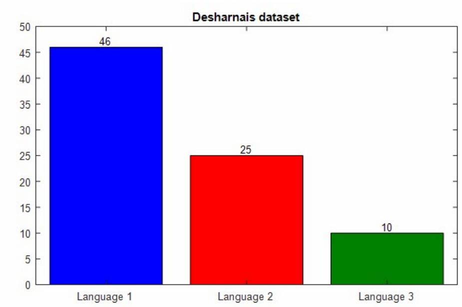
(b)



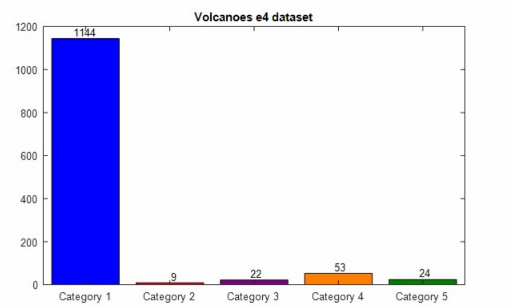
(c)



(d)



(e)



(f)

Figure 1. Illustrates an example of the nature of imbalanced datasets

As shown in Figure. 1, selecting two datasets, Desharnais and Volcanoes_e4, represents different degrees of class imbalance. Desharnais, with three classes, exhibits a lower LRID value, while Volcanoes_e4, with five classes, demonstrates a higher LRID value to facilitate a clearer understanding of the nature of imbalanced datasets. Figure. 1(a) depicts the 2D scatter plot of the Desharnais dataset with two attributes: project and effort. Figure. 1(c) shows a 3D plot of the Desharnais dataset with three attributes: project, effort, and entities—the attribute language used as the class label. Figure. 1(e) illustrates the bar chart of the Desharnais dataset. Figure. 1(b) displays the 2D scatter plot of the Volcanoes_e4 dataset with two attributes: V1 and V2. Figure. 1(d) presents a 3D plot of the Volcanoes_e4 dataset with three attributes: V1, V2, and V3. Figure. 1(f) illustrates the bar chart of the Volcanoes_e4 dataset. It can be observed that a high LRID value can hinder the model's learning efficiency.

2.2. The Related Methods for Imbalanced Data Issue

Recently, numerous works have been proposed to address the issue of imbalanced data [14]. In this section, some oversampling methods are briefly summarized below.

2.2.1. Synthetic Minority Over-sampling Technique The Synthetic Minority Over-sampling Technique (SMOTE) was introduced by Nitesh V. Chawla [29] as a method to handle imbalanced data in classification tasks. It increases the number of samples in the minority class by creating synthetic samples, resulting in a more balanced data distribution.

The SMOTE method randomly selects a sample from the minority class and finds its K-nearest neighbors based on a distance metric like Euclidean distance. It then generates synthetic samples along the line segment connecting the selected sample and its nearest neighbor. The synthetic samples are created by interpolating the features of the selected sample and its nearest neighbor. This process increases the number of samples in the minority class and balances the data distribution.

SMOTE effectively addresses data imbalance by increasing samples in the minority class, improving classification model performance. Widely used in machine learning (ML), it enhances accuracy and robustness, especially for skewed class distributions.

2.2.2. Minority Cloning Technique The Minority Cloning Technique (MCT) was introduced by Liangxiao Jiang [35]. It is an oversampling method designed to address imbalanced data by duplicating and generating synthetic samples for the minority class. The procedural steps of MCT are as follows:

1. Splitting the original data into the minority class and other classes.
2. Calculating each sample's importance value in the minority class using suitable techniques like proportional representation.
3. Copying minority samples based on the importance values to generate new synthetic samples for the minority class.
4. Generating additional synthetic samples in the minority class by considering the importance value calculation and appropriate methods, such as copying samples with higher importance values.
5. Mixing the generated synthetic samples with the original data to create a balanced dataset with oversampling for all classes.

MCT aims to address imbalanced data problems by increasing the number of samples in the minority class, leading to a more balanced dataset for better model performance.

2.2.3. Polynom-fit-SMOTE Polynom-fit-SMOTE, proposed by S. Gazzah [31], is an oversampling technique to generate synthetic samples for the minority class. The procedure can be summarized as follows:

1. Original Data Inspection: The first step involves examining the available data to identify the minority class in the original dataset that requires Polynom-fit-SMOTE for synthetic sample generation.
2. Important Region Detection: Polynom-fit-SMOTE identifies important regions in the original data that closely resemble the minority class. These regions are crucial for creating synthetic samples with a higher probability of occurrence.
3. Synthetic Sample Generation: Synthetic samples are generated by creating straight lines between samples in the minority class and other samples within the same class, based on the identified important regions. The intersection of these straight lines results in synthetic samples.
4. Analysis and Refinement of Synthetic Samples: The quality of the synthetic samples is analyzed to ensure their appropriateness and representation of the minority class. If needed, adjustments to parameter values or the decision to create new synthetic samples may be made.
5. Mixing Synthetic and Original Samples: Finally, the synthetic samples are mixed with the original data, ensuring a consistent distribution across all features. This creates a balanced dataset with oversampling for all classes.

2.2.4. Cluster-based Synthetic Oversampling Another noteworthy technique is the Cluster-based Synthetic Oversampling (CBSO), proposed by Sukarna Barua [21], which is an oversampling technique that leverages clustering to create synthetic samples for the minority class. The procedure can be summarized as follows:

1. Set the Number of Synthetic Samples: Determine the desired number of synthetic samples to be created for the minority class.
2. Cluster the Original Data: Employ clustering techniques such as K-means, DBSCAN, or Hierarchical Clustering to divide all data points in the minority class into subgroups.
3. Generate Synthetic Samples in Each Cluster: For each subgroup obtained in step 2, generate synthetic samples by randomly selecting points within the cluster and performing interpolation between these points.
4. Mix Synthetic and Original Samples: Combine the generated synthetic samples with the original data to create a dataset with oversampling that maintains balance across all classes.

CBSO builds upon the foundation of the advanced ADASYN algorithm, integrating established synthetic oversampling methodologies with a unique clustering-based data generation approach. Notably, CBSO diverges from ADASYN in generating synthetic data samples. In CBSO, this is achieved through an unsupervised clustering technique, unlike ADASYN's k-NN approach. Similarly, the adaptive synthetic sampling approach (ADASYN), proposed by He and Garcia [13], introduces a novel method for addressing class imbalance by dynamically generating synthetic data samples. ADASYN strategically generates synthetic instances based on their inherent learning complexities, effectively tackling challenges arising from imbalanced datasets. This technique mitigates learning bias from skewed class distributions and refines the decision boundary to prioritize instances with higher learning complexity.

3. Classifier Models

3.1. Decision Tree Model

A decision tree is a supervised nonparametric machine learning technique. The decision tree constitutes a tree-shaped data structure comprising nodes and edges arranged hierarchically. A decision tree formally represents mapping input attributes to their respective output classes. Its construction involves a "*divide and conquer*" approach, recursively dividing the training objects. It divides the provided dataset into smaller subsets as the tree's depth increases, using a splitting criterion to determine node selection and the corresponding attribute value. At each step, an attribute is chosen based on its discriminative power between different decision classes, becoming a test node and resulting in the partitioning of objects into subtrees based on potential outcomes. This process continues until it reaches a leaf node, representing a decision or class label for the instance. While decision trees are commonly used for binary classification (e.g., positive or negative instances), they can be extended to handle multi-class decision-making scenarios, allowing discrimination between various decision classes (e.g., low, medium, and high-risk patients).

3.1.1. Decision Tree Algorithm The Decision Tree algorithm is a versatile supervised learning method that effectively handles regression and classification tasks [56, 57, 58, 59]. It aims to create a predictive model using simple decision rules inferred from historical data. From the root, decision nodes compare attributes to guide the traversal down the tree, leading to the final classification at the leaf nodes. The Decision Tree classifier performs the classification through tree induction and tree pruning. The dataset is segmented into subsets using a splitting criterion. Various splitting criteria have been established in the literature [60], including Gini Index (GI), entropy-based methods, Information Gain (IG), and Bayesian networks. In this paper, our emphasis is on the GI.

As described by Jain et al., [60], the construction of the Decision Tree can be described as follows: Let S_j represent a subset of training points available at internal node j. S_j^L and S_j^R denote the left and right children for node j after splitting. It follows the following properties: (1) $S_j = S_j^L \cup S_j^R$, (2) $S_j^L \cap S_j^R = \emptyset$, (3) $S_j^L = S_{(2j+1)}$ and (4) $S_j^R = S_{(2j+2)}$. In the decision tree, nodes are divided into left and right subtrees based on splitting criteria such as the GI.

The key challenge in implementing Decision Trees is determining the attributes to be selected as the root node and at each level, known as attribute selection. Various attribute selection measures are employed to identify the most suitable attribute to be considered as the root node at each level.

3.1.2. Gini Index splitting criterion The Gini Index (GI) can be comprehended as a cost function utilized to assess dataset splits. It is calculated by subtracting the sum of the squared probabilities of each class from 1. The GI tends to favor larger partitions and is relatively straightforward to implement. On the other hand, information gain tends to favor smaller partitions with distinct values. A more detailed explanation can be found in Jain et al., [60].

The GI is in the range [0,1], where 0 indicates equality, and 1 indicates inequality. For a dataset D, the GI for attribute X is calculated as follows:

$$GI_X(D) = \sum_{j=1}^{k} p(X = x_j)\{1 - \sum_{i=1}^{v} p(Y \ = \ y_i/X = x_j)^2\}, \tag{3}$$

where v represents the number of classes, k represents all possible values of attribute X, Y represents the output value, and Y_i represents the i^{th} class value. The objective of the GI is to select the attribute value $(X = x_j)$ for splitting in a way that minimizes the overall impurity present in the dataset. This objective can be expressed as follows:

$$GI(X = x_j) = \arg\max_{\forall j}\{GI_X(D)\}. \tag{4}$$

The GI operates on a categorical target variable and is constrained to binary splits. A higher GI value indicates greater inequality and heterogeneity in the dataset.

3.2. XGBoost

The boosting method that relies on Decision Trees is known as the boosting tree. The XGBoost model represents an efficient implementation of the boosting tree model. XGBoost is a highly scalable end-to-end tree lifting. Essentially, the boosting tree model can be seen as an ensemble of Decision Trees. The interested reader can refer to Sheng and Yu [61] for a more comprehensive explanation. Given the initial lifting tree $\hat{y}_i^0 = f_0(x_i) = 0$. The solution model (predicted value) of the i^{th} sample at step t is:

$$\hat{y}_i^t = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{t-1} + f_t(x_i). \tag{5}$$

The objective function of XGBoost is represented as follows:

$$Obj^t = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^t\right) + \sum_{i=1}^{t} \Omega(f_i), \tag{6}$$

where l represents a differentiable convex loss function used to quantify the disparity between the prediction and the target. The term Ω denotes the complexity of Decision Trees to prevent the model from overfitting. Also,

$$\sum_{i=1}^{t} \Omega(f_i) = \sum_{i=1}^{t-1} \Omega(f_i) + \Omega(f_i), \tag{7}$$

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2, \tag{8}$$

where T represents the number of leaves, and w signifies the score of a leaf node. λ and γ are the penalty coefficients.

$$Obj^t = \sum_{i=1}^{n}[g_i f_i(x_i) + \frac{1}{2}h_i f_i^2(x_i)] + \Omega(f_i). \tag{9}$$

Let $g_i = l'(y_i, \hat{y}_i^{t-1}), \quad h_i = l''(y_i, \hat{y}_i^{t-1})$,$G_i = \sum_{i \in I_j} g_i$,$H_i = \sum_{i \in I_j} h_i$. The extreme value can be addressed using a unary two-times function.

$$w_j^* = -\frac{G_i}{H_i + \lambda}. \tag{10}$$

The objective function attains its minimum value.

$$\text{Obj} = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_i}{H_i + \lambda} + \gamma T. \tag{11}$$

3.3. *Random Forest*

In 2001, Breiman [62] introduced the Random Forest by integrating the bagging algorithm, the random subspace algorithm, and the classification and regression tree (CART). The Random Forest has found extensive application across various domains, yielding favorable outcomes in solving traditional classification and regression problems [63, 64]. Random Forest is an ensemble learning technique that generates multiple Decision Trees through iterative learning.

