Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Enhancing SMOTE for imbalanced data with abnormal minority instances

Surani Matharaarachchi [a,*], Mike Domaratzki [b], Saman Muthukumarana [a]

[a] *Department of Statistics, University of Manitoba, Winnipeg, MB, R3T 2N2, Canada*
[b] *Department of Computer Science, Western University, London, ON, N6A 5B7, Canada*

ARTICLE INFO

Dataset link: http://archive.ics.uci.edu/ml

Keywords:
Class imbalance
Abnormal instances
Inverse distance
SMOTE
Bayesian Gaussian mixture models
Dirichlet distribution

ABSTRACT

Imbalanced datasets are frequent in machine learning, where certain classes are markedly underrepresented compared to others. This imbalance often results in sub-optimal model performance, as classifiers tend to favour the majority class. A significant challenge arises when abnormal instances, such as outliers, exist within the minority class, diminishing the effectiveness of traditional re-sampling methods like the Synthetic Minority Over-sampling Technique (SMOTE). This manuscript addresses this critical issue by introducing four SMOTE extensions: Distance ExtSMOTE, Dirichlet ExtSMOTE, FCRP SMOTE, and BGMM SMOTE. These methods leverage a weighted average of neighbouring instances to enhance the quality of synthetic samples and mitigate the impact of outliers. Comprehensive experiments conducted on diverse simulated and real-world imbalanced datasets demonstrate that the proposed methods improve classification performance compared to the original SMOTE and its most competitive variants. Notably, we demonstrate that Dirichlet ExtSMOTE outperforms most other proposed and existing SMOTE variants in terms of achieving better F1 score, MCC, and PR-AUC. Our results underscore the effectiveness of these advanced SMOTE extensions in tackling class imbalance, particularly in the presence of abnormal instances, offering robust solutions for real-world applications.

1. Introduction

The issue of class imbalance poses a common challenge in machine learning, where instances are unevenly distributed across different classes. This issue is particularly prominent in real-world applications such as medical diagnosis (Matharaarachchi et al., 2021; Mazurowski et al., 2008), fraud detection (Yang et al., 2009), and churn prediction (Zhu et al., 2018), where minority classes are of particular interest. Imbalanced data introduces substantial obstacles for machine learning algorithms, leading to the development of biased models that perform poorly on the minority class.

In these imbalanced datasets, one or more classes contain significantly higher or lower instances than others. The class with more instances is typically called the majority class, while the one with fewer instances is termed the minority class.

This type of data can be categorized based on its imbalance ratio (IR) (Fernández et al., 2010), which is defined as the ratio of the number of examples in the majority class to those in the minority class. A higher imbalance ratio indicates a greater degree of imbalance between the classes. For example, if the class imbalance is $0.25 : 0.75$, then the IR is 3. In contrast, if the imbalance is $0.05 : 0.95$, the IR is 19.

This imbalance poses difficulties for traditional machine learning methods, making it challenging to learn from such datasets effectively. When abnormal instances are present in the minority class, the problem is exacerbated. Outliers and noisy data can distort the feature distribution, leading to the generation of unrepresentative synthetic samples. This, in turn, can result in models that are biased towards the majority class and unable to predict the minority class accurately.

Many solutions have been proposed to address class imbalance; however, a substantial research gap still exists in effectively handling abnormal instances within the minority class while employing re-balancing techniques. Most existing methods struggle to maintain prediction accuracy when generating synthetic data in the presence of these abnormal instances.

In this paper, we introduce four methods for handling dataset imbalance for datasets with abnormal instances: Distance ExtSMOTE, Dirichlet ExtSMOTE, FCRP SMOTE, and BGMM SMOTE. We conducted experiments supported by simulated and real-world data, with a particular focus on the presence of abnormal instances in the minority class, demonstrating the effectiveness of these proposed solutions in enhancing classification performance. Classification was performed using three classifiers: Logistic Regression (Weisberg, 2005), K-Nearest Neighbors

[*] Corresponding author.
E-mail addresses: matharas@myumanitoba.ca (S. Matharaarachchi), mdomarat@uwo.ca (M. Domaratzki), saman.muthukumarana@umanitoba.ca (S. Muthukumarana).

https://doi.org/10.1016/j.mlwa.2024.100597
Received 30 December 2023; Received in revised form 20 September 2024; Accepted 22 October 2024
Available online 29 October 2024

(KNN) (Cover & Hart, 1967), and Random Forest (Breiman, 2001). We evaluated model performance using various accuracy measures, including F1-Score, PR-AUC, and MCC. The four proposed methods provide alternative approaches that may be competitive with each other under different circumstances.

The primary objectives of this manuscript are as follows:

- To provide insights into the challenges of abnormal instances within the minority class when using SMOTE and some popular existing extensions.
- To propose several innovative strategies capable of mitigating the adverse effects of abnormal instances on class imbalance data.

The structure of this manuscript is organized as follows: Section 2 delves deeper into the limitations of SMOTE in the presence of abnormal instances, offering a comprehensive understanding of the issue. It also explores various strategies and techniques proposed to address this challenge. Section 3 presents empirical results from experiments conducted to validate the effectiveness of these strategies. Section 4 compares the results of each method using real-world application datasets. Finally, Section 5 of this paper is evaluated with a discussion of its contributions and limitations, whereas Section 6 provides the conclusion.

2. Literature review

To address the challenge of class imbalance, researchers have developed oversampling techniques. These methods create samples for the minority class or remove samples from the majority class, helping to balance the dataset by equalizing the number of data points across all classes. One notable approach in this domain is the Synthetic Minority Over-sampling Technique (SMOTE) (Chawla et al., 2002), which aims to address class distribution imbalance by generating artificial samples for the minority class. SMOTE has proven successful in enhancing classifier performance on imbalanced datasets by augmenting the representation of the minority class and expanding the training dataset.

Nonetheless, SMOTE is not without its limitations. It frequently fails to account for crucial factors, including the distribution of minority classes and hidden noise within the dataset, as noted by Hu et al. (2009). One notable drawback is SMOTE's inclination to overly generalize the minority class, resulting in misclassifications within the majority class and disturbing the overall equilibrium of the model, as emphasized by Blagus and Lusa (2013). While SMOTE has proven effective in various applications, its performance can also be compromised by abnormal instances. Abnormal instances can distort the synthetic sample generation process, leading to suboptimal results.

Researchers have proposed several extensions to overcome the limitations of the original SMOTE technique. Some of these methods focus on enhancing the generation of synthetic data by integrating SMOTE with other oversampling techniques. Notable examples include combining SMOTE with techniques like particle swarm optimization (Gao et al., 2011), kernel-based approaches (Mathew et al., 2015), and Boosting (Chawla et al., 2003). Bej et al. (2021) addressed the issue of over-generalizing the minority class by SMOTE by employing Localized Random Affine Shadowsampling (LoRAS).

To address the limitations of SMOTE in the presence of outliers, researchers have developed various extensions and adaptations. These variants aim to create synthetic samples that are robust to the influence of outliers, focusing on more accurate and reliable models.

Borderline-SMOTE (Han et al., 2005) is a variant that focuses on the borderline instances of the minority class. Generating synthetic samples near the boundary between classes helps mitigate the impact of outliers, providing better class separation.

ADASYN (Adaptive Synthetic Sampling) method (He et al., 2008) adapts the number of synthetic samples generated for each minority instance based on the local distribution and classification difficulty. Synthetic observations were created based on the difficulty level of learning specific instances within the minority class.

SVM-SMOTE (Suh et al., 2017) integrates SMOTE with the Support Vector Machine (SVM) algorithm to guide the synthetic sample generation. SVM identifies the support vectors that define the decision boundary between classes. Synthetic samples are generated along the support vectors, ensuring they are close to the decision boundary.

SMOTE-IPF (Sáez et al., 2015) incorporates an iterative ensemble-based noise filtering mechanism called Iterative-Partitioning Filter to handle noisy and borderline examples. After each iteration of synthetic sample generation, the filter evaluates the dataset and removes outliers. This iterative process continues until no further noise is detected, resulting in a synthetic dataset that is free from outliers and more representative of the minority class distribution.

Outlier-SMOTE, proposed by Turlapati and Prusty (2020), adopts a strategy of oversampling each data point based on its distance from other data points.

SMOTE-LOF (SMOTE combined with Local Outlier Factor) (Asniar et al., 2022) enhances the synthetic sample generation process by integrating LOF to identify and exclude outliers in the minority class. LOF calculates the local density deviation of data points to detect outliers. By removing these outliers before generating synthetic samples, SMOTE-LOF ensures that the synthetic data is more representative of the minority class's true distribution. This approach prevents the generation of synthetic samples around outliers, which could lead to poorly generalized synthetic samples and negatively impact the classifier's performance.

SMOTE-TomekLinks (Batista et al., 2004) integrates SMOTE with Tomek Links, a data cleaning technique that removes overlapping instances between classes. After generating synthetic samples using SMOTE, Tomek Links are applied to identify and remove instances near the decision boundary, likely to be noise or outliers. This combined approach enhances the dataset by cleaning outliers and reducing class overlap, leading to better classifier performance.

SMOTE-ENN (Edited Nearest Neighbors) (Batista et al., 2004) combines SMOTE with ENN, which removes instances misclassified by their nearest neighbors, effectively cleaning the dataset of noise and outliers. After generating synthetic samples with SMOTE, ENN filters out unreliable samples, ensuring that outliers in the minority class are not retained. This hybrid method improves the robustness of the synthetic dataset.

MWMOTE (Majority Weighted Minority Oversampling Technique) (Barua et al., 2014) identifies hard-to-learn minority instances, often outliers, and assigns them higher weights during the oversampling process. However, it avoids generating synthetic samples directly around these outliers, focusing instead on more informative minority instances. This reduces the influence of outliers on the synthetic dataset, leading to better classification performance.

DBSMOTE (Density-Based SMOTE) (Bunkhumpornpat et al., 2012) uses a density-based clustering approach to identify and handle outliers in the minority class. By applying a density-based clustering algorithm, DBSMOTE identifies sparse regions where outliers are likely to exist and generates synthetic samples within denser regions. This method ensures that outliers do not adversely affect the synthetic data generation process.

DSMOTE (Mahmoudi et al., 2014) is similar to DBSMOTE in using density information to guide synthetic sample generation. It focuses on generating synthetic samples in denser regions of the minority class, avoiding sparse areas where outliers are more likely to be found. This density-based approach helps create a more representative and outlier-free synthetic dataset.

ProWSyn (Proximity Weighted Synthetic Oversampling) (Barua et al., 2013) uses proximity information to guide the oversampling process. It assigns weights to minority instances based on their proximity to other instances, with outliers receiving lower weights. This weighted approach minimizes the impact of outliers on the synthetic

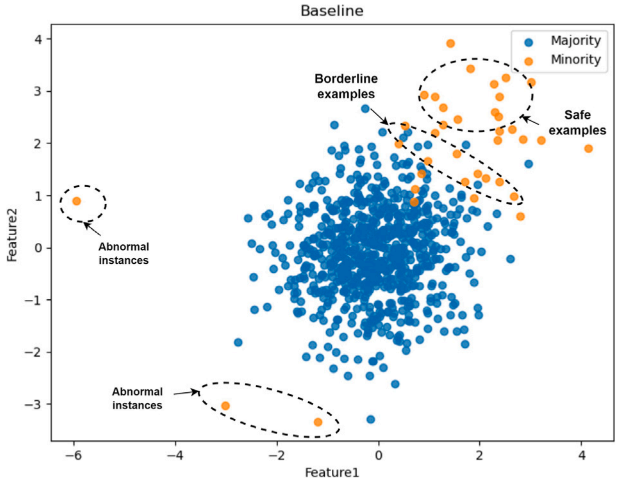


Fig. 1. Abnormal instances in the minority class.

dataset, ensuring the synthetic data remains representative of the minority class.

Cluster SMOTE (Cieslak et al., 2006) applies clustering algorithms to group minority instances before performing oversampling. Clusters containing outliers are either excluded or given less importance in the oversampling process. By generating synthetic samples within well-defined clusters and avoiding outlier-prone clusters, Cluster SMOTE ensures that the synthetic data accurately represents the minority class distribution.

Gaussian SMOTE (Lee et al., 2017) generates synthetic samples based on Gaussian distributions fitted to the minority class instances. Outliers, which do not conform to the main data distribution, have less influence on the Gaussian parameters, resulting in fewer synthetic samples being generated around them. This method effectively mitigates the impact of outliers on the synthetic dataset.

One key insight from analysing existing SMOTE variants is that many methods clean the data by removing outliers from the original dataset. While this may seem beneficial, it necessitates the addition of more synthetic instances to rebalance the dataset. This approach can overlook the valuable information that outliers might provide, potentially reducing the robustness and accuracy of the model's predictions.

By exploring the outcomes of some of the most common and competitive extensions (Kovács, 2019) alongside our approaches, we aim to gain valuable insights into improving the handling of imbalanced datasets in machine learning.

3. Methodology

3.1. Abnormal instances

In binary classification, abnormal instances refer to data points that deviate significantly from most of the dataset or are incorrectly labelled. These instances can be outliers, noise, or mislabelled data, causing challenges in building accurate predictive models (Fig. 1).