The mathematical expression for the Random Forest is expressed as follows [64]:

$$\{h(x, \beta_i), \quad i = 1, 2, 3, \ldots\}, \tag{12}$$

where, $h\left(x, \beta_i\right)$ represents the base classifier (e.g., CART) used to construct the Random Forest. x represents the dataset, while β is a vector set randomly selected from x using the Bagging algorithm.

These Decision Trees are then combined to form a forest, and their individual predictions are aggregated using majority voting to obtain the final prediction result. The fundamental unit of a Random Forest remains the Decision Tree. The complete algorithm involves two main phases: training and reasoning. During the training process, the algorithm constructs a Random Forest by creating several Decision Trees from the source data, and this forest is stored for later use in reasoning. During the reasoning process, the new data is evaluated using the trained Decision Trees, and the final prediction is determined based on the majority vote from all the trees, which is used to identify the category that received the highest number of votes.

The construction process of classical Random Forest is briefly presented as follows.

1. Initially, given a sample $x = \ x_1, x_2, \ldots, x_n$ with labels $l = l_1, l_2, \ldots, l_n$ of size N, this sample is drawn k times to generate k training sets. The Bagging algorithm is then employed to extract n samples from the original dataset x. The N samples are then trained to obtain a decision tree, which serves as the sample at the root node of the decision tree. Typically, n is less than N.
2. When each sample consists of M attributes, and for each node of the decision tree that needs to be split in each CART, m attributes are randomly selected from the pool of M attributes. Some splitting criteria are used to determine the feature, e.g., Gini coefficient or Information Gain. Typically, m is less than M. This results in each training set having a corresponding feature subset. Subsequently, we acquire multiple training sets and feature subsets, train multiple CARTs, and thus construct the Random Forest.
3. In the classification or regression process, multiple CARTs within the Random Forest simultaneously generate decisions, followed by a majority voting mechanism to consolidate these decisions and produce the final result.

Interested readers can refer to Sun et al., [64] for further information on the random forest.

4. Performance Measures and Datasets

This section elaborates the performance evaluation criteria and datasets in the following subsections.

4.1. Performance Measures

The performance evaluation involves using F-Measure and statistical methods, such as the Kruskal-Wallis test and Borda Count, to rank the data imbalance handling capabilities of the 66 methods. Below are the evaluation criteria and relevant details:

4.1.1. F-Measure The F-measure, also known as F1-score, represents the harmonic mean of Precision and Recall, effectively balancing both metrics simultaneously. While accuracy is commonly used for balanced datasets, it may not be suitable for imbalanced datasets, as it can result in high accuracy by misclassifying most minority class samples and classifying all samples as the majority class. In contrast, the F-measure is advantageous in cases where adjacent class prediction or handling imbalanced data is crucial. F-measure is valuable when there is a need to balance and consider both Precision (P) and Recall (R) simultaneously [65].

Precision, also known as positive predictive value, quantifies the proportion of true positive identifications out of all positive identifications made, where a higher value indicates better performance.

$$P = \frac{TP}{TP + FP}. \tag{13}$$

Recall, also known as true positive rate, measures the proportion of positive identifications correctly made out of all positive instances that should have been identified, where a higher value indicates better performance.

$$R = \frac{TP}{TP + FN}. \tag{14}$$

F-Measure is a widely used evaluation metric that combines precision and recall into a single value, typically with equal weighting on both measures.

$$F\text{-}Measure(x) = \frac{2PR}{P + R} \times 100, \tag{15}$$

where TP is true positive, FP is false positive, and FN is false negative.

4.1.2. Borda Count The Borda count, proposed by Jean-Charles de Borda [43], serves as a voting method. Voting methods rank and select alternatives, simplifying decision-making. The Borda count orders alternatives based on ranking sums, facilitating the process [66]. The Borda count belongs to a family of positional voting rules, wherein each candidate is assigned several points based on their rank in each ballot, reflecting the number of candidates ranked lower. In its original form, the lowest-ranked candidate receives 0 points, and the subsequent candidate receives 1 point, and so on, with the highest-ranked candidate obtaining $n - 1$ points, where n denotes the total number of candidates. Upon tallying all the votes, the option or candidate with the highest cumulative points emerges as the winner. The primary objective of the Borda count is to elect options or candidates that garner broad acceptability across the electorate rather than exclusively favoring the majority's preferences. Consequently, the Borda count is commonly acknowledged as a consensus-based voting system, as it strives to identify candidates with overall support from the voting population, thereby promoting inclusivity and representation in the electoral process.

$$B = \begin{array}{c} \\ A_1 \\ A_2 \\ \vdots \\ A_n \end{array} \begin{array}{c} \begin{array}{cccc} A_1 & A_2 & \cdots & A_n \end{array} \\ \begin{bmatrix} 0 & b_{12} & \cdots & b_{1n} \\ b_{21} & 0 & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & 0 \end{bmatrix} \end{array} \begin{array}{c} \text{Row sum} \\ S_1 \\ S_2 \\ \vdots \\ S_n \end{array} \tag{16}$$

Consider the Borda count matrix $B = [b_{ij}]_{n\times n}$ representing an election with a set of alternatives $A = \{A_i | i = 1, 2, \dots, n\}$. The matrix's rows and columns are labeled with the alternatives' names. The entry b_{ij} in the row labeled i and the column labeled j represents the result obtained by multiplying the "number of voters" by the "point value" when comparing alternative A_i with alternative A_j by the voters. The row sum provides the Borda scores $S = \{S_i | i = 1, 2, \dots, n\}$ for the alternatives. The Borda ranking is performed by ordering the Borda scores [66].

4.1.3. Kruskal-Wallis test This paper uses the Kruskal-Wallis [42] feature selection method. The Kruskal-Wallis test is a nonparametric alternative to one-way ANOVA, comparing medians of multiple data groups to determine if they come from the same population or populations with the same distribution. It utilizes ranks instead of numeric values for test statistics, assigning them by ordering the data across all groups. Tied observations receive the average rank. The test replaces the F-statistic used in classical ANOVA with a chi-square statistic, with the *P*-value indicating significance. Assumptions include samples having the same continuous distribution, except for location parameters due to group effects and independence of observations. It tests the null hypothesis of equal medians for two or more groups. The value of the Kruskal-Wallis test (G) can be calculated using the following formula:

$$G = \left[\frac{12}{N(N-1)} \sum_{j=1}^{k} \frac{R_j^2}{n_j} \right] - 3(N+1), \tag{17}$$

where N represents the total number of observations in all groups, n_j denotes the number of observations in group j, R_j signifies the rank of group j, and k represents the number of observations in a single group. The test statistic is normalized by multiplying it by a constant factor of $12/\left(N\left(N-1\right)\right)$ and subtracting another constant of $3(N+1)$ [67].

4.2. Datasets

The imbalanced datasets utilized in this study are segmented into two groups within the sub-experimental section. The initial group comprises 20 benchmark imbalanced datasets, chosen to compare the performance of each minority oversampling method.

To validate 66 of the methods for handling imbalanced datasets (minority oversampling methods), the evaluation datasets have been refined to a universally accepted set. We have chosen 20 widely recognized imbalanced datasets from the UCI repository, most of which are derived from multi-class datasets [68]. The datasets included in the study exhibit diverse characteristics, encompassing the real, nominal, categorical, and binary value attributes. Table 1 presents a comprehensive overview of the characteristic properties of the remaining 20 datasets, sorted by the Likelihood Ratio Imbalance Degree (LRID). The table includes details such as the number of data points (N), dimensions (D), target classes (T), and the LRID for each dataset.

The second group consisted of 30 imbalanced datasets, comprising 30 datasets sourced from the OpenML repository [69]. These datasets differ from those mentioned in the first group and Kovács's work [14]. To determine the optimal sampler, we selected these datasets to compare the performance of four top-performing oversampling methods. These methods include (a) two methods suggested by Kovács [14] which are the SMOTE-IPF, and ProWSyn, and (b) our top-performing oversampling practices observed in experimental results, namely MCT and CBSO. We present an overview of the critical characteristics of the 30 testing datasets in Table 2, sorted by the LRID.

5. Experimental Results

This paper has organized the experimental setup into two distinct subsections. The first subsection presents the outcomes of our comparison regarding the performance of various imbalanced dataset handling methods, using a total of 20 benchmark imbalanced datasets. In the second subsection, we compare the performance of the two top-ranking over-samplers suggested by Kovács [14], namely SMOTE-IPF and ProWSyn, against our findings of the highest-ranking over-samplers, specifically the MCT and CBSO methods we've identified as the most effective. Therefore, we focused on comparing the performance of these four highly effective oversampling methods. For this specific experiment, we employed a collection of 30 imbalanced datasets. The code for the competitive techniques addressing imbalanced datasets and classifier algorithms were sourced from Kovács [14].

The experimental setup's framework is presented in Figure. 2. All algorithms were implemented using Python and executed on a personal computer equipped with an AMD Ryzen 5 3.6 GHz CPU, 16 GB RAM, and the Windows 10 Professional 64-bit platform. The detailed information is outlined as follows.

Table 1. The 20 well-known imbalanced datasets from the UCI repository and their characteristics

No.	Datasets	N	D	T	LRID
1	Breast Tissue	106	10	6	2.99
2	Wine	178	14	3	4.48
3	Estimation of obesity levels based on eating habits and physical condition	2,111	17	7	14.00
4	Hayes-Roth	160	5	3	15.41
5	Maternal Health Risk	1,014	7	3	26.68
6	Higher Education Students Performance Evaluation	145	32	8	27.75
7	User Knowledge Modeling	403	6	4	42.92
8	Zoo	101	17	7	58.36
9	Forest Type Mapping	325	28	4	83.87
10	Contraceptive Method Choice	1,473	10	3	93.79
11	Ecoli	336	8	4	116.57
12	Glass Identification	214	10	6	121.17
13	Flags	194	194	8	122.01
14	Lymphography	148	19	4	158.46
15	Speaker Accent Recognition	329	13	6	200.25
16	Website Phishing	1,353	10	3	530.52
17	Student Performance on an Entrance examination	666	12	9	1,489.68
18	Car Evaluation	1,728	7	4	1,902.66
19	Dry Bean	13,611	17	7	3,033.31
20	Nursery	12,960	9	5	10,877.42

Note: N = Number of data, D = Number of Dimension, T = Number of Target, LRID = Likelihood Ratio Imbalance Degree

Table 2. The 30 imbalanced datasets and their characteristics

No.	Datasets	N	D	T	LRID
1	Waveform-5000	5,000	41	3	0.58
2	Vehicle-reproduced	846	19	4	1.09
3	AutoUniv-au7-700	700	13	3	2.47
4	Led24	3,200	25	10	5.47
5	Touch2	265	11	8	12.13
6	Grub-damage	155	9	4	16.32
7	Robot-failures-lp5	164	91	5	16.77
8	Diggle-table-a2	310	9	9	21.07
9	Desharnais	81	13	3	25.30
10	Hear-long-beach	200	14	5	40.86
11	Prnn-viruses	61	19	4	48.14
12	Heart-switzerland	123	13	5	59.00
13	Meta-all	71	63	6	78.53
14	AutoUniv-au6-750	750	41	8	110.57
15	Thyroid-new	215	6	3	119.17
16	AutoUniv-au7-500	500	13	5	120.44
17	Prnn-fglass	214	10	6	121.17
18	Heart-h	294	14	5	277.78
19	Solar Flare	1,066	13	6	329.56
20	Engine1	383	6	3	344.65
21	GesturePhaseSegmentationProcessed-seed-1-nrows-2000-nclasses-10-ncols*	2,000	33	5	356.94
22	JapaneseVowels	9,961	15	9	522.75
23	Artificial Characters	10,218	8	10	525.56
24	Steel-plates-fault	1,941	28	7	1,067.34
25	Yeast	1,484	9	10	1,710.63
26	cardiotocography	2,126	36	3	1,800.06
27	Volcanoes-e4	1,252	4	5	3,031.96
28	Wine-quality-white	4,898	12	7	6,420.96
29	Allrep	3,772	30	4	9,099.14
30	Page-blocks	5,473	11	5	12,795.53

Note: The full name of this item is 'GesturePhaseSegmentationProcessed-seed-1-nrows-2000-nclasses-10-ncols-100-stratify-True'.

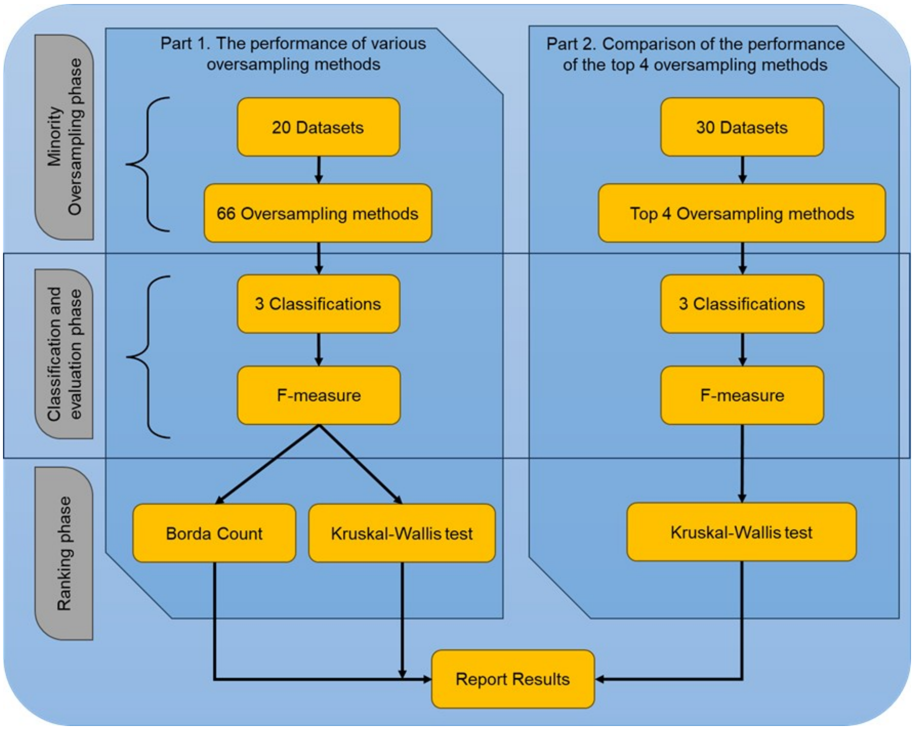


Figure 2. Illustrates the framework of the experimental setup.

5.1. Comparison Results with 20 Well-Known Imbalanced Datasets

In this section, we compare the performance of various imbalanced dataset handling methods using 20 well-known imbalanced datasets. In the experiments the trials were divided into two subsections. We conducted processing time comparisons for data imputation in the first subsection using various imbalance handling techniques. In the second subsection, we compared the classification performance of three classification models. The details are listed below.

5.1.1. Comparison of Execution Times In this section, 20 imbalanced datasets undergo data handling techniques to achieve a balanced or nearly balanced state between data groups, where one group contains notably fewer instances than the other. Due to page constraints, we employed four methods for data imputation: SMOTE, Polynom-fit-SMOTE, MCT, and CBSO. Subsequently, the time required for handling all 20 datasets is compared. The experiments comprise 100 independent runs. A comparative analysis of processing times between the remaining three techniques and the SMOTE method is executed with a confidence level of α = 0.05. The relative outcomes are detailed in Table 3.

Table 3 shows that the Polynom-fit-SMOTE, CBSO, MCT, and SMOTE methods exhibit varying data handling times across each dataset. Upon examining the Polynom-fit-SMOTE, MCT, and CBSO techniques, it is observed that they generally incur slower data handling times compared to SMOTE. Specifically, concerning the Student Performance on an Entrance examination and Website Phishing datasets, we found that Polynom-fit-SMOTE, MCT, and CBSO exhibit slower data management times than SMOTE. In the case of the Nursery, Student Performance on an Entrance examination, and Website Phishing datasets, MCT and CBSO demonstrate slower data handling times than SMOTE. For the Zoo dataset, both Polynom-fit-SMOTE and CBSO achieve faster data handling times compared to SMOTE. For most datasets, the data handling speeds for all four methods do not significantly differ statistically.

Therefore, based on the experimental findings, when considering an overall selection of data handling methods ranked by average time, it is evident that the Polynom-fit-SMOTE method exhibits the shortest processing time,

Table 3. Shows the comparison of the execution times for all three imbalanced data handling techniques compared to the SMOTE technique.

Datasets	Techniques			
	SMOTE	**Polynom-fit-SMOTE**	**MCT**	**CBSO**
Breast Tissue	0.0076	0.0074	0.0074	0.0073
Car Evaluation	0.0160	0.0142	0.0140	0.0141
Contraceptive Method Choice	0.0087	0.0086	0.0088	0.0085
Dry Bean	0.3050	0.3100	0.2950	0.3020
Ecoli	0.0066	0.0056	0.0054	0.0054
Estimation of obesity levels*	0.0830	0.0818	0.0737	**0.0690**†
Flags	0.0243	0.0243	0.0247	0.0282
Forest Type Mapping	0.0119	0.0114	0.0116	0.0115
Glass Identification	0.0129	0.0128	0.0130	0.0129
Hayes-Roth	**0.0032**	0.0041‡	0.0031	0.0031
Higher Education Students	0.0234	0.0237	0.0244	0.0259
Lymphography	0.0111	0.0112	0.0112	0.0115
Maternal Health Risk	**0.0059**	0.0058	0.0064‡	0.0061
Nursery	0.2190	0.2140	0.2320‡	0.2360‡
Speaker Accent Recognition	0.0093	0.0093	0.0092	0.0097
Student Performance on	**0.0288**	0.0338‡	0.0382‡	0.0358‡
User Knowledge Modeling	**0.0053**	0.0063‡	0.0054	0.0067‡
Website Phishing	0.0108	0.0125‡	0.0118‡	0.0118‡
Wine	0.0033	0.0032	0.0032	0.0032
Zoo	0.0574	0.0414‡	0.0533	**0.0404**†
Average	0.0427	**0.0421**	0.0426	0.0425

’†’ and ’‡’ denote statistically significant improvement or degradation over SMOTE, respectively, according to the t-test at a 0.05 significance level. The best result for each dataset is highlighted in bold. The full name of this item is ”Estimation of obesity levels based on eating habits and physical condition”

followed by CBSO, MCT, and then SMOTE. It can be observed that Polynom-fit-SMOTE, MCT, and CBSO demonstrate superior time efficiency in addressing imbalanced data compared to the SMOTE method.

Subsequently, the next section involves conducting a comparative experiment on imbalanced data handling techniques, followed by classification using classification models.

5.1.2. Ranking Score Comparison In this section, we conduct an experiment that compares various popular techniques for handling imbalanced data. A total of 66 methods are considered, which have been well-regarded from the past to the present. These methods are employed to address imbalanced data across 20 datasets. Once the data is balanced, it is classified using three classifier models: Decision Tree, Random Forest, and XGBoost. The performance comparison is based on the F-Measure metric, and the rankings are established using the Borda Count and Kruskal-Wallis test strategies. Both ranking approaches aim to assign higher cumulative points to superior methods, with the highest cumulative points indicating the winning technique.

In the experiments, each of the 66 oversampling methods and classifier models is executed across 30 independent runs. The summarized comparative performance outcomes can be found in Table 4.

From Table 4, to facilitate a better understanding of our interpretation, we have established vital points considering the reported results as follows:

1. The SMOTE score is established as a baseline for comparison with each over-sampler.
2. The SMOTE scores from the six columns across the three classifiers are examined, and the highest score is chosen as the baseline. This means that any scores lower than this baseline are not considered.
3. Consequently, only the top 10% of all the best scores relative to the baseline are considered.