An outlier is "an observation or data point that significantly deviates from the majority of other observations in a dataset, often arousing suspicion that a different or abnormal process may have generated it" (Hawkins, 1980). Noise refers to data points deviating from general patterns, while mislabelled data are incorrectly categorized instances.

In this paper, we define the outlier ratio (OR) as the ratio of the number of outlier instances to the number of inlier instances. A higher outlier ratio indicates a greater prevalence of outliers relative to the inliers within the dataset. For example, an OR of approximately 0.0526 (≈ 0.05) corresponds to a 5% prevalence of outliers, and an OR of approximately 0.1111 (≈ 0.1) corresponds to a 10% prevalence of outliers.

Abnormal instances in a dataset may reflect legitimate extreme observations due to random fluctuations, representing inherent sampling characteristics. These should be retained and treated equally in analysis. However, these abnormal instances might not be useful for extensively generating new synthetic instances. For instance, abnormal instances may distort the dataset in medical diagnosis, misleading algorithms and biasing predictions. These instances, residing at the data's edges, challenge SMOTE's efficacy by generating unrepresentative synthetic samples. Therefore, effective strategies are crucial to applying SMOTE and ensuring reliable predictions.

3.2. Overview of SMOTE

SMOTE (Chawla et al., 2002) is a widely used and effective approach for tackling class imbalance in classification datasets. SMOTE addresses the issue of data scarcity in the minority class by generating synthetic samples that bridge the gap between minority class instances in feature space. The fundamental idea behind SMOTE is to create new synthetic instances by interpolating feature values between randomly selected minority class instances and one of their k-nearest neighbours.

The SMOTE algorithm (Algorithm 1) follows a simple and intuitive process. For each minority class instance, it selects k-nearest neighbours from the minority class instances. It then creates synthetic instances along the line segments connecting the selected instance with one of its k-nearest neighbours. These synthetic instances introduce new data points in the feature space and increase the representation of the minority class.

Algorithm 1 Original SMOTE

```
Require: $X \in \mathbb{R}^{n \times p}$ the features.
Require: $Y \in \{0,1\}^n$ the binary class label outputs.
Require: $k \in \mathbb{N}$ the number of neighbors to select for the $k$-Nearest Neighbors.
Ensure: Generated data $X_{new} \in \mathbb{R}^{q \times p}$ and $Y_{new} \in \{0,1\}^q$ with $q$ points created.
Denote by $S_1$ the number of points labelled as the minority class and $S_0$ the number of points labelled as the majority class.
Initialize $X_{new}$ and $Y_{new}$ as empty vectors.
while $S_1 < S_0$ do
    Filter $\mathcal{D} = X_i | Y_i = 1$, the set of points labelled as minority class 1.
    Randomly choose $r \in \mathcal{D}$ and find the indices of its $k$ nearest neighbors.
    Randomly choose an index $r_2$ among these neighbors.
    $x^{new} \leftarrow \alpha \times x_{r_1} + (1-\alpha) \times x_{r_2}$ with $\alpha \in [0,1]$ randomly chosen.
    $y^{new} \leftarrow 1$
    $S_1 = S_1 + 1$
    Append $x^{new}$ to $X_{new}$, append $y^{new}$ to $Y_{new}$
end while
return $X_{new}, Y_{new}$
```

The main goal of SMOTE is to balance classes by creating new similar samples for the minority class. However, when abnormal instances are part of this process, SMOTE might prioritize replicating these abnormalities instead of the typical minority class examples. This can lead to noisy synthetic samples that do not represent the true characteristics of the minority class well. Abnormal instances could make SMOTE generate new samples in data areas that do not truly represent the minority class, affecting the model's ability to generalize accurately and harming its performance. Fig. 2 shows the SMOTE-generated data based on the initial dataset displayed in Fig. 1. Within this visualization, there are three noticeable abnormal instances, and it is clear how these instances create an unusual data bridge between the minority data points and the abnormal ones.

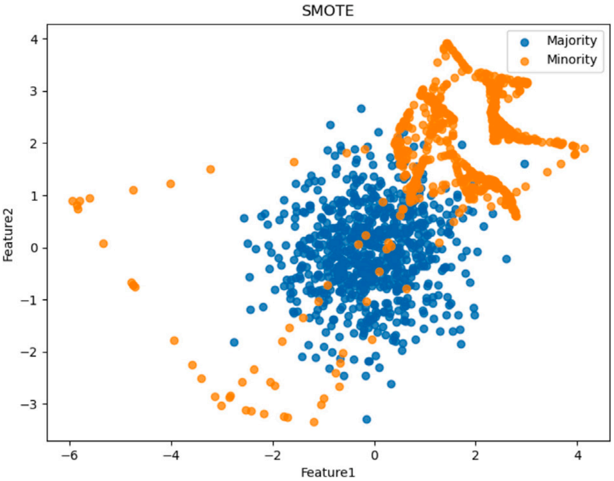


Fig. 2. Re-sampled data with SMOTE.

3.3. Proposed methods - based on weighted average

As a solution, we propose new over-sampling techniques called Distance ExtSMOTE, Dirichlet ExtSMOTE, FCRP SMOTE and BGMM SMOTE, Weighted average-based Minority Over-sampling TEchniques, which use weighted averages of nearest neighbours instead of a basic linear combination, allowing for learning from a more extensive set of nearest neighbours. Fig. 3 illustrates how our method modifies the synthetic data generation process compared to the standard SMOTE approach. In the following sections, we introduce various methods for determining appropriate weights, w_j to improve the model's ability to handle abnormal instances more effectively.

3.4. The distance measure

Distance is used to quantify the similarity between instances. Most of the extensions of SMOTE, which use distance metrics, consider the distance between the minority class instances to generate synthetic instances (Han et al., 2005; He et al., 2008; Mahmoudi et al., 2014). According to Feng et al. (2022), the selection of this distance does not impact the overall performances of existing SMOTE-based techniques, nor does the distance metric.

In contrast to these extensions, our approach involves leveraging the distances between the median centroid and the chosen nearest neighbours to determine the weights. The fundamental idea is to provide higher weights to instances closer in feature space, thus giving them more influence in the synthetic instance creation. This strategy proves particularly valuable in the presence of abnormal instances within the minority class, where minimizing their influence is essential. We opt for the median centroid, known for its robustness in the face of outliers. Even if the selected point is not abnormal, it is important to note that the generation of the synthetic data point will be distributed fairly around its neighbouring data points.

We initiate the process by determining the median centroid of the minority class, represented as μ, which is essentially the point that serves the "center" of these instances in a multidimensional space. It is calculated as the median of each feature (dimension) separately. After that, we identify the nearest neighbours for a random instance. These nearest neighbours, denoted as x_i, manifest as feature vectors that capture the attribute values of these proximate data points. Each x_i vector represents the feature values of the ith nearest neighbour. To calculate the distance between μ and each nearest neighbour x_i, we use the Euclidean distance (L2 norm), a measure of the straight-line distance between two points in Euclidean space (Eq. (1)).

$$d(\mu, x_i) = \sqrt{\sum_{j=1}^{n} (\mu_j - x_{ij})^2} \tag{1}$$

By taking the inverse of distances, instances closer to the median centroid receive higher weight, ensuring that the synthetic samples are more representative of the central tendency of the minority class and effectively minimize the effect of abnormal instances in the minority data. It enhances the robustness and accuracy of synthetic sample generation, contributing to improved classification performance in the presence of imbalanced datasets with abnormal instances.

3.5. Distance ExtSMOTE

In the first proposed method, Distance ExtSMOTE, the normalized inverse distance between the median centroid and the chosen k nearest neighbours is directly used as the weights (Algorithm 2).

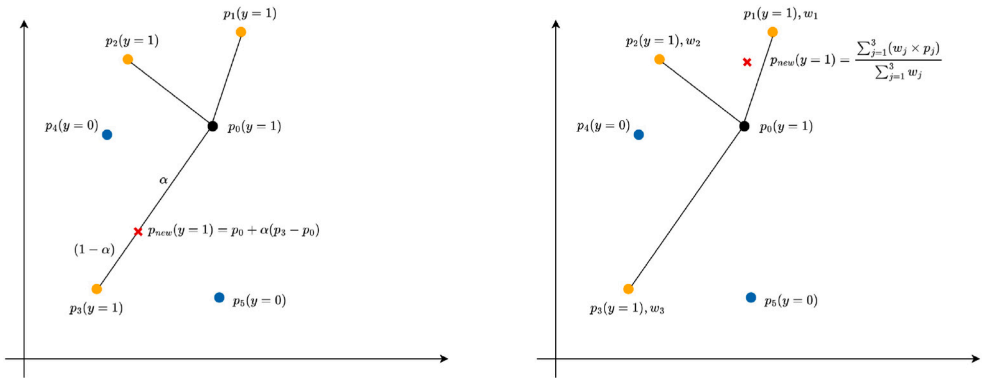
Fig. 4 demonstrates generating a new point using our suggested technique. In Fig. 4(b), a closer view of the data generation in Fig. 4(a) is presented. In a typical SMOTE scenario, the new point usually falls midway along one of the three lines. Yet, in our method, we position the new point nearer to the minority cluster, reducing the impact of outliers. Within Fig. 4(b), the initial blue value in parenthesis represents the distance from the median centroid. Meanwhile, the subsequent number exhibits the assigned weights for each neighbour, with those closer to the centroid receiving higher weights.

Algorithm 2 Distance ExtSMOTE

```
Require: $X \in \mathbb{R}^{n\times p}$ the features.
Require: $Y \in \{0,1\}^n$ the binary class label outputs.
Require: $k \in \mathbb{N}$ the number of neighbors to select for the $k$-Nearest Neighbors.
Ensure: Generated data $X_{new} \in \mathbb{R}^{q\times p}$ and $Y_{new} \in \{0,1\}^q$ with $q$ points created.
Denote by $S_1$ the number of points labelled as the minority class and $S_0$ the number of points labelled as the majority class.
Initialize $X_{new}$ and $Y_{new}$ as empty vectors.
Obtain the median centroid $(\mu)$ of the minority class.
while $S_1 < S_0$ do
    Filter $\mathcal{D} = X_i | Y_i = 1$, the set of points labelled as minority class 1.
    Randomly choose $r \in \mathcal{D}$ and find the indices of its $k$ nearest neighbors, $r_1, \dots, r_k$.
    Consider the inverse distances, from $\mu$, to each nearest neighbour as weights, $w_j = d_j^{-1}$
    $x^{new} \leftarrow \frac{\sum (w_j \times x_{r_j})}{\sum w_j}$ for all $j$ from 1 to $k$.
    $y^{new} \leftarrow 1$
    $S_1 = S_1 + 1$
    Append $x^{new}$ to $X_{new}$, append $y^{new}$ to $Y_{new}$
end while
return $X_{new}, Y_{new}$
```

3.6. Dirichlet ExtSMOTE

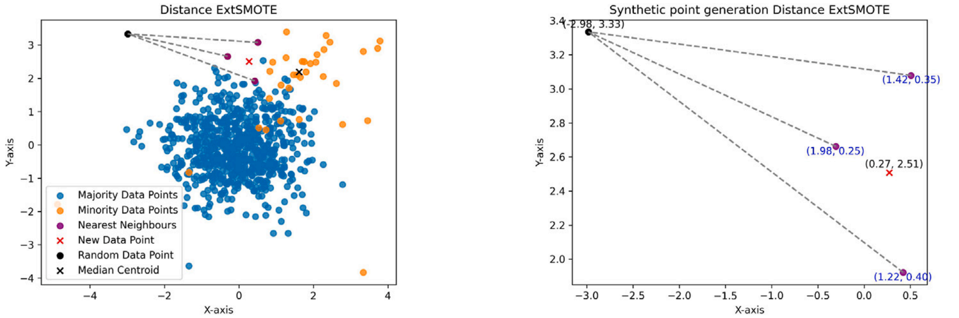
The Dirichlet ExtSMOTE method introduces a probabilistic framework to assign weights. It utilizes the Dirichlet distribution parameters obtained through three distinct approaches. These parameters are then employed as weights for each nearest neighbour in the form of generated Dirichlet samples.



(a) SMOTE data generation

(b) Proposed data generation

Fig. 3. Data generation mechanisms.



(a) This scenario occurs when an abnormal instance is chosen as a neighbouring point.

(b) The values within parentheses indicate (d_j, w_j).

Fig. 4. Distance ExtSMOTE data generation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3.6.1. Dirichlet distribution

The Dirichlet distribution is a multivariate probability distribution defined on the simplex, which allows us to capture the class proportions and guide the creation of more representative synthetic instances. The Dirichlet distribution is used in diverse applications across numerous domains. It plays a fundamental role in the modelling of compositional data, Bayesian analysis, statistical genetics, nonparametric inference, distribution-free tolerance intervals, multivariate analysis, order statistics, reliability, probability inequalities, probabilistic constrained programming models, limit laws, delivery problems, stochastic processes, and other areas (Ng, Tian, & Tang, 2011).

It also can be considered a distribution over probability distributions (Bela et al., 2010; Ng, Tian, & Tang, 2011). Each draw from a Dirichlet distribution yields a vector of probabilities that can be interpreted as a probability distribution, where each element represents the likelihood of an outcome within a specific category or dimension.

Mathematically, the Dirichlet distribution is defined by a set of parameters $\alpha_1, \alpha_2, \ldots, \alpha_K$, where K is the dimensionality of the probability simplex. The parameters are positive real numbers, and the distribution is typically represented as $Dir(\boldsymbol{\alpha})$. The Dirichlet distribution is parameterized by $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_K]$, which can be considered pseudo-counts or prior observations.

Let $\boldsymbol{p} = [p_1, p_2, \ldots, p_k]$ be a K-dimensional vector s.t $\forall j : p_j \geq 0, j = 1, 2, \ldots, k$ and $\sum_{j=1}^{K} p_j = 1$. If Γ is the gamma function, then the probability density function (pdf) of the Dirichlet distribution for a point $\boldsymbol{p}$ on the simplex is given by Bela et al. (2010):

$$P(\boldsymbol{p}|\boldsymbol{\alpha}) \sim Dir(\alpha_1, \alpha_2, \ldots, \alpha_K) \overset{\text{def}}{=} \frac{\Gamma(\sum_j \alpha_j)}{\prod_j \Gamma(\alpha_j)} \prod_{j=1}^{K} p_j^{\alpha_j - 1} \quad (2)$$

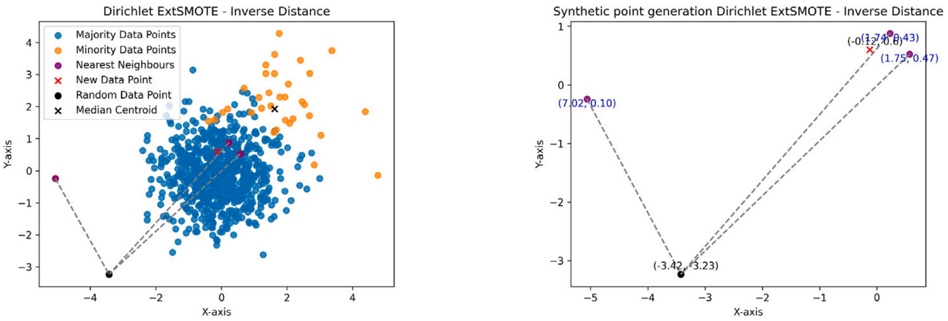
Three distinct approaches are utilized to generate Dirichlet parameters in the Dirichlet ExtSMOTE method. The first method involves generating random data from a $Uniform(0,1)$ distribution, known as the uniform distribution approach. The second method, termed the uniform vector approach, generates data from a unit vector of size k. Finally, the inverse distance approach determines parameters by considering the inverse distances between the median centroid and the chosen k nearest neighbors. To enhance variability with the concentration parameter, each vector is multiplied by a scalar m. These approaches collectively contribute to the diverse generation of Dirichlet parameters within the Dirichlet ExtSMOTE framework.

By incorporating Dirichlet weights, the algorithm promotes the creation of more diverse and relevant synthetic instances (Fig. 5). This results in synthetic samples that better reflect the characteristics of the minority class. The Dirichlet distribution's ability to generate diverse weights mitigates the risk of overfitting that traditional SMOTE might face.

The algorithm (Algorithm 3) enables better generalization of classifiers on imbalanced datasets, improving performance in real-world applications with class imbalance challenges. In the subsequent section, we present the experimental setup and evaluation results, highlighting the advantages of Dirichlet ExtSMOTE over standard SMOTE and other state-of-the-art approaches in handling class imbalance.

3.7. FCRP SMOTE

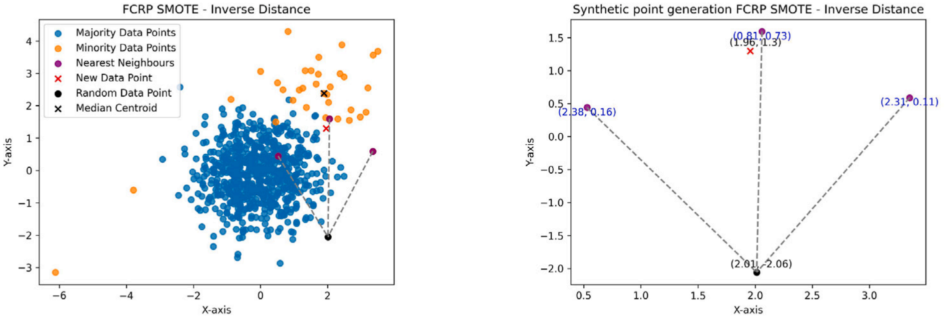
The Chinese Restaurant Process (CRP) (Pitman & Picard, 2006) stands as a foundational probabilistic concept frequently employed to display random allocation or grouping in the domains of Bayesian



(a) This scenario occurs when an abnormal instance is chosen as a neighbouring point.

(b) The values within parentheses indicate (d_j, w_j).

Fig. 5. Dirichlet ExtSMOTE (Inverse Distance).



(a) This scenario occurs when an abnormal instance is chosen as a neighbouring point.

(b) The values within parentheses indicate (d_j, w_j).

Fig. 6. FCRP SMOTE.

statistics and machine learning. FCRP SMOTE adopts the CRP concept but integrates a predetermined number of tables, denoted as k, and assigns initial preferences to each table.

Consider a scenario with k nearest neighbours represented as tables in a restaurant setting. In this framework, customers emulate the role of developing the weights to aid in selecting a synthetic data point. Initially, there exist specific preferences allocated to each table, which are essentially the normalized inverse distances. When a customer arrives, they select a table based on these probabilities. Subsequently, after this choice, the probability for that particular table is adjusted using an α value, and recalculations are made for all other probabilities in relation to that selection. This sequential process iterates for a total of N customers. The resulting probabilities obtained through these iterations serve as the desired weights for each nearest neighbour within the FCRP SMOTE methodology as shown in Algorithm 4. Fig. 6 illustrates the generation of a synthetic instance.

3.8. BGMM SMOTE

In BGMM SMOTE (A cluster-based Synthetic Minority Oversampling Technique), the approach relies on assigning each neighbouring data point to a specific cluster. Subsequently, the algorithm assigns weights to each nearest neighbour based on the likelihood associated with each cluster. This method uses the clustering information to guide the creation of synthetic instances in the minority class, considering the likelihood estimates derived from the clusters to generate new samples more effectively.

3.9. Bayesian Gaussian mixture models with EM algorithm

In cases where abnormal instances are present within the minority class, it is often observed that these outlier points tend to reside at a considerable distance from the primary cluster. This also suggests that these points may belong to a distinct cluster separate from the main cluster where the median centroid resides. However, such disparity in data distribution raises a fundamental challenge – determining the number of clusters that emerge from these scattered data points. In such scenarios, traditional clustering methods may fall short due to the absence of prior knowledge about the data's inherent structure.

To address this challenge effectively, we turn to a non-parametric Bayesian approach called the Bayesian Finite Mixture Model (Roberts et al., 1998). This approach allows us to overcome the need to specify the number of clusters beforehand. Instead, it dynamically adapts to the data, flexibly accommodating the varying degrees of cluster complexity within the chosen dataset.

The likelihood function, given n independent and identically distributed observations $\boldsymbol{y_i} \in \mathbb{R}^p$ and a fixed number of components K, is defined as:

$$\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{y}, K) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k f_k(\boldsymbol{y_i}|\boldsymbol{\theta_k}) \tag{3}$$

where $\boldsymbol{y}$ are the observations from the mixture model and $f_k()$ a parametric density with parameters θ_k and π_k is the cluster mixing weight.

Furthermore, estimating cluster mixing weights, which define each cluster's relative contribution to the overall data distribution, becomes a critical aspect of this task. In the absence of prior information,

Algorithm 3 Dirichlet ExtSMOTE

```
Require: $X \in \mathbb{R}^{n \times p}$ the features.
Require: $Y \in \{0,1\}^n$ the binary class label outputs.
Require: $k \in \mathbb{N}$ the number of neighbors to select for the $k$-Nearest Neighbors.
Require: $m \in \mathbb{N}$, the multiplier of the parameter of the distribution.
Ensure: Generated data $X_{new} \in \mathbb{R}^{q \times p}$ and $Y_{new} \in \{0,1\}^q$ with $q$ points created.
  Denote by $S_1$ the number of points labelled as the minority class and $S_0$ the number of points labelled as the majority class.
  Initialize $X_{new}$ and $Y_{new}$ as empty vectors.
  Obtain the median centroid ($\mu$) of the minority cluster.
  while $S_1 < S_0$ do
      Filter $\mathcal{D} = X_i | Y_i = 1$, the set of points labelled as minority class 1.
      Randomly choose $r \in \mathcal{D}$ and find the indices of its $k$ nearest neighbors, $r_1, \dots, r_k$.
      if Type is ‘Inverse distance (D)’ then
          Calculate the distances, $\boldsymbol{D} = [d_1, \dots, d_k]$ from $\mu$ to each nearest neighbour and obtain the reciprocal of each distance $\boldsymbol{D}^{-1} = [\frac{1}{d_1}, \dots, \frac{1}{d_k}]$. Then $\boldsymbol{\alpha} = \boldsymbol{D}^{-1} \times m$
      else if Type is ‘Uniform Vector (UV)’ then
          Generate a vector $\boldsymbol{\alpha} = \mathbf{1_k} \times m$, where $\mathbf{1_k} = [1, \dots, 1]$
      else if Type is ‘Uniform Distribution (UD)’ then
          Generate vector $\boldsymbol{U}$ of size $k$ from $uniform(0,1)$ distribution, then $\boldsymbol{\alpha} = \boldsymbol{U} \times m$.
      end if
      Use $\boldsymbol{\alpha}$, as parameters to the Dirichlet Distribution and generate random weights $w_j \sim Dir(\boldsymbol{\alpha})$
      $x^{new} \leftarrow \sum w_j x_{r_j}$ for all $j$ from 1 to $k$, as $\sum w_j = 1$
      $y^{new} \leftarrow 1$
      $S_1 = S_1 + 1$
      Append $x^{new}$ to $X_{new}$, append $y^{new}$ to $Y_{new}$
  end while
  return $X_{new}, Y_{new}$
```

Algorithm 4 FCRP SMOTE

```
Require: $X \in \mathbb{R}^{n \times p}$ the features, $Y \in \{0,1\}^n$ the binary class label outputs.
Require: $k \in \mathbb{N}$ the number of neighbors to select for the $k$-Nearest Neighbors.
Require: $\alpha \in \mathbb{R}$, scalar parameter to update preferences.
Ensure: Generated data $X_{new} \in \mathbb{R}^{q \times p}$ and $Y_{new} \in \{0,1\}^q$ with $q$ points created.
  Denote by $S_1$ the number of points labelled as the minority class and $S_0$ the number of points labelled as the majority class.
  Initialize $X_{new}$ and $Y_{new}$ as empty vectors.
  Filter $\mathcal{D} = X_i | Y_i = 1$, the set of points labelled as minority class 1 and obtain the median centroid ($\mu$) of the minority cluster.
  while $S1 < S_0$ do
      Randomly choose $r \in \mathcal{D}$ and find the indices of its $k$ nearest neighbors, $\{r_1, \dots, r_k\}$.
      Consider the normalized inverse distances, from $\mu$, to each nearest neighbour as initial preferences, $\boldsymbol{P} = \boldsymbol{D}^{-1}_{norm}$ and let the initial selection of the nearest neighbour occur with a probability $p_i$, $i$ from $1, \dots, k$.
      for N-1 do
          Choose the next nearest neighbour with the following updated probabilities $q_i$,
          $q_i = \begin{cases} \frac{p_i + \alpha}{1 + \alpha}, & \text{for previously chosen neighbour} \\ \frac{p_i}{1 + \alpha}, & \text{for other neighbours} \end{cases}$
          $p_i = q_i$
      end for
      Weights $w_j$ are obtained from the final preferences for each neighbour $p_i$.
      $x^{new} \leftarrow \sum (w_j \times x_{r_j})$ for all $j$ from 1 to $k$ and $y^{new} \leftarrow 1$
      $S_1 = S_1 + 1$
      Append $x^{new}$ to $X_{new}$, append $y^{new}$ to $Y_{new}$
  end while
  return $X_{new}, Y_{new}$
```

Bayesian methods provide an appealing solution. By introducing a prior distribution over these cluster mixing weights, Bayesian inference is observed through the possibility of adapting the parameter K, which is, in practical use cases, often unknown.

The Expectation-Maximization (EM) algorithm is an iterative technique for fitting Gaussian Mixture Models. In the context of Bayesian Gaussian Mixture Models, the EM algorithm iterates between the Expectation (E) step and the Maximization (M) step:

1. E-step: Expectation
   Estimate the posterior probability of each data point belonging to each cluster, known as the responsibility, using the current parameters.
2. M-step: Maximization
   Update the parameters (such as means, covariances, and mixing weights) based on the newly calculated responsibilities.