Table 4. Ranking of the three algorithms using two strategies among 66 imbalanced handling techniques across 20 imbalanced datasets

Rank	Decision Tree				Random Forest				XGBoost			
	Borda Count	Score	Kruskal-Wallis Test	Score	Borda Count	Score	Kruskal-Wallis Test	Score	Borda Count	Score	Kruskal-Wallis Test	Score
1	**MCT**	**1,254**	**MCT**	**890.50**	**MCT**	**1,250**	**MCT**	**853.38**	**MCT**	**1,258**	**MCT**	**852.10**
2	**Polynom-fit-SMOTE**	**1,133**	**Polynom-fit-SMOTE**	**813.55**	**Polynom-fit-SMOTE**	**1,119**	**CBSO**	**808.80**	**Polynom-fit-SMOTE**	**1,147**	**Polynom-fit-SMOTE**	**801.05**
3	**CBSO**	**1,129**	**CBSO**	**801.00**	**CBSO**	**1,117**	**Polynom-fit-SMOTE**	**805.05**	**CBSO**	**1,115**	**CBSO**	**796.00**
4	G-SMOTE	1,025	G-SMOTE	769.20	SN-SMOTE	1,004	Supervised-SMOTE	781.50	Selected-SMOTE	1,037	G-SMOTE	766.25
5	A-SUWO	1,016	A-SUWO	761.25	G-SMOTE	1,002	G-SMOTE	777.05	G-SMOTE	1,014	Supervised-SMOTE	764.20
6	Cluster-SMOTE	972	NDO-sampling	756.70	SVM-balance	998	SVM-balance	776.73	SN-SMOTE	982	SN-SMOTE	758.85
7	TRIM-SMOTE	953	Supervised-SMOTE	751.15	Selected-SMOTE	977	SN-SMOTE	775.03	DSRBF	959	DSRBF	757.90
8	NDO-sampling	944	DSRBF	739.70	DSRBF	957	NDO-sampling	765.00	CE-SMOTE	948	SVM-balance	753.18
9	SDSMOTE	931	Distance-SMOTE	739.60	Supervised-SMOTE	933	DSRBF	763.80	Supervised-SMOTE	938	SDSMOTE	751.00
10	SN-SMOTE	900	SDSMOTE	738.10	SDSMOTE	931	SDSMOTE	760.75	A-SUWO	932	Selected-SMOTE	749.65
11	SVM-balance	883	Assembled-SMOTE	737.30	A-SUWO	929	Lee	760.75	SDSMOTE	927	Lee	749.63
12	DSRBF	865	SVM-balance	737.05	NDO-sampling	922	Assembled-SMOTE	757.10	Cluster-SMOTE	905	SMOTE-IPF	748.33
13	Selected-SMOTE	865	CE-SMOTE	736.60	CE-SMOTE	918	SMOTE-IPF	757.05	SMOTE-IPF	896	SMOTE	748.13
14	SMOTE-IPF	856	SN-SMOTE	736.25	Cluster-SMOTE	916	SMOTE	751.58	SVM-balance	895	Distance-SMOTE	746.15
15	Supervised-SMOTE	854	Cluster-SMOTE	730.85	Assembled-SMOTE	908	Selected-SMOTE	750.50	Lee	893	MSYN	743.85
16	AND-SMOTE	848	SMOTE-IPF	730.85	TRIM-SMOTE	893	Random-SMOTE	748.63	SMOTE	888	Edge-Det-SMOTE	743.30
17	CE-SMOTE	847	TRIM-SMOTE	730.60	Lee	887	MSYN	748.28	Assembled-SMOTE	888	SMOTE-OUT	742.75
18	Lee	838	Edge-Det-SMOTE	728.80	SMOTE-IPF	875	Edge-Det-SMOTE	747.90	TRIM-SMOTE	868	NDO-sampling	741.50
19	SMOTE	816	SMOTE	728.05	SMOTE	857	SMOTE-TomekLinks	746.30	MSYN	849	Assembled-SMOTE	741.43
20	Distance-SMOTE	816	MSYN	727.13	ADASYN	822	CE-SMOTE	744.40	NT-SMOTE	843	Random-SMOTE	740.53
21	Assembled-SMOTE	813	Selected-SMOTE	726.65	NT-SMOTE	818	Distance-SMOTE	744.30	AND-SMOTE	836	CE-SMOTE	737.15
22	MSYN	803	Lee	726.55	AND-SMOTE	813	ADASYN	741.18	Edge-Det-SMOTE	834	SMOTE-TomekLinks	733.90
23	Edge-Det-SMOTE	802	SMOTE-OUT	722.60	SMOTE-TomekLinks	810	SMOTE-OUT	734.70	NDO-sampling	832	A-SUWO	733.75
24	LN-SMOTE	788	ADASYN	716.00	MSYN	801	A-SUWO	730.80	ADASYN	820	ADASYN	727.35
25	SMOTE-OUT	788	Random-SMOTE	714.90	SMOTE-D	789	NT-SMOTE	725.55	Distance-SMOTE	817	ADOMS	721.15
26	SOI-CJ	786	DSMOTE	711.65	Borderline-SMOTE1	788	Cluster-SMOTE	721.18	SMOTE-TomekLinks	814	NT-SMOTE	719.25
27	SMOTE-D	782	SMOTE-D	711.20	Edge-Det-SMOTE	770	TRIM-SMOTE	721.00	SMOTE-OUT	813	Cluster-SMOTE	711.50
28	NT-SMOTE	774	NT-SMOTE	709.15	SMOTE-Cosine	756	ADOMS	718.70	SMOTE-D	788	ProWSyn	708.75
29	ADASYN	766	Borderline-SMOTE1	705.80	Random-SMOTE	749	SMOTE-Cosine	714.90	Random-SMOTE	774	TRIM-SMOTE	707.63
30	DBSMOTE	763	SMOTE-TomekLinks	703.05	Distance-SMOTE	745	ProWSyn	712.70	LN-SMOTE	752	SMOTE-D	706.40
31	SMOTE-TomekLinks	760	SMOTE-Cosine	702.35	SOI-CJ	733	SMOTE-D	706.05	Borderline-SMOTE1	748	SMOTE-Cosine	702.18
32	DSMOTE	722	ADOMS	697.65	DBSMOTE	724	Borderline-SMOTE1	702.05	SOI-CJ	748	DSMOTE	688.85
33	Random-SMOTE	720	AND-SMOTE	695.20	LN-SMOTE	715	SMOTE-ENN	694.90	SMOTE-Cosine	710	SMOTE-ENN	687.08
34	SMOTE-Cosine	712	SOI-CJ	692.85	SMOTE-OUT	695	DSMOTE	686.85	DBSMOTE	691	Borderline-SMOTE1	685.20
35	MWMOTE	712	MWMOTE	690.35	ADOMS	687	MWMOTE	685.83	MWMOTE	691	MWMOTE	683.48
36	OUPS	710	LN-SMOTE	687.70	Borderline-SMOTE2	676	AND-SMOTE	683.50	ADOMS	676	AND-SMOTE	674.15
37	Borderline-SMOTE1	709	MSMOTE	678.80	ANS	661	Borderline-SMOTE2	675.25	DSMOTE	647	LN-SMOTE	670.43
38	CURE-SMOTE	698	Borderline-SMOTE2	670.95	MWMOTE	656	SOI-CJ	666.65	OUPS	624	MSMOTE	669.75
39	MSMOTE	685	CURE-SMOTE	666.75	DSMOTE	634	LN-SMOTE	666.10	CURE-SMOTE	615	SOI-CJ	662.40
40	ANS	660	ProWSyn	666.25	ProWSyn	621	MSMOTE	659.35	LLE-SMOTE	614	OUPS	660.95
41	ADOMS	656	OUPS	665.10	CURE-SMOTE	610	RWO-sampling	651.45	ANS	603	RWO-sampling	651.45
42	LLE-SMOTE	648	SMOTE-ENN	663.90	LLE-SMOTE	607	LVQ-SMOTE	645.65	LVQ-SMOTE	588	LVQ-SMOTE	651.35
43	Borderline-SMOTE2	644	DBSMOTE	655.65	OUPS	601	SMOTE-RSB	640.48	ProWSyn	587	Borderline-SMOTE2	650.15
44	NRAS	611	LVQ-SMOTE	644.85	MSMOTE	577	OUPS	634.30	MSMOTE	586	CURE-SMOTE	649.65
45	LVQ-SMOTE	601	SMOTE-RSB	641.90	NRAS	570	LLE-SMOTE	630.78	Borderline-SMOTE2	583	SMOTE-RSB	643.90
46	SMOTE-RSB	566	NRAS	641.55	LVQ-SMOTE	567	CURE-SMOTE	629.95	SMOTE-ENN	558	LLE-SMOTE	633.60
47	SMOTE-ENN	553	ANS	636.40	SMOTE-ENN	553	SSO	619.20	RWO-sampling	555	SSO	621.60
48	ProWSyn	553	LLE-SMOTE	635.20	ASMOBD	515	DBSMOTE	610.00	NRAS	533	PDFOS	620.95
49	RWO-sampling	490	Kmeans-SMOTE	612.90	RWO-sampling	494	PDFOS	605.60	SMOTE-RSB	488	NRAS	612.50
50	ASMOBD	487	RWO-sampling	612.00	SMOTE-RSB	487	NRAS	601.70	PDFOS	429	DBSMOTE	608.25
51	Kmeans-SMOTE	482	PDFOS	588.85	Kmeans-SMOTE	451	ROSE	600.85	Kmeans-SMOTE	427	ROSE	601.70
52	SMMO	390	SSO	586.60	SSO	448	ANS	589.40	CCR	424	ANS	601.65
53	KernelADASYN	375	ROSE	582.95	PDFOS	423	CCR	566.50	ASMOBD	423	Kmeans-SMOTE	577.73
54	SSO	374	ASMOBD	577.75	SMMO	412	Kmeans-SMOTE	559.05	ROSE	414	CCR	559.83
55	PDFOS	364	V-SYNTH	547.10	ROSE	396	ASMOBD	539.70	KernelADASYN	414	MDO	546.00
56	ROSE	362	SOMO	546.30	KernelADASYN	392	MDO	532.70	SSO	399	SOMO	545.75
57	SOMO	360	MDO	530.05	CCR	380	SOMO	517.13	SMMO	374	ASMOBD	536.23
58	V-SYNTH	357	SMMO	529.70	SOMO	311	SMMO	512.90	MDO	316	SMMO	521.90
59	CCR	334	CCR	525.10	V-SYNTH	302	V-SYNTH	492.40	SOMO	298	V-SYNTH	515.00
60	MDO	286	KernelADASYN	500.90	MDO	298	KernelADASYN	486.15	V-SYNTH	268	KernelADASYN	497.35
61	Gazzah	241	Gazzah	500.00	DE-oversampling	237	Gazzah	460.70	SL-graph-SMOTE	221	Gazzah	478.95
62	DE-oversampling	232	DE-oversampling	491.00	SL-graph-SMOTE	223	DE-oversampling	453.85	DE-oversampling	220	DE-oversampling	476.75
63	SL-graph-SMOTE	220	SMOBD	478.45	Gazzah	189	SMOBD	441.00	Gazzah	169	SMOBD	465.20
64	SMOBD	174	SL-graph-SMOTE	446.65	SMOBD	157	SL-graph-SMOTE	439.60	SMOBD	149	SL-graph-SMOTE	446.20
65	Gaussian-SMOTE	84	Safe-Level-SMOTE	385.13	Gaussian-SMOTE	116	Safe-Level-SMOTE	336.95	Gaussian-SMOTE	120	Safe-Level-SMOTE	367.25
66	Safe-Level-SMOTE	78	Gaussian-SMOTE	286.40	Safe-Level-SMOTE	51	Gaussian-SMOTE	323.95	Safe-Level-SMOTE	58	Gaussian-SMOTE	325.10

4. Ultimately, the oversampling methods that appear consistently in all six columns across the three classifiers are selected as the top-performing methods.

Table 4 reveals that when utilizing the Borda Count strategy (column 3) with the Decision Tree algorithm, the SMOTE percentage is noted at 65.07%. Remarkably, the following methods display higher percentages compared to SMOTE: MCT, Polynom-fit-SMOTE, CBSO, G-SMOTE, A-SUWO, Cluster-SMOTE, TRIM-SMOTE, NDO-sampling, SDSMOTE, SN-SMOTE, SVM-balance, DSRBF, Selected-SMOTE, SMOTE-IPF, Supervised-SMOTE, AND-SMOTE, CE-SMOTE, and Lee, all rank higher than SMOTE.

When utilizing the Kruskal-Wallis test strategy (column 5) with the Decision Tree algorithm, the SMOTE percentage is noted at 81.75%. Notably, the following methods display higher percentages compared to SMOTE: MCT, Polynom-fit-SMOTE, CBSO, G-SMOTE, A-SUWO, NDO-sampling, Supervised-SMOTE, DSRBF, Distance-SMOTE, SDSMOTE, Assembled-SMOTE, SVM-balance, CE-SMOTE, SN-SMOTE, cluster-SMOTE, SMOTE-IPF, TRIM-SMOTE, and Edge-Det-SMOTE, all rank higher than SMOTE.