The posterior distribution with a fixed K is represented as:

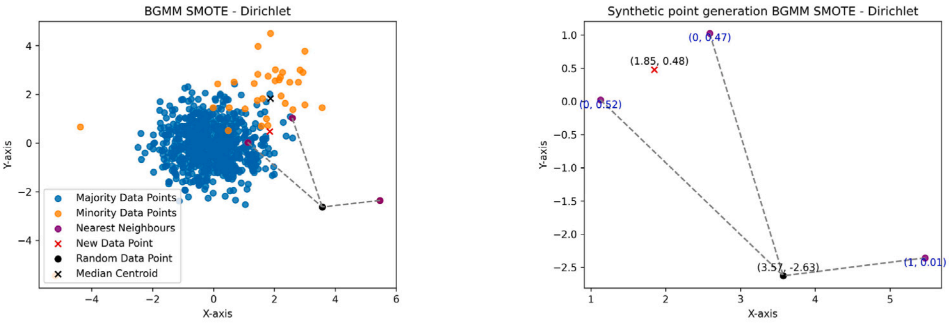
$$p(\theta|\boldsymbol{y}) \propto p(\theta) \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k f_k(y_i|\theta_k)$$

For the weight distribution, two distinct priors are integrated:

1. Dirichlet Distribution Prior
   Uses a finite mixture model with a symmetric Dirichlet prior $Dir(e_0)$ for the component weights:
   $$\pi_1, \dots, \pi_K \sim Dir_K(e_0), \pi_k \geq 0 \text{ for all k and } \sum_{k=1}^{K} \pi_k = 1$$
2. Dirichlet Process Prior
   Uses an infinite mixture model through the Dirichlet Process, approximating the Dirichlet Process inference algorithm using a truncated distribution, known as the Stick-breaking representation. This process allows for the flexible creation of an infinite number of components. Consider a stick length of 1, where $V_k \sim Beta(1, \alpha_0)$ for $k = 1, 2, 3, \dots$. The probabilities π_i for each component are calculated by the length taken away in each step of this process (Blei & Jordan, 2006):
   $$\pi_k(\boldsymbol{v}) = v_k \prod_{j=1}^{k-1} (1 - v_j)$$
   The stick-breaking process generates mixture weights from a distribution defined by the Dirichlet Process, allowing for flexibility in handling unknown or potentially infinite numbers of clusters in the data.

After obtaining the mixing weights, they are used for generating new synthetic data points, enriching the understanding of the data's underlying structure. In Bayesian Gaussian Mixture Models, cluster assignment is typically "soft", computing the probability of a point belonging to each cluster instead of definitively assigning it to a single cluster. This yields posterior probabilities $p(\theta|\boldsymbol{y})$, indicating the likelihood of individual data points belonging to each cluster within the mixture model.

Fig. 7 demonstrates generating a new point using suggested BGMM SMOTE technique. Within Fig. 7(b), the initial blue value in parenthesis



(a) This scenario occurs when an abnormal instance is chosen as a neighbouring point.

(b) The values within parentheses indicate (c_j, w_j).

Fig. 7. BGMM SMOTE. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Algorithm 5 BGMM SMOTE

```
Require: $X \in \mathbb{R}^{n\times p}$ the features.
Require: $Y \in \{0,1\}^n$ the binary class label outputs.
Require: $k \in \mathbb{N}$ the number of neighbors to select for the $k$-Nearest Neighbors.
Ensure: Generated data $X_{new} \in \mathbb{R}^{q\times p}$ and $Y_{new} \in \{0,1\}^q$ with $q$ points created.
Denote by $S_1$ the number of points labelled as the minority class and $S_0$ the number of points labelled as the majority class.
Initialize $X_{new}$ and $Y_{new}$ as empty vectors.
Filter $\mathcal{D} = X_i | Y_i = 1$, the set of points labelled as minority class 1.
Obtain the median centroid ($\mu$) of the minority cluster.
while $S_1 < S_0$ do
    Randomly choose $r \in \mathcal{D}$ and find the indices of its $k$ nearest neighbors, $r_1, \dots, r_k$.
    Estimate the mixing weights for each cluster for each selected data point, $r_1, \dots, r_k, \mu$.
    if Prior is 'Dirichlet Distribution (D)' then
        Use Bayesian Gaussian Mixture with Dirichlet distribution as the weight concentration prior.
    else if Prior is 'Dirichlet Process (DP)' then
        Use Bayesian Gaussian Mixture with Dirichlet Process as the weight concentration prior.
    end if
    Retrieve the mixing weights $w_j$ of the cluster to which the median centroid $\mu$ belongs.
    $x^{new} \leftarrow \frac{\sum(w_j \times x_{r_j})}{\sum w_j}$ for all $j$ from 1 to $k$.
    $y^{new} \leftarrow 1$
    $S_1 = S_1 + 1$
    Append $x^{new}$ to $X_{new}$, append $y^{new}$ to $Y_{new}$
end while
return $X_{new}, Y_{new}$
```

represents the cluster to which the data point most likely belongs to. Meanwhile, the subsequent number exhibits the assigned weights for each neighbour, with those belonging to the same cluster as the centroid receiving higher weights.

3.10. Role of weighting mechanism in mitigating abnormal instances

The weighting mechanism plays a critical role in reducing the influence of abnormal instances by addressing several key aspects, such as minimizing the influence of outliers, enhancing representative sampling and maintaining data integrity.

- In our methods, lower weights are assigned to such outliers due to their greater distance from the median centroid or lower probability of belonging to the core minority class distribution, ensuring they have minimal impact on generating synthetic samples while preserving the integrity and representativeness of the minority class data.
- By giving higher weights to instances closer to and more representative of the minority class, the synthetic samples generated are more reflective of the true distribution of the minority class. This leads to a more accurate and robust classification performance.
- The approach ensures that the synthetic data maintains the intrinsic properties of the minority class without being distorted by the extreme values introduced by abnormal instances. This is crucial for maintaining the validity and interpretability of the synthetic data.
- Abnormal instances are retained in our method because they may contain valuable information that can be useful to classification models. This retention ensures that potentially significant data is not lost during the resampling process.

3.11. Performance measures

There are many widely accepted performance measures for binary classification; however, not all are suitable when dealing with imbalanced data. In our analysis, we used three key measures: F1 Score, PR-AUC, and MCC. Introducing the notations TP (True Positives), TN (True Negatives), FP (False Positives), and FN (False Negatives),
Precision: $\frac{TP}{TP+FP}$, which measures the accuracy of the positive predictions, indicating the proportion of true positive instances among all instances predicted as positive.
Recall: $\frac{TP}{TP+FN}$, which measures the ability of the classifier to identify all positive instances, representing the proportion of true positive instances among all actual positive instances.

The selected measures are defined as follows:

- **F1 Score:** $2 \times \frac{Precision \times Recall}{Precision + Recall}$, which is the harmonic mean of Precision and Recall, providing a single metric that balances the trade-off between them, especially useful when there is an uneven class distribution.
- **PR-AUC:** The area under the Precision-Recall curve, which summarizes the trade-off between Precision and Recall across different thresholds.
- **MCC (Matthews Correlation Coefficient):** $\frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$, which takes into account all four confusion matrix categories and provides a balanced measure, even for imbalanced datasets.

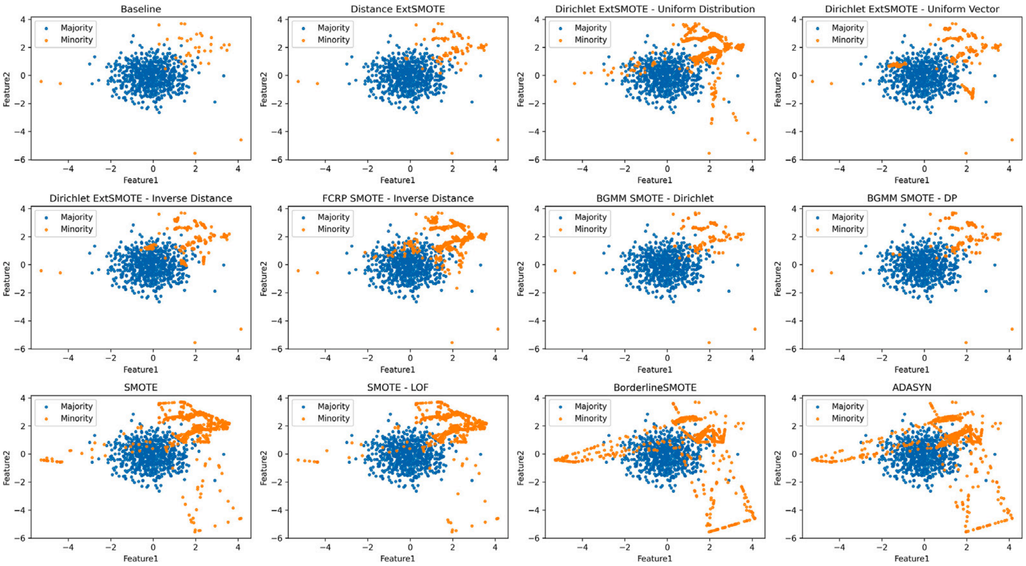


Fig. 8. Results comparison of one simulated dataset with two features and five abnormal instances.

4. Simulation results

4.1. Synthetic data generation

In our research, we performed an experimental study in which we created two distinct classes using a synthetic approach based on multivariate normal distributions. We generated data from two sets of bi-variate normal distributions, $\boldsymbol{X}_{minority-outliers} \sim \mathcal{N}_2(\boldsymbol{\mu_1}, \boldsymbol{\Sigma_1})$ and $\boldsymbol{X}_{majority} \sim \mathcal{N}_2(\boldsymbol{\mu_2}, \boldsymbol{\Sigma_2})$, representing the minority and majority classes, respectively. The simulation results in this manuscript are obtained using these distributions, including the means and covariances, which were specified as follows:

$$\boldsymbol{\mu_1} = \begin{bmatrix}0\\0\end{bmatrix}, \boldsymbol{\Sigma_1} = \begin{bmatrix}2 & 0\\0 & 2\end{bmatrix}, \boldsymbol{\mu_2} = \begin{bmatrix}3\\4\end{bmatrix}, \boldsymbol{\Sigma_2} = \begin{bmatrix}2 & 0\\0 & 2\end{bmatrix}$$

To accentuate the challenges posed by abnormal instances within the minority class, we took a deliberate step to introduce data points from a uniform distribution as outliers, $\boldsymbol{X}_{outliers} \sim Uniform(-10, 10)$. These abnormal instances were strategically added to the minority class to mimic scenarios where anomalous instances exist within the underrepresented class. Fig. 8 displays the bivariate plot of the resulting resampled data for one of these scenarios.

To assess performance across various dataset shapes, we synthetically generated data using 'make_moons' and 'make_circles' libraries library in python scikit-learn.datasets (Pedregosa et al., 2011). Figures in the appendix A demonstrate the capacity of the proposed methods to generate synthetic data points, effectively mitigating the influence of abnormal instances. In contrast, SMOTE and its extensions form an inaccurate data bridge that links these abnormal instances with normal ones.

4.2. Simulation setup

We generated a dataset comprising 1000 samples with two features, maintaining class imbalance ratios of 3 and 19. For an imbalance ratio of 3, the class distribution between majority and minority is 75:25, while for an imbalance ratio of 19, it is 95:5. We also considered two different outlier ratios, 0.05% and 0.1% in the analysis. Our initial step involved training classifiers without oversampling. This is the baseline method of our results. We used three different classifiers in the analysis: the Logistic Regression model, the K-Nearest Neighbors (KNN) classifier and theRandom Forest classifier.

Subsequently, we trained the models with oversampled data generated by various proposed and several well-performed SMOTE variants, including BorderlineSMOTE, ADASYN, SMOTE-LOF (with negative outlier factor < -1.5 as outliers) (Asniar et al., 2022) SMOTE IPF, DB-SMOTE, ProWSyn, DSMOTE, SMOTE TomekLinks, SMOTE ENN, cluster SMOTE, Gaussian SMOTE. All algorithms, except for the proposed once and most common SMOTE, SMOTE-LOF, BorderlineSMOTE, and ADASYN, were implemented using the smote-variant python tool package (Kovács, 2019) available at https://github.com/analyticalmindsltd/smote_variants.

Then, oversampled data was applied only to the 75% training fold while the 25% test fold was used for validation. This process was repeated across 100 simulation datasets to mitigate variability and ensure robustness in our analysis.

In terms of parameterization, all sampling algorithms employ a shared neighbourhood parameter set at $k = 5$, ensuring an unbiased comparison. This particular setting at $k = 5$ is not only the most commonly used but also serves as the prevalent standard across SMOTE and most of its variants (Chawla et al., 2002). For the classification models, the parameter k in KNN is also designated as five.

We also examined several key parameters to optimize the performance of our proposed methods. Specifically, we experimented with various values of alpha for FCRP SMOTE (0.1 and 0.5), different multipliers (0.01 and 100), types (distance, uniform distribution, uniform vector) for Dirichlet ExtSMOTE, and different priors (Dirichlet distribution, Dirichlet Process) for BGMM SMOTE. The optimal set of parameters, determined by the highest F1-score, was selected for each method, ensuring we identified the most effective configurations.

These final average F1-score, PR-AUC and MCC distributions when $OR \approx 0.05$ are graphically represented in Fig. 9 , Fig. 10 and Fig. 11 respectively. The boxplots to the left of the dashed red line represent the proposed methods. The simulation setup demonstrates that the proposed methods outperform existing methods across different machine learning classifiers, showcasing their robustness in classifier selection. Detailed results for $OR \approx 0.1$ for different IR, and accuracy measures are provided in Appendix B. The results show similar results across these measures.

4.3. Sensitivity analysis of the parameters

We systematically varied the parameter of interest for each proposed method while maintaining all other parameters constant. Specifically, for FCRP SMOTE, we tested various alpha values ranging from

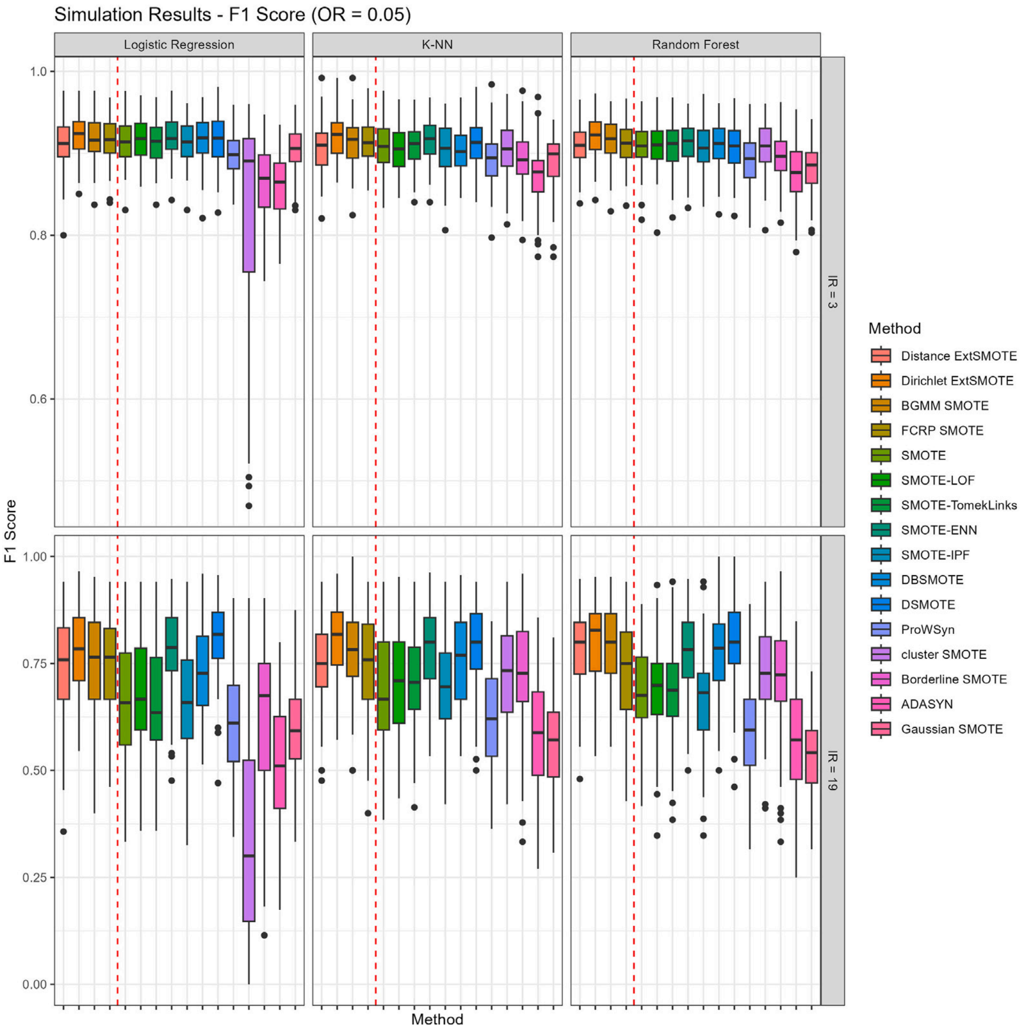


Fig. 9. F1 Scores across 100 simulated datasets with two different imbalance ratios and an outlier ratio of 0.05 were computed for three different classifiers. On the left-hand side of the dashed line are the results obtained from the proposed methods, while on the right-hand side are the results from existing methods. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

0.1 to 1. For Dirichlet ExtSMOTE, we experimented with different multipliers (0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100) and types (inverse distance, uniform distribution, uniform vector). For BGMM SMOTE, we explored different priors, including the Dirichlet distribution and the Dirichlet Process.

We then analysed the impact of these changes on key performance metrics, including F1-score, PR-AUC, and MCC, using three different classifiers: Logistic Regression, KNN, and Random Forest. This comprehensive approach allowed us to enhance the performance of our proposed algorithms by identifying the most effective parameter configurations. The results are detailed in Appendix C. The findings underscore the importance of carefully tuning parameters to achieve the best balance between model complexity and predictive accuracy in imbalanced datasets.

5. Application results

5.1. Datasets description

We collected 25 binary-class imbalanced data sets, commonly used in similar research scenarios, from the UCI machine learning repository (Newman et al., 1998; Pedregosa et al., 2011) to verify the effectiveness of our proposed algorithm. These datasets have 3–100 features, 159–6435 instances, and the IR varies from 1.25 to 41.40. Detailed descriptions of the data sets are provided in Table 1. To understand the spatial distribution of anomalies in the dataset, we used the Local Outlier Factor (LOF) outlier detection method (Breunig et al., 2000; Duan et al., 2007). The figures in appendix D show the principal component plots for the first two principal components, illustrating how the outliers are distributed within the minority class.

The proposed algorithms were also compared with the same SMOTE extensions used in the simulation study. These comparisons allowed us to evaluate the performance and robustness of our methods against established techniques under varying conditions.

5.2. Experimental settings

In our experimental setup, we aimed to ensure robust model performance by incorporating comprehensive cross-validation and calibration steps. Cross-validation involved splitting the training data into five subsets, training the model on four subsets, and validating it on the remaining subset.

After the cross-validation step, we calibrated the classifiers using isotonic regression. Calibration was performed through the ‘CalibratedClassifierCV‘ method in Python, which recalibrates the predicted probabilities to reflect the true likelihood of outcomes better. By integrating these steps, we ensured that the classifiers achieved high accuracy and provided well-calibrated probability estimates, crucial for reliable decision-making in imbalanced datasets.

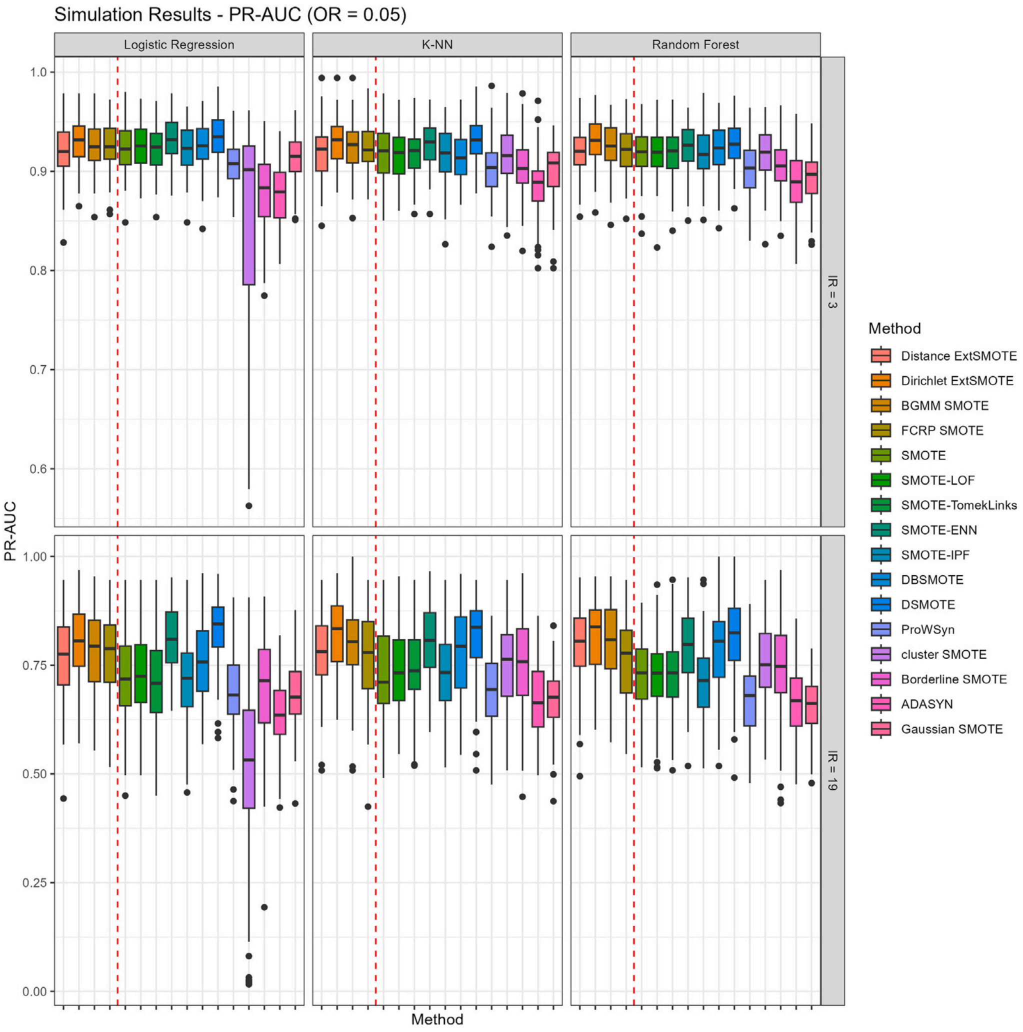


Fig. 10. PR-AUCs across 100 simulated datasets with two different imbalance ratios and an outlier ratio of 0.05 were computed for three different classifiers. On the left-hand side of the dashed line are the results obtained from the proposed methods, while on the right-hand side are the results from existing methods. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Additionally, to ensure the robustness and generalizability of our findings, we conducted 100 iterations for each of the 25 application datasets. This extensive iteration process allowed us to account for variability in the results and provided a thorough evaluation of each method's performance across different datasets.

The rest of the experimental setup and parameter settings used in the application setup are consistent with those in the simulation study.

Moreover, in this study, we employed a ranking mechanism to evaluate and compare the performance of various classification methods based on their mean F1 scores across multiple classifiers, datasets, and trials. Specifically, for each unique combination of classifier, dataset, and trial, we calculated the mean F1 score for each method. Then, we ranked these methods in descending order, with higher mean F1 scores receiving better ranks. Ties were handled using the “min” method, which assigns the minimum rank to all tied values, ensuring that the ranks consistently reflect the best possible performance.

Fig. 12 shows the density plots of the rankings of each trial of different datasets for different re-sampling methods. The proposed methods are highlighted in red. The proposed density curves particularly Dirichlet SMOTE, FCRP SMOTE, and BGMM SMOTE, are skewed to the right regardless of the classifier, indicating that they achieved higher ranks (lower values) compared to existing methods.

5.2.1. *Statistical analysis*

This ranking was further validated using the Friedman test for statistical significance (Friedman, 1937), and post-hoc pairwise comparisons were performed using Dunn's test with Holm adjustment (Dunn, 1964; Holm, 1979) to identify specific differences between methods, ultimately providing a robust comparative analysis of classification performance.

Tables 2, 3, and 4 present the test results for three classifiers, sorted according to average ranks. Dirichlet SMOTE consistently ranks first and rejects the null hypothesis (H_0: no difference in the ranks of the groups being compared) every time, regardless of the classifier used. The Dirichlet SMOTE significantly differs from other SMOTE variants and suggested methods with all classifiers. This indicates that Dirichlet SMOTE is the most effective method, followed closely by FCRP SMOTE and BGMM SMOTE, showcasing their robustness across different classifiers.