When utilizing the Borda Count strategy (column 7) with the Random Forest algorithm, the SMOTE percentage is noted at 68.56%. Remarkably, the following methods display higher percentages compared to SMOTE: MCT, Polynom-fit-SMOTE, CBSO, SN-SMOTE, G-SMOTE, SVM-balance, Selected-SMOTE, DSRBF, Supervised-SMOTE, SDSMOTE, A-SUWO, NDO-sampling, CE-SMOTE, cluster-SMOTE, Assembled-SMOTE, TRIM-SMOTE, Lee, and SMOTE-IPF, all rank higher than SMOTE.

When utilizing the Kruskal-Wallis test strategy (column 9) with the Random Forest algorithm, the SMOTE percentage is noted at 88.07%. Notably, the following methods display higher percentages compared to SMOTE: MCT, CBSO, Polynom-fit-SMOTE, Supervised-SMOTE, G-SMOTE, SVM-balance, SN-SMOTE, NDO-sampling, DSRBF, SDSMOTE, Lee, Assembled-SMOTE, and SMOTE-IPF, all rank higher than SMOTE.

When utilizing the Borda Count strategy (column 11) with the XGBoost algorithm, the SMOTE percentage is noted at 70.58%. Remarkably, the following methods display higher percentages compared to SMOTE: MCT, Polynom-fit-SMOTE, CBSO, Selected-SMOTE, G-SMOTE, SN-SMOTE, DSRBF, CE-SMOTE, Supervised-SMOTE, A-SUWO, SDSMOTE, cluster-SMOTE, SMOTE-IPF, SVM-balance, and Lee, all rank higher than SMOTE.

When utilizing the Kruskal-Wallis test strategy (column 13) with the Random Forest algorithm, the SMOTE percentage is noted at 87.79%. Notably, the following methods display higher percentages compared to SMOTE: MCT, Polynom-fit-SMOTE, CBSO, G-SMOTE, Supervised-SMOTE, SN-SMOTE, DSRBF, SVM-balance, SDSMOTE, Selected-SMOTE, Lee, and SMOTE-IPF, all rank higher than SMOTE.

According to Table 4, specifically in column 9, when the Kruskal-Wallis test strategy is used with the Random Forest algorithm, the SMOTE percentage surpasses the SMOTE percentages in all six columns, reaching 88.07%. The SMOTE method with the highest score is utilized as the reference point for ranking and comparison.

When analyzing the Decision Tree algorithm with the Borda Count strategy (column 3), it becomes evident that methods exceeding 88.07% (reference point) demonstrate superior performance in comparison to all SMOTE variations (highlighted in dark and light gray). Specifically, these methods include MCT, Polynom-fit-SMOTE, CBSO, G-SMOTE, and A-SUWO.

When analyzing the Decision Tree algorithm with the Kruskal-Wallis test strategy (column 5), it becomes evident that methods exceeding 88.07% (reference point) demonstrate superior performance in comparison to all SMOTE variations (highlighted in dark gray). Specifically, these methods include MCT, Polynom-fit-SMOTE, and CBSO.

When analyzing the Random Forest algorithm with the Borda Count strategy (column 7), it becomes evident that methods exceeding 88.07% (reference point) demonstrate superior performance in comparison to all SMOTE variations (highlighted in dark gray). Specifically, these methods include MCT, Polynom-fit-SMOTE, and CBSO.

When analyzing the Random Forest algorithm with the Kruskal-Wallis test strategy (column 9), it becomes evident that methods exceeding 88.07% (reference point) demonstrate superior performance in comparison to all SMOTE variations (highlighted in both dark and light gray). Specifically, these methods include MCT, CBSO, Polynom-fit-SMOTE, Supervised-SMOTE, G-SMOTE, SVM-balance, SN-SMOTE, NDO-sampling, DSRBF, SDSMOTE, Lee, Assembled-SMOTE, and SMOTE-IPF.

When analyzing the XGBoost algorithm with the Borda Count strategy (column 11), it becomes evident that methods exceeding 88.07% (reference point) demonstrate superior performance in comparison to all SMOTE variations (highlighted in dark gray). Specifically, these methods include MCT, Polynom-fit-SMOTE, and CBSO.

When analyzing the XGBoost algorithm with the Kruskal-Wallis test strategy (column 11), it becomes evident that methods exceeding 88.07% (reference point) demonstrate superior performance in comparison to all SMOTE variations (highlighted in dark and light gray). Specifically, these methods include MCT, Polynom-fit-SMOTE, CBSO, G-SMOTE, Supervised-SMOTE, SN-SMOTE, DSRBF, SVM-balance, and SDSMOTE.

On the other hand, considering the scores in Table 4 for the top approximately 10% best-performing data handling methods that outperform SMOTE across all classification algorithms and ranking strategies (as indicated by dark gray highlighting in all columns), only three methods stand out: MCT, Polynom-fit-SMOTE, and CBSO.

Based on the outcomes of the comparative and performance evaluation study, it can be inferred that the MCT, Polynom-fit-SMOTE, and CBSO methods are viable strategies for proficiently addressing imbalanced data to improve classification model performance. This offers practical guidance to practitioners in selecting suitable techniques for data management.

5.2. An Evaluation of the Top Four State-of-the-Art Oversampling Methods

Recently, numerous works have been proposed to address the issue of imbalanced data [14]. We researched the top two oversampling techniques recommended by Kovács [14], specifically SMOTE-IPF and ProWSyn. Additionally, we introduced two highly effective oversampling methods, MCT and CBSO, which were identified based on our experimental findings. The comparison of the four highly performing-oversampling methods is exhaustively investigated in this section. In this section, to substantiate our recommendation of the optimal oversampling methods ranking based on Borda Count and the Kruskal-Wallis test proposed in subsection 5.1, we provide an additional comparison of the performance of the top 4 performing oversampling procedures across a set of 30 imbalanced datasets. This evaluation is carried out using three classifier models: Decision Tree, Random Forest, and XGBoost. The performance and ranking are assessed using the F-Measure and the Kruskal-Wallis test. Please note that the Polynom-fit-SMOTE oversampling method, shown in our experimental results and in Kovács [14] work, has been omitted from consideration. This is due to its consistently identical best-performing outcomes.

The comparison details are presented in Table 5. This table displays the results of the ranking comparison achieved through the Kruskal-Wallis test across the four chosen oversampling methods: MCT, CBSO, SMOTE-IPF, and ProWSyn.

Table 5. The results of comparing ranking outcomes using the Kruskal-Wallis test among four selected oversampling methods

Oversampling Methods	Decision Tree			Random Forest			XGBoost		
	F-Measure	Kruskal-Wallis test	Rank Result	F-Measure	Kruskal-Wallis test	Rank Result	F-Measure	Kruskal-Wallis test	Rank Result
CBSO	0.7969	61.0667	2	0.8595	61.9500	2	0.8549	62.0667	2
ProWSyn	0.7482	51.4333	4	0.8262	52.2500	4	0.8200	52.9333	4
MCT	**0.8485**	**71.6333**	**1**	**0.8815**	**68.6333**	**1**	**0.8782**	**68.1000**	**1**
SMOTE-IPF	0.7779	57.8667	3	0.8482	59.1667	3	0.8432	58.9000	3

Considering Table 5 across the columns of the Decision Tree, Random Forest, and XGBoost classifiers, it is revealed that the MCT technique achieves the top rank (highlighted in boldface) with scores of 71.6333, 68.6333, and 68.1000, respectively. The second rank belongs to the CBSO technique with scores of 61.0667, 61.9500, and 62.0667, respectively. Ranking third is the SMOTE-IPF technique, with scores of 57.8667, 59.1667, and 58.9000, respectively. Lastly, in fourth place, we have the ProWSyn technique with scores of 51.4333, 52.2500, and 52.9333, respectively.

The efficacy of the leading four oversampling techniques in terms of performance was assessed using the Kruskal-Wallis test across 30 imbalanced datasets and three classifiers. This test aims to comprehensively understand the overall performance of the top four oversampling methods: CBSO, ProWSyn, MCT, and SMOTE-IPF. The top four performance-based oversampling methods were compared, as shown in Figure. 3, and Table 6 presents the statistical comparisons. The Kruskal-Wallis classification shown in Figure. 3 is derived from the experimental results presented in Table 5. A lower mean rank indicates better performance that can be achieved from the oversampling methods. Figure. 3 and Tables 5-6 reveal that MCT outperforms its competitors as it boasts the lowest mean ranks. ProWSyn achieved the lowest mean ranks. CBSO comes closest to MCT, followed by SMOTE-IPF.

Furthermore, the mean ranks of the MCT and ProWSyn groups exhibit a significant difference, with a *P*-value of 0.0060 at a significance level of 0.05. This implies that the ranking results between these two groups are not identical. Therefore, we can conclude that when ranking competition results by using the Kruskal-Wallis test, the ranked positions of the top 4 performance oversampling methods have changed. In our study, MCT holds the top rank. To better understand the ranking outcomes, we examine the experimental results presented in Table 7 alongside the LRID characteristics. Table 7 shows the performance ranking results for the four over-samplers across 30 imbalanced datasets.

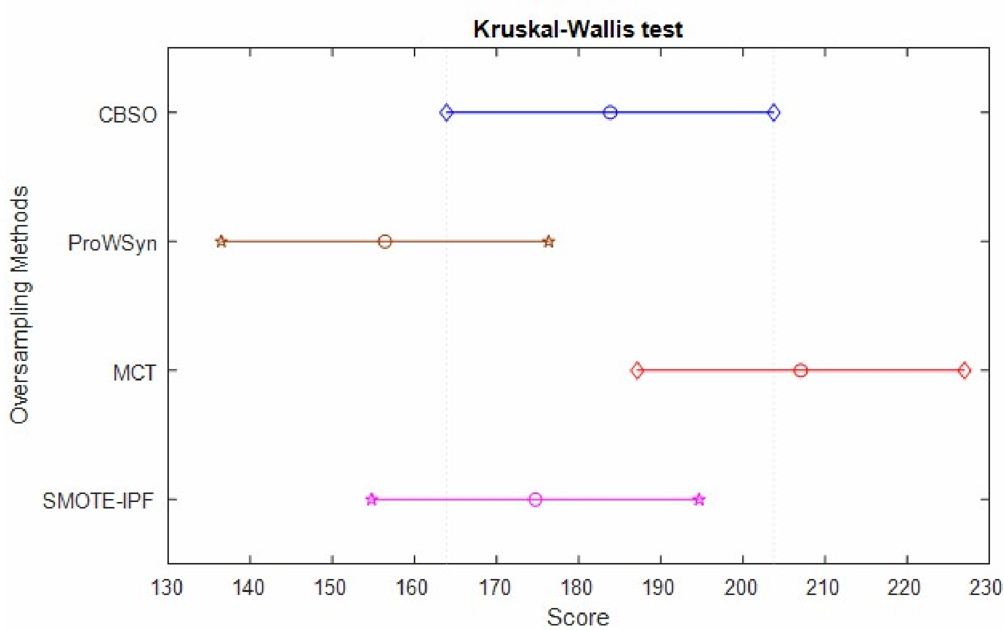


Figure 3. Illustrates Kruskal-Wallis test outcomes for the top four oversampling methods on 30 imbalanced datasets, employing three classifiers

Table 6. The statistical comparisons of the top 4 performance-based oversampling methods

Group A	Group B	Lower Limit	A-B	Upper Limit	*P*-value
CBSO	ProWSyn	-12.4323	27.4222	67.2767	0.2890
CBSO	MCT	-63.0712	-23.2167	16.6378	0.4395
CBSO	SMOTE-IPF	-30.7489	9.1056	48.9601	0.9361
ProWSyn	MCT	-90.4934	-50.6389	-10.7844	0.0060*
ProWSyn	SMOTE-IPF	-58.1712	-18.3167	21.5378	0.6391
MCT	SMOTE-IPF	-7.5323	32.3222	72.1767	0.1584

* Denote statistically significant differences between groups at a 0.05 significance level.