For the same experiment, Table 5, Table 6, and Table 7 compare the average F1 scores obtained by each resampling method and the baseline dataset. Results for other performance measures, PR-AUC and MCC are presented in the appendix E. The tables highlight the highest F1 score for each dataset in bold, indicating that Dirichlet SMOTE consistently delivers superior performance across most datasets, particularly with the Random Forest classifier. ADASYN, however, may exhibit inherent

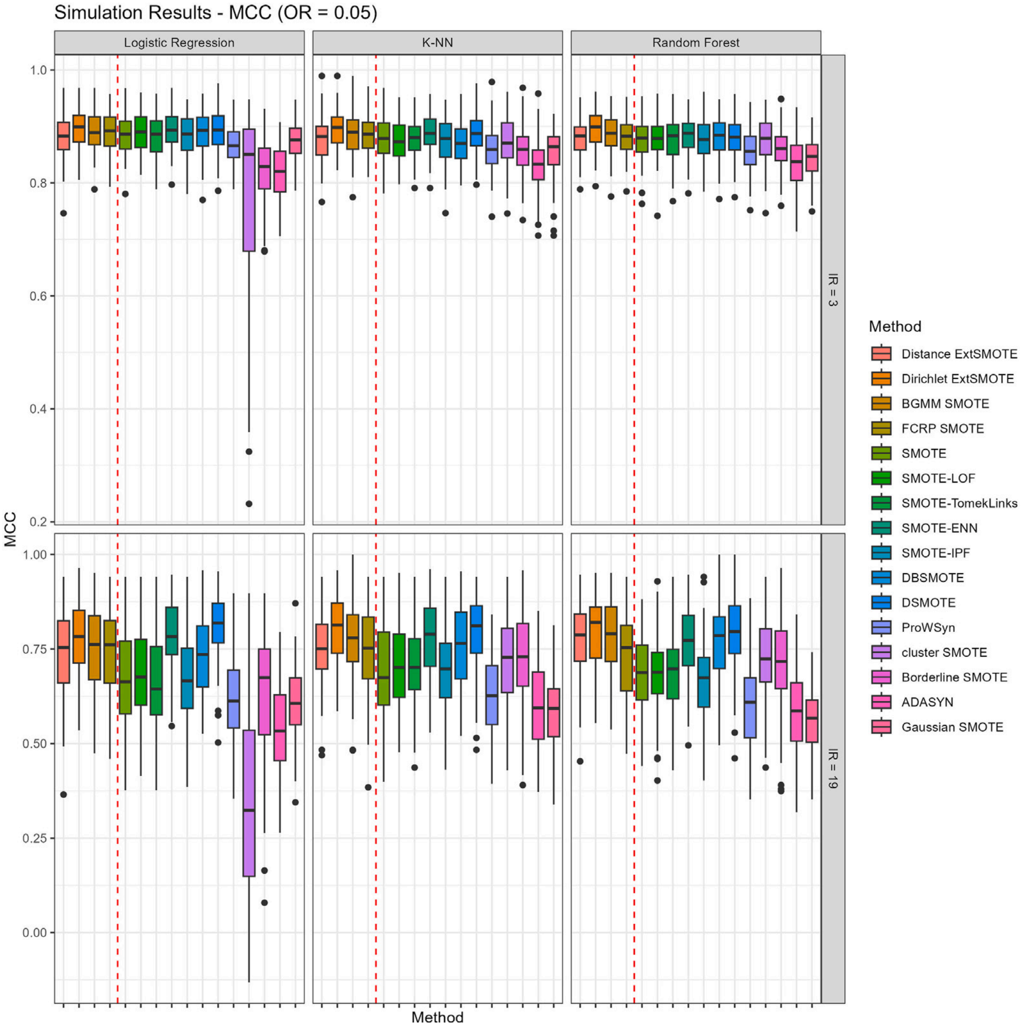


Fig. 11. MCCs across 100 simulated datasets with two different imbalance ratios and an outlier ratio of 0.05 were computed for three different classifiers. On the left-hand side of the dashed line are the results obtained from the proposed methods, while on the right-hand side are the results from existing methods. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

limitations in handling certain data distributions or noise levels, as observed in the banknote dataset.

6. Discussion

Class imbalance is a common challenge in many real-world datasets, where one class is significantly underrepresented compared to the other(s). In such situations, it can be beneficial to generate synthetic data that addresses the class imbalance and minimizes the impact of abnormal instances, especially within the minority class. Abnormal instances in the minority class can introduce significant bias and inaccuracies in synthetic samples, affecting the overall robustness and representativeness of the data. To tackle this issue effectively, synthetic data generation should be tailored to rebalance the class distribution while simultaneously reducing the influence of extreme data points.

This manuscript explores the fundamental concept of generating synthetic data with minimal impact from abnormal instances, recognizing its significance in various data-driven applications. By mitigating the influence of abnormal instances, we can improve the quality of synthetic data, ensuring that it is more reliable, dependable, and suitable for tasks such as machine learning, predictive modelling, and data augmentation.

By prioritizing the creation of robust synthetic data that is balanced and outlier-resistant, we can improve the performance of machine learning models in class imbalance problems. Upon visual inspection of our findings, we can see that the proposed tools significantly diminish the presence of data points drifting toward abnormal instances. Synthetic samples that accurately represent the minority class without being overly influenced by abnormal instances can lead to better generalization, more precise model predictions, and, ultimately, more reliable decision-making in applications like fraud detection, medical diagnosis, or rare-event forecasting. In this manuscript, we explore various strategies and techniques to achieve this dual objective, emphasizing the significance of generating synthetic data that minimizes the impact of abnormal instances, particularly in class imbalance, thereby contributing to developing more effective and robust data-driven solutions.

We have proposed several methods that offer differing approaches to the problem. In real-world data sets, we have seen differing results based on different approaches. In practice, it is critical for machine learning practitioners to examine different approaches to determine if any provide an advantage over the others.

Proposing multiple algorithms addresses the diverse characteristics of data, enabling tailored solutions for different scenarios. Our empirical evaluation highlights the strengths of each method, demonstrating that the optimal approach depends on the specific structure of the data. By offering various options and tools, we empower data users to accommodate varying complexities, dimensionalities, imbalance situations, nature of outliers, and underlying structure. Then they can use

Table 1
Characteristics of the binary class datasets used in the computational study.

No	Dataset	Instances	Features	Minority class	Majority class	%Minority	%Majority	IR	Presence of LOF Outliers
1	yeast6 (Nakai, 1996)	1484	8	EXC	Remaining classes	2.36	97.64	41.40	Yes
2	yeast5 (Nakai, 1996)	1484	8	EXC, ERL	Remaining classes	2.70	97.30	36.10	Yes
3	yeast-1289vs7	947	8	VAC	NUC, CYT, ERL, POX	3.17	96.83	30.57	Yes
4	yeast4 (Nakai, 1996)	1484	8	ME2	Remaining classes	3.44	96.56	28.10	Yes
5	yeast-2vs8 (Nakai, 1996)	483	8	POX	CYT	4.14	95.86	23.15	Yes
6	glass12357vs6 (German, 1987)	214	9	6	Remaining classes	4.21	95.79	22.78	Yes
7	yeast-1458vs7 (Nakai, 1996)	693	8	VAC	NUC, ME3, ME2, POX	4.33	95.67	22.10	Yes
8	oil (Lemaitre et al., 2017)	937	49	minority	majority	4.38	95.62	21.85	No
9	abalone9_18 (Nash et al., 1995)	731	7	9, 18	Remaining classes	5.75	94.25	16.40	Yes
10	glass12367vs5 (German, 1987)	214	9	5	Remaining classes	6.07	93.93	15.46	Yes
11	thyroid_sick (Lemaitre et al., 2017)	3772	52	sick	healthy	6.12	93.88	15.33	Yes
12	yeast-1vs7 (Nakai, 1996)	459	8	VAC	NUC	6.54	93.46	14.30	Yes
13	us_crime (Lemaitre et al., 2017)	1994	100	>0.65	<=0.65	7.52	92.48	12.29	Yes
14	glass12vs5 (German, 1987)	159	9	5	1, 2	8.18	91.82	11.23	Yes
15	spectrometer (mis, 1988)	531	93	>=44	<44	8.47	91.53	10.80	Yes
16	landsat_satellite (Srinivasan, 1993)	6435	36	2	Remaining classes	9.73	90.27	9.28	Yes
17	mfeatmor0 (Duin, 2024)	2000	6	0, 1	Remaining classes	10.00	90.00	9.00	Yes
18	yeast3 (Nakai, 1996)	1484	8	ME3	Remaining classes	10.98	89.02	8.10	Yes
19	mfeatmor01 (Duin, 2024)	2000	6	0	Remaining classes	20.00	80.00	4.00	Yes
20	glass123vs567 (German, 1987)	214	9	5, 6, 7	Remaining classes	23.83	76.17	3.20	Yes
21	parkinsons (Little, 2008)	195	22	1	0	24.62	75.38	3.06	Yes
22	habermans_survival (Haberman, 1999)	306	3	2	1	26.47	73.53	2.78	Yes
23	glass23567vs1 (German, 1987)	214	9	1	Remaining classes	32.71	67.29	2.06	Yes
24	breast_cancer (Wolberg et al., 1995)	569	30	M	B	37.26	62.74	1.68	Yes
25	banknote (Lohweg, 2013)	1372	4	1	Remaining classes	44.46	55.54	1.25	Yes

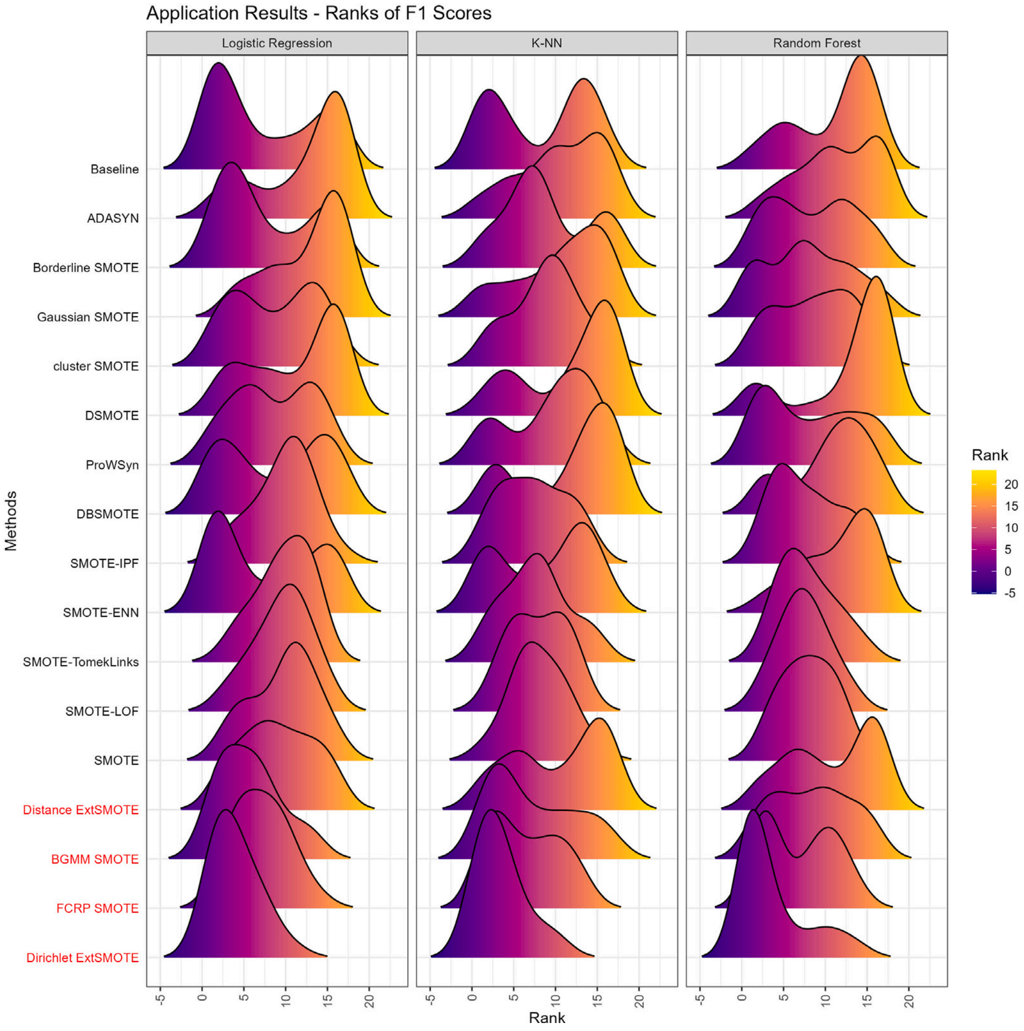


Fig. 12. F1 Score Ranks for the datasets with 100 × 5-fold cross validation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 2
Statistical analysis results of F1 Scores of comparative algorithms at the confidence level of $\alpha = 0.05$ based on Logistic classifier.

	Method	Average_Rank	Distance ExtSMOTE	Dirichlet ExtSMOTE	FCRP SMOTE	BGMM SMOTE
1	Dirichlet ExtSMOTE	4.49	Rejected	–	Rejected	Rejected
2	BGMM SMOTE	6.37	Rejected	Rejected	Rejected	–
3	FCRP SMOTE	7.15	Rejected	Rejected	–	Rejected
4	Baseline	7.40	Rejected	Rejected	Not Rejected	Rejected
5	SMOTE-ENN	7.64	Rejected	Rejected	Not Rejected	Rejected
6	Borderline SMOTE	7.75	Rejected	Rejected	Rejected	Rejected
7	DBSMOTE	8.46	Not Rejected	Rejected	Rejected	Rejected
8	ProWSyn	8.60	Not Rejected	Rejected	Rejected	Rejected
9	Distance ExtSMOTE	8.65	–	Rejected	Rejected	Rejected
10	cluster SMOTE	8.93	Not Rejected	Rejected	Rejected	Rejected
11	SMOTE-LOF	9.16	Rejected	Rejected	Rejected	Rejected
12	SMOTE-IPF	9.31	Rejected	Rejected	Rejected	Rejected
13	SMOTE-TomekLinks	9.34	Rejected	Rejected	Rejected	Rejected
14	SMOTE	9.40	Rejected	Rejected	Rejected	Rejected
15	DSMOTE	10.94	Rejected	Rejected	Rejected	Rejected
16	ADASYN	12.01	Rejected	Rejected	Rejected	Rejected
17	Gaussian SMOTE	12.42	Rejected	Rejected	Rejected	Rejected

Table 3
Statistical analysis results of F1 Scores of comparative algorithms at the confidence level of $\alpha = 0.05$ based on KNN classifier.