6. Discussion

In this section, we delve into an in-depth examination of the characteristics of imbalanced data and the behavior of different methods for handling such data. Due to space constraints, our study primarily investigates the most effective imbalanced data handling approaches—specifically MCT, Polynom-fit-SMOTE, and CBSO. Our objective is to thoroughly analyze and demonstrate the functionality of each method in contrast to the SMOTE technique, as elaborated below.

Table 7. the ranking of performance results for the four over-samplers across 30 imbalanced datasets

No.	Dataset	LRID	Decision Tree				Random Forest				XGBoost			
			CBSO	ProWSyn	MCT	SMOTE-IPF	CBSO	ProWSyn	MCT	SMOTE-IPF	CBSO	ProWSyn	MCT	SMOTE-IPF
1	Waveform-5000	0.58	2	3	**1**	4	4	2	**1**	3	3	4	**1**	2
2	Vehicle-reproduced	1.09	2	4	**1**	3	3	4	2	**1**	4	3	2	**1**
3	AutoUniv-au7-700	2.47	2	4	**1**	3	2	4	**1**	3	2	4	**1**	3
4	Led24	5.47	4	3	**1**	2	3	4	**1**	2	4	2	**1**	3
5	Touch2	12.13	2	4	**1**	3	**1**	4	2	3	**1**	4	2	3
6	Grub-damage	16.32	2	4	**1**	3	2	4	**1**	3	2	4	**1**	3
7	Robot-failures-lp5	16.77	2	3	**1**	4	**1**	4	3	2	2	4	**1**	3
8	Diggle-table-a2	21.07	**1**	2	4	3	**1**	2	4	3	4	3	**1**	2
9	Desharnais	25.30	2	4	**1**	3	2	3	**1**	4	2	3	**1**	4
10	Hear-long-beach	40.86	2	4	**1**	3	2	4	**1**	3	3	4	**1**	2
11	Prnn-viruses	48.14	**1**	3	2	4	**1**	2	3	4	**1**	3	2	4
12	Heart-switzerland	59.00	2	4	**1**	3	2	4	**1**	3	2	3	**1**	4
13	Meta-all	78.53	2	4	1	3	3	4	1	2	3	4	1	2
14	AutoUniv-au6-750	110.57	2	4	**1**	3	2	4	**1**	3	2	4	**1**	3
15	Thyroid-new	119.17	3	2	**1**	4	4	3	**1**	2	2	4	**1**	3
16	AutoUniv-au7-500	120.44	2	4	**1**	3	2	4	**1**	3	2	4	**1**	3
17	Prnn-fglass	121.17	2	4	**1**	3	2	4	**1**	3	2	4	**1**	3
18	Heart-h	277.78	3	4	**1**	2	3	4	**1**	2	3	4	**1**	2
19	Solar Flare	329.56	**1**	4	2	3	**1**	4	2	3	**1**	4	2	3
20	Engine1	344.65	4	3	**1**	2	3	4	**1**	2	3	4	**1**	2
21	GesturePhaseSegmentation*	356.94	2	4	**1**	3	3	4	**1**	2	3	4	**1**	2
22	JapaneseVowels	522.75	2	4	**1**	3	3	4	**1**	2	2	4	**1**	3
23	Artificial Characters	525.56	**1**	4	2	3	**1**	4	2	3	**1**	4	2	3
24	Steel-plates-fault	1,067.34	2	4	**1**	3	3	4	**1**	2	3	4	**1**	2
25	Yeast	1,710.63	2	4	**1**	3	3	4	**1**	2	2	4	**1**	3
26	cardiotocography	1,800.06	2	3	**1**	4	2	4	**1**	3	2	4	**1**	3
27	Volcanoes-e4	3,031.96	2	4	**1**	3	2	4	**1**	3	2	4	**1**	3
28	Wine-quality-white	6,420.96	2	4	**1**	3	2	4	**1**	3	2	4	**1**	3
29	Allrep	9,099.14	3	4	**1**	2	3	4	**1**	2	3	4	**1**	2
30	Page-blocks	12,795.53	4	3	**1**	2	4	3	**1**	2	3	4	2	**1**
	Overall Average Rank		2.17	3.63	**1.20**	3.00	2.33	3.70	**1.37**	2.60	2.37	3.77	**1.20**	2.67

6.1. Imbalanced Data Characteristics and Oversampling Methods

Due to the space limitations, we selected the Ecoli dataset to examine imbalanced data characteristics. The Ecoli dataset exhibits a Likelihood Ratio Imbalance Degree (LRID) of 116.57, indicating significant data imbalance. It comprises eight attributes and 336 instances, with multivariate dataset characteristics tailored for classification tasks. The attributes are of the real value type. Plotting the Ecoli dataset in a 3-dimensional graph provides insight into the data distribution and the presence of minority groups, as shown in Figure. 4.

From Figure. 4 (a), a 3D plot of the original Ecoli dataset is shown, featuring three attributes: feature 1 (Sequence Name: Accession number for the SWISS-PROT database), feature 5 (chg: the presence of charge on N-terminus of predicted lipoproteins–Binary attribute), and feature 7 (alm1: the score of the ALOM membrane-spanning region prediction program). The data distribution is visible. It can be observed from the plot that Group 2 and Group 3 have the fewest instances, with 52 and 25 data points, respectively, compared to the other groups.

6.2. Imbalanced Data Handling

Imbalanced data can lead to classification inaccuracies or biases towards the majority class. When applying the best-performing methods, namely MCT, Polynom-fit-SMOTE, and CBSO, to address the imbalanced data issue, the results are illustrated in Figures. 4 (b), (c), (d), and (e).

Figure. 4 (b) generated using the SMOTE method involves augmenting minority samples in Group 2 and Group 3 to achieve a size similar to the majority group. The distribution is adjusted around the original data, which, unfortunately, does not yield favorable classification outcomes.

Figure. 4 (c) generated using the CBSO method involves augmenting data in Group 2 and Group 3 to achieve a similar size as the majority group. The data distribution is around the original data, but with a distinct difference from SMOTE in that the data is not spread out to overlap with other groups. This results in more integrity compared to the SMOTE method.

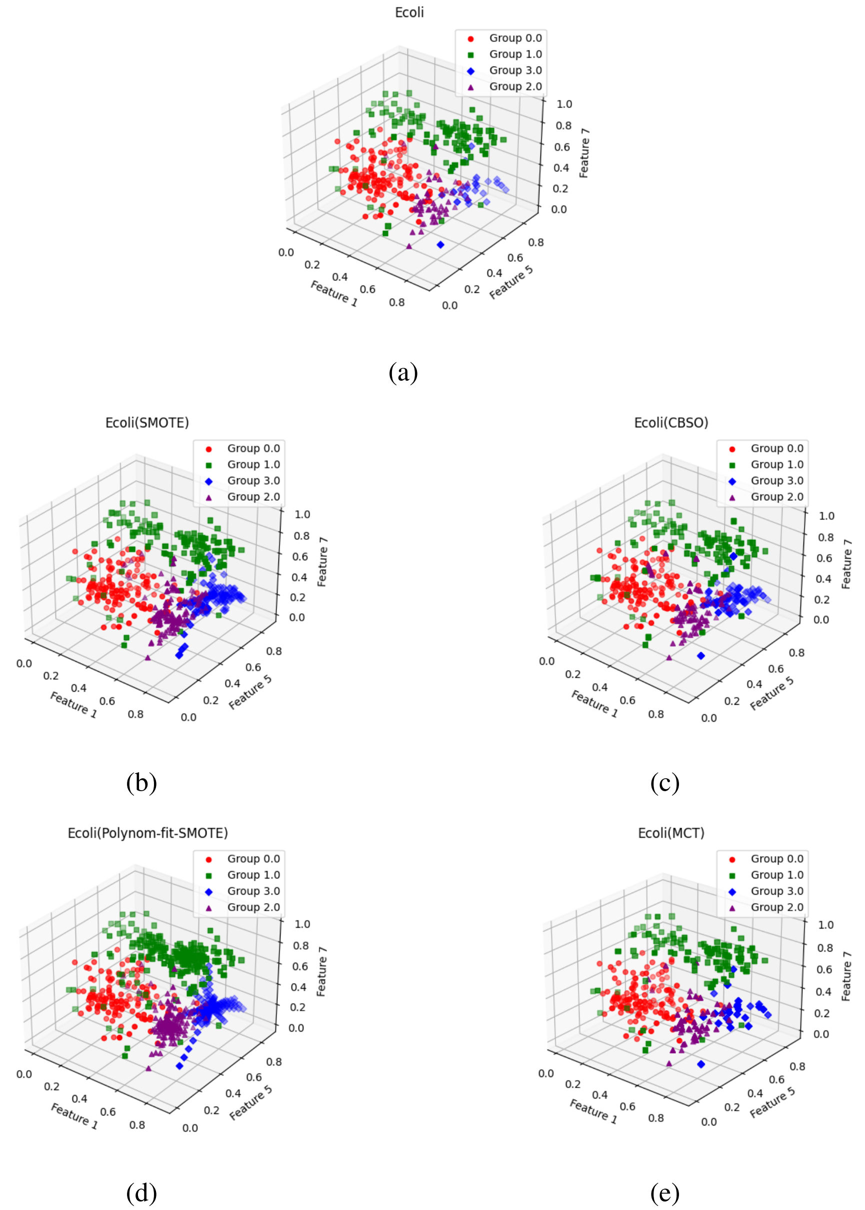


Figure 4. presents a 3-dimensional graph of the Ecoli dataset: (a) original Ecoli dataset, (b) SMOTE-generated, (c) CBSO-generated, (d) Polynom-fit-SMOTE-generated, and (e) MCT-generated.

Figure. 4 (d) generated using the Polynom-fit-SMOTE method involves augmenting data in Group 2 and Group 3 to achieve a similar size as the majority group. The added data are distributed like straight lines in multiple directions from the central point.

Lastly, Figure. 4 (e), generated through the MCT method, involves augmenting data in Group 2 and Group 3 to attain a proportion similar to that of the majority group. The added data is distributed around the original data, closely resembling the original data's distribution with minimal overlap with other group data. This characteristic contributes to the highest classification performance across all methods.