	Method	Average_Rank	Distance ExtSMOTE	Dirichlet ExtSMOTE	FCRP SMOTE	BGMM SMOTE
1	Dirichlet ExtSMOTE	4.43	Rejected	–	Rejected	Rejected
2	FCRP SMOTE	6.45	Rejected	Rejected	–	Rejected
3	BGMM SMOTE	7.37	Rejected	Rejected	Rejected	–
4	SMOTE-LOF	7.59	Rejected	Rejected	Rejected	Not Rejected
5	SMOTE-IPF	7.64	Rejected	Rejected	Rejected	Not Rejected
6	SMOTE-TomekLinks	7.71	Rejected	Rejected	Rejected	Not Rejected
7	SMOTE	7.81	Rejected	Rejected	Rejected	Not Rejected
8	Baseline	8.27	Rejected	Rejected	Rejected	Rejected
9	Borderline SMOTE	8.48	Rejected	Rejected	Rejected	Rejected
10	SMOTE-ENN	8.71	Rejected	Rejected	Rejected	Rejected
11	cluster SMOTE	8.75	Rejected	Rejected	Rejected	Rejected
12	ProWSyn	8.93	Rejected	Rejected	Rejected	Rejected
13	Distance ExtSMOTE	10.11	–	Rejected	Rejected	Rejected
14	ADASYN	10.19	Not Rejected	Rejected	Rejected	Rejected
15	Gaussian SMOTE	10.42	Not Rejected	Rejected	Rejected	Rejected
16	DBSMOTE	11.33	Rejected	Rejected	Rejected	Rejected
17	DSMOTE	11.57	Rejected	Rejected	Rejected	Rejected

Table 4
Statistical analysis results of F1 Scores of comparative algorithms at the confidence level of $\alpha = 0.05$ based on Random Forest classifier.

	Method	Average_Rank	Distance ExtSMOTE	Dirichlet ExtSMOTE	FCRP SMOTE	BGMM SMOTE
1	Dirichlet ExtSMOTE	4.55	Rejected	–	Rejected	Rejected
2	FCRP SMOTE	6.93	Rejected	Rejected	–	Rejected
3	Gaussian SMOTE	7.82	Rejected	Rejected	Rejected	Rejected
4	SMOTE-LOF	7.86	Rejected	Rejected	Rejected	Rejected
5	ProWSyn	8.09	Rejected	Rejected	Rejected	Not Rejected
6	SMOTE-IPF	8.19	Rejected	Rejected	Rejected	Not Rejected
7	SMOTE-TomekLinks	8.21	Rejected	Rejected	Rejected	Not Rejected
8	SMOTE	8.25	Rejected	Rejected	Rejected	Not Rejected
9	Borderline SMOTE	8.27	Rejected	Rejected	Rejected	Not Rejected
10	BGMM SMOTE	8.38	Rejected	Rejected	Rejected	–
11	cluster SMOTE	8.56	Rejected	Rejected	Rejected	Not Rejected
12	Baseline	9.75	Rejected	Rejected	Rejected	Rejected
13	DBSMOTE	10.27	Not Rejected	Rejected	Rejected	Rejected
14	SMOTE-ENN	10.33	Not Rejected	Rejected	Rejected	Rejected
15	Distance ExtSMOTE	10.41	–	Rejected	Rejected	Rejected
16	ADASYN	10.57	Not Rejected	Rejected	Rejected	Rejected
17	DSMOTE	12.15	Rejected	Rejected	Rejected	Rejected

multiple accuracy measures to determine which method performs best for particular needs as we have illustrated in the applications.

Given that results can vary significantly with different classifiers, selecting the appropriate classifier for the dataset is crucial. The choice of classifier can greatly impact the performance and accuracy of the model. For example, in our study, we used three classifiers: Logistic Regression, k-Nearest Neighbors (k-NN), and Random Forest. Notably, Dirichlet ExtSMOTE outperforms most other SMOTE variants in terms of F1 score, MCC, and PR-AUC, especially with a Random Forest classifier.

Random Forest is often more effective for imbalanced data due to its ensemble approach, combining multiple decision trees to enhance accuracy and robustness. This helps capture complex patterns and interactions within the data, which is crucial for imbalanced datasets. Additionally, Random Forest is more resilient to outliers and noise compared to Logistic Regression, which assumes linear relationships, and k-NN, which can struggle with high-dimensional data and noise. Studies show that Random Forest generally achieves higher accuracy and better generalization in imbalanced datasets compared to Logistic Regression and k-NN (Couronné et al., 2018; Shah et al., 2020). However, This highlights the need to experiment with multiple classifiers and select

Table 5
F1 Score results of 17 comparative algorithms on 25 imbalanced datasets using 100 × 5 fold cross-validation with a Logistic Regression classifier.

Dataset	Distance	Dirichlet	BGMM	FCRP	SMOTE	SMOTE	SMOTE	SMOTE	SMOTE	DBSMOTE	ProWSyn	DSMOTE	cluster	Gaussian	Borderline	ADASYN	Baseline
	ExtSMOTE	ExtSMOTE	SMOTE	SMOTE		LOF	TomekLinks	ENN	IPF				SMOTE	SMOTE	SMOTE		
yeast6	0.3812	0.3843	0.4213	0.3634	0.3065	0.3078	0.3057	**0.4315**	0.306	0.4157	0.2939	0.3235	0.2381	0.257	0.4192	0.2579	**0.4315**
yeast5	0.4065	0.4064	0.4311	0.3924	0.3637	0.3601	0.3589	0.48	0.3583	**0.5557**	0.3336	0.4354	0.3211	0.3128	0.4382	0.2997	0.48
yeast-1289vs7	0.1797	**0.2286**	0.1918	0.1738	0.1758	0.1754	0.1663	0.2071	0.1651	0.0939	0.1618	0.2079	0.2184	0.1979	0.2145	0.1645	0.2071
yeast4	0.2423	0.2858	0.3062	0.265	0.2571	0.2598	0.2604	0.2584	0.2625	**0.382**	0.2589	0.2478	0.2658	0.2616	0.3382	0.2399	0.2584
yeast-2vs8	0.2189	0.3927	0.2614	0.2517	0.4727	0.4704	0.448	0.5733	0.4454	0.2802	0.4979	0.6133	**0.6169**	0.4999	0.2075	0.1347	0.5733
glass12357vs6	0.4901	0.6299	0.5467	0.5704	0.6133	0.6133	0.6133	0.48	0.6133	**0.6489**	0.6133	0.4218	0.6079	0.4329	0.6133	0.6133	0.48
yeast-1458vs7	0.1656	**0.1978**	0.1685	0.1816	0.1696	0.1689	0.1653	0	0.1631	0.1256	0.1392	0.1592	0.1361	0.1765	0.1777	0.164	0
oil	0.5596	0.5582	0.5141	0.5197	0.4951	0.4954	0.5009	**0.5598**	0.5001	0.5046	0.5082	0.4509	0.4875	0.3919	0.482	0.4936	**0.5598**
abalone9_18	0.5101	0.5089	0.4759	0.4841	0.4245	0.4276	0.4427	0.4733	0.4426	**0.5401**	0.4785	0.4739	0.4609	0.3753	0.5271	0.3908	0.4733
glass12367vs5	0.4654	0.5624	0.5612	0.5015	0.5364	0.5366	0.5306	0.3733	0.5324	0.2267	0.5504	0.3438	0.4302	0.4733	**0.574**	0.5454	0.3733
thyroid_sick	0.5922	0.5995	0.5655	0.5814	0.5567	0.5611	0.5539	0.7067	0.5531	**0.7103**	0.5858	0.5894	0.6357	0.3493	0.5318	0.5198	0.7067
yeast-1vs7	0.3625	0.3536	**0.3747**	0.3435	0.2995	0.3001	0.303	0.1833	0.3041	0.2636	0.3019	0.3102	0.334	0.3457	0.3337	0.2969	0.1833
us_crime	0.4902	0.4986	0.5016	0.5014	0.4841	0.4832	0.48	0.4924	0.4815	0.4791	0.5	0.5171	0.5016	0.4764	**0.53**	0.4679	0.4924
glass12vs5	0.6381	0.7966	0.6984	0.779	0.7887	0.7874	0.7845	0.7381	0.7845	0.5017	0.7969	0.6219	0.7582	0.6566	0.7511	**0.8005**	0.7381
spectrometer	0.8538	0.857	0.8618	0.8549	0.8473	0.8447	0.8493	0.8617	0.8492	0.8472	0.8575	0.8284	0.8595	0.7929	0.8514	0.8334	**0.8676**
landsat_satellite	0.2907	0.2929	0.2923	0.2919	0.2941	0.2935	0.2936	0.0453	0.2941	0.0528	**0.2978**	0.1563	0.2961	0.2745	0.2934	0.2894	0.0453
mfeatmor0	0.9923	0.9931	0.9923	0.9923	0.9916	0.9923	0.9911	0.9909	0.9914	0.9923	**0.9945**	0.9731	0.9871	0.9764	0.9923	0.9899	0.9923
yeast3	0.7032	0.6992	0.703	0.6936	0.6425	0.6434	0.6445	**0.7582**	0.6446	0.7106	0.6695	0.7302	0.6445	0.6492	0.6409	0.6323	**0.7582**
mfeatmor01	0.9172	0.9174	0.9132	0.9157	0.9078	0.9114	0.9116	0.9239	0.9079	0.9238	0.907	0.9059	0.9062	0.8624	0.8335	0.7638	**0.9258**
glass123vs567	0.8303	0.8573	**0.8657**	0.857	0.8434	0.8421	0.8445	0.7932	0.8449	0.8401	0.8334	0.8373	0.8432	0.8315	0.8345	0.8391	0.7902
parkinsons	0.6616	**0.6794**	0.6677	0.6658	0.658	0.6596	0.6569	0.6752	0.6591	0.6739	0.6426	0.6344	0.6565	0.6192	0.6677	0.6678	0.6752
haber-mans_survival	0.4874	**0.5196**	0.5028	0.5017	0.4981	0.5	0.4996	0.3157	0.5013	0.4924	0.5008	0.3419	0.515	0.4777	0.4859	0.5097	0.3157
glass23567vs1	0.6362	0.6494	0.6423	0.6385	0.6235	0.6247	0.6269	**0.6594**	0.6222	0.625	0.6323	0.6466	0.6406	0.6399	0.6462	0.642	**0.6594**
breast_cancer	0.9643	0.9661	0.9663	0.9647	0.9629	0.9627	0.9617	**0.9664**	0.9634	0.9646	0.9612	0.9318	0.9624	0.9631	0.9505	0.9501	0.9653
banknote	0.9888	**0.9903**	0.9902	0.9897	0.9892	0.9891	0.9892	0.9892	0.9891	0.9889	0.9894	0.9412	0.9892	0.972	0.9892	NA	0.9892

Table 6
F1 Score results of 17 comparative algorithms on 25 imbalanced datasets using 100 × 5 fold cross-validation with a K-NN classifier.