Through the comprehensive comparison and in-depth analysis of imbalanced data characteristics and the behaviors of the three selected methods - MCT, Polynom-fit-SMOTE, and CBSO - the analytical results distinctly illustrate the differences in their functionalities. Consequently, it becomes evident that the MCT method, with its evenly distributed and minimally overlapping augmented data, emerges as the most effective approach among all methods.

6.3. Examining the Impact of Different Ranking Methods

As the findings in subsection 5.1.2 reveal, the top-performing oversampling methods differ from the highly ranked oversampling techniques proposed by Kovács [14], specifically SMOTE-IPF and ProWSyn. Hence, conducting a more in-depth examination of the influence of various ranking methods is necessary. In this study, we have provided rankings based on Borda Count and the Kruskal-Wallis test. Based on the experimental results outlined in subsection 5.2, it is evident that the simple average ranking presented by Kovács [14] yields different results than our ranking based on Borda Count and the Kruskal-Wallis test. Hence, we can conclude that selecting an appropriate ranking technique is crucial when evaluating competition results.

7. Conclusion

In the current data science field, there has been a significant amount of research and development in addressing the issue of imbalanced data. These endeavors are based on modifying datasets to achieve equilibrium. Data's imbalanced nature can notably influence classification models' direct efficacy, improving data balance as a pivotal principle in this undertaking. Our objective is to study how a ranking technique influences the evaluation of traditional methods for handling imbalanced data. This research selected the best-performing imbalanced data handling methods in traditional contexts for study and performance comparison. Sixty-six methods were examined, utilizing three classification models: Decision Tree, Random Forest, and XGBoost. These techniques were assessed using a dataset compilation from the UCI and OpenML repositories, consisting of 20 and 30 datasets, respectively. The assessment included classification performance measured by F-Measure and statistical methods such as the Kruskal-Wallis test and Borda Count for ranking the effectiveness of the 66 imbalanced data handling techniques.

In the experimental section, we present a performance comparison of numerous methods for managing imbalanced datasets across a wide range of diverse imbalanced datasets. The empirical findings emphasize that MCT, Polynom-fit-SMOTE, and CBSO are the top three approaches. Additionally, the runtime for managing imbalanced data with these methods is comparable. Furthermore, in this research, we also conducted an in-depth analysis of the characteristics of imbalanced data and the behavior of the top three best-performing methods, namely MCT, Polynom-fit-SMOTE, and CBSO, in addressing data imbalance through data augmentation or imbalanced data handling. The study involved a comprehensive exploration and comparison of the imbalanced data's nature and the imbalanced data handling methods' behavior.

Finally, this research could benefit researchers, data analysts, and individuals dealing with imbalanced data. It can guide practitioners to apply these techniques in managing suitable datasets. One of the limitations of this study is that it does not encompass all the available methods within the classification models used. This study did not explore unsupervised learning models or other machine learning techniques. As future work, further studies should explore and apply these techniques to diverse datasets. The development of new methods for handling imbalanced data remains crucial and challenging.

Acknowledgement

The authors would like to thank Dr. György Kovács for his intellectual resources, which he has kindly shared with us throughout this project. We also wish to express our thanks to other relevant techniques, all of which contributed to the success of this research study.

Author Contributions

All authors contributed to the study's conception and design. **Tanawan Watthaisong** had the idea for the article, data collection, experimental work, result analysis, and contributed to the initial draft of the manuscript. **Khamron Sunat** conceptualized and defined the problem statement, designed the research plan, coordinated the study, participated in manuscript writing and finalization, conducted platform analysis and testing, performed result analysis, interpreted findings, and took on a supervisory role throughout the project. **Nipotepat Muangkote** participated in manuscript writing and finalization, contributed to presentation, and designed visualizations for figures. All authors read and approved the final manuscript.

Data availability

The datasets and third-party libraries used in the experiments are open source and accessible online through the UCI (http://archive.ics.uci.edu/ml/datasets.php) and OpenML (https://www.openml.org/search?type=data&sort=runs&status=any) repositories.

REFERENCES

1. G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, *Learning from class-imbalanced data: Review of methods and applications*, Expert Syst Appl, vol. 73, pp. 220–239, 2017, doi: 10.1016/j.eswa.2016.12.035.
2. M. Lv, Y. Ren, and Y. Chen, *Research on imbalanced data : based on SMOTE-AdaBoost algorithm*, in 2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE), 2019, pp. 1165–1170. doi: 10.1109/EITCE47263.2019.9094859.
3. M. Y. Arafat, S. Hoque, S. Xu, and D. M. Farid, *An Under-Sampling Method with Support Vectors in Multi-class Imbalanced Data Classification*, in 2019 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), 2019, pp. 1–6. doi: 10.1109/SKIMA47702.2019.8982391.
4. L. Yanze and H. OGAI, *Quasi-Linear SVM with Local Offsets for High-dimensional Imbalanced Data Classification*, in 2020 59th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), 2020, pp. 882–887. doi: 10.23919/SICE48898.2020.9240376.
5. B. Mirzaei, B. Nikpour, and H. Nezamabadi-Pour, *An under-sampling technique for imbalanced data classification based on DBSCAN algorithm*, in 2020 8th Iranian Joint Congress on Fuzzy and intelligent Systems (CFIS), 2020, pp. 21–26. doi: 10.1109/CFIS49607.2020.9238718.
6. J. Zheng, H. Qu, Z. Li, and L. Li, *Conformal transformation twin-hyperspheres for highly imbalanced data to binary classification*, in 2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA), 2022, pp. 1–6. doi: 10.1109/DSAA54385.2022.10032448.
7. A. Dixit and A. Mani, *Sampling technique for noisy and borderline examples problem in imbalanced classification*, Appl Soft Comput, vol. 142, p. 110361, 2023, doi: https://doi.org/10.1016/j.asoc.2023.110361.
8. S. Badrouchi, A. Ahmed, M. Mongi Bacha, E. Abderrahim, and T. Ben Abdallah, *A machine learning framework for predicting long-term graft survival after kidney transplantation*, Expert Syst Appl, vol. 182, p. 115235, 2021, doi: 10.1016/j.eswa.2021.115235.
9. P. Moghadam and A. Ahmadi, *A machine learning framework to predict kidney graft failure with class imbalance using Red Deer algorithm*, Expert Syst Appl, vol. 210, p. 118515, 2022, doi: 10.1016/j.eswa.2022.118515.
10. W. C. I. V Sleeman and B. Krawczyk, *Bagging Using Instance-Level Difficulty for Multi-Class Imbalanced Big Data Classification on Spark*, in 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 2484–2493. doi: 10.1109/BigData47090.2019.9006058.
11. R. Ji, M. Zhao, and Y. Wu, *Classification of Imbalanced Tumor Gene Expression Data Based on Weighted Gaussian Process*, in 2020 Chinese Automation Congress (CAC), 2020, pp. 5827–5831. doi: 10.1109/CAC51589.2020.9326958.
12. Q. Han, R. Yang, Z. Wan, S. Chen, M. Huang, and H. Wen, *Imbalanced data classification based on DB-SLSMOTE and random forest*, in 2020 Chinese Automation Congress (CAC), 2020, pp. 6271–6276. doi: 10.1109/CAC51589.2020.9326743.
13. H. He and E. A. Garcia, *Learning from Imbalanced Data*, IEEE Trans Knowl Data Eng, vol. 21, no. 9, pp. 1263–1284, 2009, doi: 10.1109/TKDE.2008.239.
14. G. Kovács, *An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets*, Appl Soft Comput, vol. 83, p. 105662, 2019, doi: 10.1016/j.asoc.2019.105662.
15. I. Dey and V. Pratap, *A Comparative Study of SMOTE, Borderline-SMOTE, and ADASYN Oversampling Techniques using Different Classifiers*, in 2023 3rd International Conference on Smart Data Intelligence (ICSMDI), 2023, pp. 294–302. doi: 10.1109/ICSMDI57622.2023.00060.
16. H. A. Gameng, B. B. Gerardo, and R. P. Medina, *Modified Adaptive Synthetic SMOTE to Improve Classification Performance in Imbalanced Datasets*, in 2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS), 2019, pp. 1–5. doi: 10.1109/ICETAS48360.2019.9117287.

17. F. Nhita, Adiwijaya, and I. Kurniawan, *Performance and Statistical Evaluation of Three Sampling Approaches in Handling Binary Imbalanced Data Sets*, in 2023 International Conference on Data Science and Its Applications (ICoDSA), 2023, pp. 420–425. doi: 10.1109/ICoDSA58501.2023.10276805.
18. R. Liu, *A novel synthetic minority oversampling technique based on relative and absolute densities for imbalanced classification*, Applied Intelligence, vol. 53, no. 1, pp. 786–803, 2023, doi: 10.1007/s10489-022-03512-5.
19. S. Feng, J. Keung, Y. Xiao, P. Zhang, X. Yu, and X. Cao, *Improving the undersampling technique by optimizing the termination condition for software defect prediction*, Expert Syst Appl, vol. 235, p. 121084, 2024, doi: https://doi.org/10.1016/j.eswa.2023.121084.
20. Abdullah-All-Tanvir, I. A. Khandokar, and S. Shatabda, *New boosting approaches for improving cluster-based undersampling in problems with imbalanced data*, Decision Analytics Journal, vol. 8, p. 100316, 2023, doi: https://doi.org/10.1016/j.dajour.2023.100316.
21. S. Barua, Md. M. Islam, and K. Murase, *A Novel Synthetic Minority Oversampling Technique for Imbalanced Data Set Learning*, in *Neural Information Processing*, B.-L. Lu, L. Zhang, and J. Kwok, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 735–744. doi: 10.1007/978-3-642-24958-7_85.
22. R. Zhang, S. Lu, B. Yan, P. Yu, and X. Tang, *A density-based oversampling approach for class imbalance and data overlap*, Comput Ind Eng, vol. 186, p. 109747, 2023, doi: https://doi.org/10.1016/j.cie.2023.109747.
23. X. Tao, X. Guo, Y. Zheng, X. Zhang, and Z. Chen, *Self-adaptive oversampling method based on the complexity of minority data in imbalanced datasets classification*, Knowl Based Syst, vol. 277, p. 110795, 2023, doi: https://doi.org/10.1016/j.knosys.2023.110795.
24. Z. Teng, P. Cao, M. Huang, Z. Gao, and X. Wang, *Multi-label borderline oversampling technique*, Pattern Recognit, vol. 145, p. 109953, 2024, doi: https://doi.org/10.1016/j.patcog.2023.109953.
25. C. Vairetti, J. L. Assadi, and S. Maldonado, *Efficient hybrid oversampling and intelligent undersampling for imbalanced big data classification*, Expert Syst Appl, vol. 246, p. 123149, 2024, doi: https://doi.org/10.1016/j.eswa.2024.123149.
26. T. Sandhan and J. Y. Choi, *Handling Imbalanced Datasets by Partially Guided Hybrid Sampling for Pattern Recognition*, in 2014 22nd International Conference on Pattern Recognition, 2014, pp. 1449–1453. doi: 10.1109/ICPR.2014.258.
27. D. Elreedy and A. F. Atiya, *A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance*, Inf Sci (N Y), vol. 505, pp. 32–64, 2019, doi: 10.1016/j.ins.2019.07.070.
28. B. S. Raghuwanshi and S. Shukla, *SMOTE based class-specific extreme learning machine for imbalanced learning*, Knowl Based Syst, vol. 187, p. 104814, 2020, doi: 10.1016/j.knosys.2019.06.022.
29. N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, *SMOTE: Synthetic Minority Over-sampling Technique*, Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.
30. D. A. Cieslak, N. V Chawla, and A. Striegel, *Combating imbalance in network intrusion datasets*, in 2006 IEEE International Conference on Granular Computing, 2006, pp. 732–737. doi: 10.1109/GRC.2006.1635905.
31. S. Gazzah and N. E. B. Amara, *New Oversampling Approaches Based on Polynomial Fitting for Imbalanced Data Sets*, in 2008 The Eighth IAPR International Workshop on Document Analysis Systems, 2008, pp. 677–684. doi: 10.1109/DAS.2008.74.
32. F. Koto, *SMOTE-Out, SMOTE-Cosine, and Selected-SMOTE: An enhancement strategy to handle imbalance in data level*, in 2014 International Conference on Advanced Computer Science and Information System, 2014, pp. 280–284. doi: 10.1109/ICACSIS.2014.7065849.
33. F. Fernández-Navarro, C. Hervás-Martínez, and P. Antonio Gutiérrez, *A dynamic over-sampling procedure based on sensitivity for multi-class problems*, Pattern Recognit, vol. 44, no. 8, pp. 1821–1833, 2011, doi: 10.1016/j.patcog.2011.02.019.
34. M. A. H. Farquad and I. Bose, *Preprocessing unbalanced data using support vector machine*, Decis Support Syst, vol. 53, no. 1, pp. 226–233, 2012, doi: 10.1016/j.dss.2012.01.016.
35. L. Jiang, C. Qiu, and C. Li, *A Novel Minority Cloning Technique for Cost-Sensitive Learning*, Intern J Pattern Recognit Artif Intell, vol. 29, no. 04, p. 1551004, Mar. 2015, doi: 10.1142/S0218001415510040.
36. Y. Han et al., *Production prediction modeling of food waste anaerobic digestion for resources saving based on SMOTE-LSTM*, Appl Energy, vol. 352, p. 122024, 2023, doi: https://doi.org/10.1016/j.apenergy.2023.122024.
37. D. Liu, S. Zhong, L. Lin, M. Zhao, X. Fu, and X. Liu, *Feature-level SMOTE: Augmenting fault samples in learnable feature space for imbalanced fault diagnosis of gas turbines*, Expert Syst Appl, vol. 238, p. 122023, 2024, doi: https://doi.org/10.1016/j.eswa.2023.122023.
38. C. Tao, T. Tao, S. He, X. Bai, and Y. Liu, *Wind turbine blade icing diagnosis using B-SMOTE-Bi-GRU and RFE combined with icing mechanism*, Renew Energy, vol. 221, p. 119741, 2024, doi: https://doi.org/10.1016/j.renene.2023.119741.
39. A. D. Amirruddin, F. M. Muharam, M. H. Ismail, N. P. Tan, and M. F. Ismail, *Synthetic Minority Over-sampling TEchnique (SMOTE) and Logistic Model Tree (LMT)-Adaptive Boosting algorithms for classifying imbalanced datasets of nutrient and chlorophyll sufficiency levels of oil palm (Elaeis guineensis) using spectroradiometers and unmanned aerial vehicles*, Comput Electron Agric, vol. 193, p. 106646, 2022, doi: 10.1016/j.compag.2021.106646.
40. R. A. Prasojo et al., *Precise transformer fault diagnosis via random forest model enhanced by synthetic minority over-sampling technique*, Electric Power Systems Research, vol. 220, p. 109361, 2023, doi: 10.1016/j.epsr.2023.109361.
41. A. Imakura, M. Kihira, Y. Okada, and T. Sakurai, *Another use of SMOTE for interpretable data collaboration analysis*, Expert Syst Appl, vol. 228, p. 120385, 2023, doi: 10.1016/j.eswa.2023.120385.
42. W. H. Kruskal and W. A. Wallis, *Use of Ranks in One-Criterion Variance Analysis*, J Am Stat Assoc, vol. 47, no. 260, pp. 583–621, 1952.
43. J. C. Borda, *Memoire sur les Elections au Scrutin*, Paris: Paris: Histoire de I' Academie Royale des Sciences, 1784.
44. R. Zhu, Z. Wang, Z. Ma, G. Wang, and J.-H. Xue, *LRID: A new metric of multi-class imbalance degree based on likelihood-ratio test*, Pattern Recognit Lett, vol. 116, pp. 36–42, 2018, doi: 10.1016/j.patrec.2018.09.012.
45. X. Liang, Y. Gao, and S. Xu, *ASE: Anomaly scoring based ensemble learning for highly imbalanced datasets*, Expert Syst Appl, vol. 238, p. 122049, 2024, doi: https://doi.org/10.1016/j.eswa.2023.122049.
46. M. Okkalioglu, *TF-IGM revisited: Imbalance text classification with relative imbalance ratio*, Expert Syst Appl, vol. 217, p. 119578, 2023, doi: https://doi.org/10.1016/j.eswa.2023.119578.
47. J. Ortigosa-Hernández, I. Inza, and J. A. Lozano, *Measuring the class-imbalance extent of multi-class problems*, Pattern Recognit Lett, vol. 98, pp. 32–38, 2017, doi: https://doi.org/10.1016/j.patrec.2017.08.002.

48. M. Bourel et al., *Machine learning methods for imbalanced data set for prediction of faecal contamination in beach waters*, Water Res, vol. 202, p. 117450, 2021, doi: https://doi.org/10.1016/j.watres.2021.117450.
49. S.-Y. Bae, J. Lee, J. Jeong, C. Lim, and J. Choi, *Effective data-balancing methods for class-imbalanced genotoxicity datasets using machine learning algorithms and molecular fingerprints*, Computational Toxicology, vol. 20, p. 100178, 2021, doi: https://doi.org/10.1016/j.comtox.2021.100178.
50. M. M. Chowdhury, R. S. Ayon, and M. S. Hossain, *An investigation of machine learning algorithms and data augmentation techniques for diabetes diagnosis using class imbalanced BRFSS dataset*, Healthcare Analytics, vol. 5, p. 100297, 2024, doi: https://doi.org/10.1016/j.health.2023.100297.
51. M. Mori, R. Gonzalez Flores, Y. Suzuki, K. Nukazawa, T. Hiraoka, and H. Nonaka, *Prediction of Microcystis Occurrences and Analysis Using Machine Learning in High-Dimension, Low-Sample-Size and Imbalanced Water Quality Data*, Harmful Algae, vol. 117, p. 102273, 2022, doi: https://doi.org/10.1016/j.hal.2022.102273.
52. J. L. García-Lapresta, M. Martínez-Panero, and L. C. Meneses, *Defining the Borda count in a linguistic decision making context*, Inf Sci (N Y), vol. 179, no. 14, pp. 2309–2316, 2009, doi: 10.1016/j.ins.2008.12.021.
53. P. S. Reel et al., *Machine learning for classification of hypertension subtypes using multi-omics: A multi-centre, retrospective, data-driven study Articles*, EBioMedicine, vol. 84, p. 104276, 2022, doi: 10.1016/j.ebiom.2022.104276.
54. W. Kanyongo and A. E. Ezugwu, *Machine learning approaches to medication adherence amongst NCD patients: A systematic literature review*, Inform Med Unlocked, vol. 38, p. 101210, 2023, doi: https://doi.org/10.1016/j.imu.2023.101210.
55. H. Guan, Y. Zhang, M. Xian, H. D. Cheng, and X. Tang, *SMOTE-WENN: Solving class imbalance and small sample problems by oversampling and distance scaling*, Applied Intelligence, vol. 51, no. 3, pp. 1394–1409, 2021, doi: 10.1007/s10489-020-01852-8.
56. C.-Y. Cheng and T.-P. Lin, *Decision tree analysis of thermal comfort in the courtyard of a senior residence in hot and humid climate*, Sustain Cities Soc, vol. 101, p. 105165, 2024, doi: https://doi.org/10.1016/j.scs.2023.105165.
57. I. Rahmatillah, E. Astuty, and I. D. Sudirman, *An Improved Decision Tree Model for Forecasting Consumer Decision in a Medium Groceries Store*, in 2023 IEEE 17th International Conference on Industrial and Information Systems (ICIIS), 2023, pp. 245–250. doi: 10.1109/ICIIS58898.2023.10253592.
58. C. J. D. Albia and D. Origines, *Selecting of Pet Adopters using C4.5 Decision Tree Model Algorithm*, in 2022 2nd International Conference in Information and Computing Research (iCORE), 2022, pp. 30–35. doi: 10.1109/iCORE58172.2022.00025.
59. R. Zhou, C. Li, and X. Wang, *Bank Customer Classification Algorithm Based on Improved Decision Tree*, in 2022 2nd Asia Conference on Information Engineering (ACIE), 2022, pp. 30–33. doi: 10.1109/ACIE55485.2022.00014.
60. V. Jain, A. Phophalia, and J. S. Bhatt, *Investigation of a Joint Splitting Criteria for Decision Tree Classifier Use of Information Gain and Gini Index*, in TENCON 2018 - 2018 IEEE Region 10 Conference, 2018, pp. 2187–2192. doi: 10.1109/TENCON.2018.8650485.
61. C. Sheng and H. Yu, *An optimized prediction algorithm based on XGBoost*, in 2022 International Conference on Networking and Network Applications (NaNA), 2022, pp. 1–6. doi: 10.1109/NaNA56854.2022.00082.
62. L. Breiman, *Random Forests*, Mach Learn, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
63. M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, *Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?*, Journal of Machine Learning Research, vol. 15, no. 90, pp. 3133–3181, 2014, [Online]. Available: http://jmlr.org/papers/v15/delgado14a.html
64. Z. Sun, G. Wang, P. Li, H. Wang, M. Zhang, and X. Liang, *An improved random forest based on the classification accuracy and correlation measurement of decision trees*, Expert Syst Appl, vol. 237, p. 121549, 2024, doi: 10.1016/j.eswa.2023.121549.
65. T. Wangchamhan, S. Chiewchanwattana, and K. Sunat, *Efficient algorithms based on the k-means and Chaotic League Championship Algorithm for numeric, categorical, and mixed-type data clustering*, Expert Syst Appl, vol. 90, pp. 146–167, 2017, doi: 10.1016/j.eswa.2017.08.004.
66. W.-W. Wu, *Beyond Travel & Tourism competitiveness ranking using DEA, GST, ANN and Borda count*, Expert Syst Appl, vol. 38, no. 10, pp. 12974–12982, 2011, doi: 10.1016/j.eswa.2011.04.096.
67. Z. Wang and Z. Xie, *Infrared face recognition based on local binary patterns and Kruskal-Wallis test*, in 2014 IEEE/ACIS 13th International Conference on Computer and Information Science (ICIS), 2014, pp. 185–188. doi: 10.1109/ICIS.2014.6912131.
68. M. Kelly, R. Longjohn, and K. Nottingham, *The UCI Machine Learning Repository*, Accessed: Sep. 24, 2023. [Online]. Available: https://archive.ics.uci.edu
69. J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo, *OpenML: Networked Science in Machine Learning*, SIGKDD Explorations, vol. 15, no. 2, pp. 49–60, 2013, doi: 10.1145/2641190.2641198.