Dataset	Distance	Dirichlet	BGMM	FCRP	SMOTE	SMOTE	SMOTE	SMOTE	SMOTE	DBSMOTE	ProWSyn	DSMOTE	cluster	Gaussian	Borderline	ADASYN	Baseline
	ExtSMOTE	ExtSMOTE	SMOTE	SMOTE		LOF	TomekLinks	ENN	IPF				SMOTE	SMOTE	SMOTE		
yeast6	0.4724	0.5098	**0.5153**	0.5001	0.4047	0.4046	0.3992	0.4978	0.4024	0.4454	0.3615	0.4614	0.4057	0.3628	0.3985	0.362	0.4978
yeast5	0.5118	0.6008	0.5543	0.574	0.4949	0.4914	0.4845	**0.6487**	0.4891	0.6413	0.4287	0.5988	0.5135	0.3931	0.5397	0.4394	**0.6487**
yeast-1289vs7	0.1364	0.2636	0.2465	0.2231	0.2402	0.2452	0.2379	0.1943	0.2401	0.1016	0.1892	**0.3036**	0.2699	0.232	0.2406	0.2379	0.1943
yeast4	0.3266	**0.4089**	0.3771	0.3882	0.3557	0.3543	0.3662	0.2914	0.3608	0.3051	0.3177	0.3479	0.351	0.3713	0.3631	0.349	0.2914
yeast-2vs8	0.48	0.5513	0.4342	0.4188	0.4008	0.4004	0.3755	**0.6033**	0.3794	0.58	0.4085	0.5783	0.4442	0.413	0.3476	0.3704	**0.6033**
glass12357vs6	**0.8**	0.7783	0.658	0.7483	0.7833	0.7863	0.7917	0.7333	0.7927	0.7333	0.726	0.6811	0.7353	0.5037	0.783	0.7833	0.7333
yeast-1458vs7	0.0333	**0.1732**	0.0494	0.0841	0.1403	0.1403	0.1426	0.04	0.1472	0.1071	0.1591	0.1204	0.1545	0.1238	0.1399	0.1326	0.04
oil	0.4897	0.6003	0.5557	0.563	0.5698	0.5725	0.5722	0.5463	0.5732	0.5927	0.5659	0.3487	0.5536	0.4232	**0.6147**	0.5614	0.5463
abalone9_18	0.3519	0.4253	**0.4507**	0.4432	0.4044	0.4032	0.4089	0.4205	0.4089	0.4025	0.4058	0.4292	0.3629	0.4027	0.4185	0.3804	0.4205
glass12367vs5	0.5733	0.6712	0.633	0.6707	0.6631	0.6674	0.6823	0.4171	**0.6926**	0.2	0.661	0.3777	0.5714	0.5552	0.6508	0.6387	0.4171
thyroid_sick	0.5249	0.5566	0.5612	0.568	0.5779	0.575	0.5737	0.5472	0.5775	0.4542	0.5461	0.4904	0.5686	0.4978	0.56	**0.58**	0.5472
yeast-1vs7	0.3249	0.4092	0.3348	0.3822	0.4041	0.4038	0.3936	**0.4537**	0.3937	0.2564	0.3545	0.3693	0.4267	0.3583	0.4304	0.3871	**0.4537**
us_crime	0.3877	0.4125	0.4125	0.4186	0.4059	0.4054	0.4091	0.4025	0.4048	0.3405	**0.4332**	0.4135	0.4094	0.4096	0.4065	0.3948	0.4025
glass12vs5	0.28	0.576	0.5293	0.5582	0.6518	0.6347	0.6679	0.3314	0.6643	0.2133	0.6611	0.2976	0.6165	**0.6892**	0.6443	0.6577	0.3314
spectrometer	0.8018	0.8747	0.7645	0.8329	0.8416	0.8461	0.8492	0.836	0.8469	0.8107	0.824	0.8039	0.8242	**0.8754**	0.7935	0.8223	0.8478
landsat_satellite	0.6546	**0.7056**	0.6607	0.6837	0.7011	0.7011	0.6969	0.7029	0.6979	0.7012	0.6865	0.515	0.6925	0.6032	0.6871	0.6946	0.7029
mfeatmor0	0.9923	**0.9933**	0.9923	0.9923	0.9847	0.9888	0.986	0.9899	0.9868	0.9923	0.9885	0.9498	0.9876	0.9646	0.9923	0.9875	0.9923
yeast3	0.714	0.7359	0.7478	0.7421	0.7116	0.712	0.7168	0.7572	0.7147	0.6762	**0.7589**	0.7428	0.6825	0.7151	0.6325	0.6865	0.7572
mfeatmor01	0.9488	0.9501	0.9505	0.9479	0.9453	0.9452	0.9462	0.9509	0.9461	0.9375	0.9453	0.9357	0.9418	0.9422	0.919	0.8869	**0.9559**
glass123vs567	0.8628	**0.8753**	0.8704	0.8716	0.8542	0.8528	0.8552	0.8561	0.8523	0.8512	0.8709	0.8253	0.8479	0.8703	0.843	0.8441	0.8578
parkinsons	0.8586	0.8601	0.8446	0.8566	0.8283	0.8304	0.8278	0.7668	0.8344	0.7293	0.8062	0.7379	0.8318	0.8382	0.8493	**0.8645**	0.7668
haber-mans_survival	0.2906	0.4161	0.3847	0.3847	0.4123	0.4118	0.4054	0.1628	0.4145	0.2838	0.4458	0.3086	0.3882	**0.455**	0.4118	0.4159	0.1628
glass23567vs1	0.7238	**0.7536**	0.7348	0.7375	0.714	0.7136	0.7146	0.6828	0.7127	0.6651	0.6875	0.6636	0.7361	0.6807	0.7195	0.7283	0.6828
breast_cancer	0.947	**0.9544**	0.9493	0.9487	0.9473	0.948	0.9455	0.9438	0.948	0.9432	0.9453	0.9219	0.9455	0.9439	0.9382	0.9385	0.9434
banknote	0.9984	**0.9987**	0.9984	0.9984	0.9984	0.9984	0.9984	0.9984	0.9984	0.9984	0.9982	0.947	0.9984	0.9859	0.9984	NA	0.9984

the one that best captures the underlying patterns in the data, ensuring robust and reliable outcomes.

Use of synthetic data has possible implications for safety and bias. This is especially relevant when working with human data, including medical and social data. There is evidence of bias that can be introduced into systems for the related problem of data augmentation (e.g., (Jain et al., 2018)). Additionally, it has been observed that tools such as up- and down-sampling and augmentation are potential tools to combat bias in datasets, though this remains challenging (Pastaltzidis et al., 2022; Sharma et al., 2020; Tomalin et al., 2021). This is an area of ongoing research, but practitioners should exert caution when employing tools for data imbalance or data augmentation, both with the output variables and with informative features.

6.1. Practical implications

The enhancements to SMOTE for handling imbalanced data with abnormal minority instances have significant practical implications across various domains, including healthcare, finance, and fraud detection. By introducing advanced SMOTE techniques that are specifically designed to address the presence of abnormal instances within minority classes, our study provides robust tools for improving the accuracy and reliability of predictive models. These enhanced methods facilitate better detection of rare but critical events, such as identifying rare diseases, detecting fraudulent transactions, or uncovering defective products. As a result, organizations can make more informed decisions, reduce risks, and optimize resource allocation. Implementing these methods empowers data practitioners to effectively tackle complex imbalanced data scenarios, ultimately leading to more precise and impactful outcomes.

6.2. Theoretical foundations of proposed methods

6.2.1. Distance ExtSMOTE

Using normalized inverse distances as weights in Distance ExtSMOTE provides a straightforward and effective mechanism for generating synthetic samples that maintain the integrity of the minority class distribution. By assigning higher weights to closer neighbours, the method captures local density variations and preserves the underlying structure of the minority class. This approach also mitigates the impact of outliers, as distant outliers receive smaller weights, reducing their

Table 7
F1 Score results of 17 comparative algorithms on 25 imbalanced datasets using 100 × 5 fold cross-validation with a Random Forest classifier.

Dataset	Distance ExtSMOTE	Dirichlet ExtSMOTE	BGMM SMOTE	FCRP SMOTE	SMOTE	SMOTE LOF	SMOTE TomekLinks	SMOTE ENN	SMOTE IPF	DBSMOTE	ProWSyn	DSMOTE	cluster SMOTE	Gaussian SMOTE	Borderline SMOTE	ADASYN	Baseline
yeast6	0.5382	0.6059	**0.6232**	0.6024	0.5449	0.5516	0.5456	0.5086	0.5519	0.5124	0.4946	0.5107	0.5656	0.539	0.5272	0.5301	0.5077
yeast5	0.6019	0.6213	0.6171	0.6249	0.5287	0.533	0.5245	0.5959	0.5197	0.5845	0.5095	**0.6442**	0.5513	0.6196	0.529	0.5123	0.5969
yeast-1289vs7	0.0643	0.147	0.0929	0.1196	0.1771	0.1741	0.1847	0.2651	0.1751	0.1012	0.1523	0.2933	0.2773	**0.3319**	0.2354	0.1451	0.2632
yeast4	0.262	0.3913	0.3735	0.3816	0.4099	0.4081	0.3973	0.327	0.3923	0.3319	0.3812	0.3293	0.4313	0.3941	**0.4374**	0.4006	0.3307
yeast-2vs8	0.5393	0.5573	0.5014	0.4629	0.4297	0.4273	0.4202	0.5599	0.4215	0.5823	0.4859	0.5704	0.511	**0.6173**	0.338	0.2939	0.5601
glass12357vs6	0.6933	**0.7154**	0.6453	0.6329	0.6067	0.6127	0.6067	0.56	0.5993	0.6033	0.6253	0.5253	0.6107	0.6105	0.612	0.5887	0.565
yeast-1458vs7	0.0023	**0.1624**	0.043	0.091	0.1335	0.1279	0.1296	0.0021	0.1218	0.0026	0.1376	0.1246	0.0997	0.0441	0.1396	0.1253	0.0048
oil	0.4213	0.5253	0.4851	0.55	0.4825	0.4881	0.4949	0.4496	0.4952	0.2503	**0.5758**	0.4259	0.427	0.5543	0.531	0.4684	0.4538
abalone9_18	0.3747	**0.4172**	0.3851	0.4	0.3711	0.3707	0.3774	0.3634	0.3755	0.3975	0.3962	0.3744	0.3601	0.3823	0.3963	0.3567	0.3692
glass12367vs5	0.6526	**0.8326**	0.6594	0.7103	0.7207	0.7191	0.7339	0.5929	0.7345	0.7129	0.7586	0.6097	0.6606	0.7248	0.7426	0.7187	0.591
thyroid_sick	0.858	**0.8835**	0.8687	0.8587	0.8749	0.8704	0.8722	0.8592	0.8728	0.8551	0.8489	0.8127	0.8778	0.8694	0.8683	0.8711	0.8582
yeast-1vs7	0.1601	0.3189	0.2205	0.2946	0.3581	0.3563	0.3413	0.3519	0.3542	0.1272	0.2641	**0.4023**	0.3603	0.3787	0.3596	0.3132	0.3512
us_crime	0.4585	0.48	0.4607	0.4876	0.4986	0.4986	0.4958	0.4899	0.4992	0.3462	0.5187	0.4831	0.4861	**0.5229**	0.4981	0.4882	0.4932
glass12vs5	0.676	**0.8031**	0.7708	0.7831	0.7842	0.7916	0.789	0.6836	0.7862	0.7435	0.7885	0.5678	0.7306	0.782	0.785	0.7821	0.6821
spectrometer	0.8233	**0.8501**	0.8066	0.8283	0.8127	0.82	0.8144	0.8359	0.8128	0.8303	0.8084	0.8053	0.8116	0.82	0.8126	0.7978	0.8236
landsat_satellite	0.6514	0.6746	0.6696	0.6712	0.6828	0.6791	0.6812	0.6642	0.6812	0.6628	0.6798	0.6492	**0.6867**	0.6384	0.6836	0.6835	0.6641
mfeatmor0	0.9923	**0.9925**	0.9923	0.9923	0.9889	0.9914	0.9887	0.9891	0.9883	0.9909	0.9912	0.9528	0.9877	0.9879	0.9923	0.9871	0.9923
yeast3	0.7656	0.7866	0.7776	0.782	0.7839	0.7852	0.7845	0.759	0.7852	0.7678	**0.7867**	0.7521	0.783	0.7684	0.7776	0.7758	0.7602
mfeatmor01	0.9527	0.9579	0.9559	0.9522	0.9504	0.9511	0.9501	0.9533	0.95	0.9545	0.9568	0.9363	0.9484	0.9538	0.9317	0.891	**0.9581**
glass123vs567	0.8906	**0.9037**	0.8905	0.8998	0.8883	0.8882	0.8869	0.8859	0.8939	0.8871	0.8963	0.8191	0.8877	0.8898	0.8772	0.8764	0.8854
parkinsons	0.7659	**0.8106**	0.7976	0.8013	0.8053	0.8089	0.8093	0.7833	0.8104	0.7956	0.787	0.7176	0.8082	0.7904	0.7982	0.8069	0.7824
habermans_survival	0.3903	0.4228	0.4006	0.4283	0.3855	0.3825	0.4044	0.322	0.4089	0.3415	0.4321	0.3177	0.3793	**0.4566**	0.3762	0.3837	0.3197
glass23567vs1	0.8147	**0.8457**	0.8274	0.8234	0.8177	0.818	0.8152	0.8179	0.814	0.8262	0.7899	0.8124	0.8165	0.7873	0.814	0.8107	0.8223
breast_cancer	0.9474	**0.9521**	0.9481	0.9493	0.9468	0.947	0.9483	0.9445	0.947	0.9471	0.946	0.9219	0.9472	0.9459	0.944	0.9497	0.9447
banknote	0.9929	**0.994**	0.9929	0.9935	0.9926	0.9925	0.9925	0.9922	0.9926	0.9924	0.9925	0.938	0.9926	0.9901	0.9922	NA	0.9922

influence on synthetic samples. Consequently, synthetic instances are more representative of the minority class, enhancing the classifier's ability to generalize. Normalizing the inverse distances ensures that the sum of the weights is 1, maintaining balanced contributions from relevant neighbours and minimizing noise. This leads to lower bias in the synthetic samples and improved model performance on imbalanced datasets.

This method is suitable for datasets where the minority class has a relatively dense and well-defined structure. However, this approach has limitations. The deterministic nature of using fixed normalized inverse distances can result in less diverse synthetic samples, potentially missing the complexity of the minority class distribution.

6.2.2. Dirichlet ExtSMOTE

Using Dirichlet samples to define weights in Dirichlet ExtSMOTE, rather than relying solely on inverse distances, offers several significant advantages. The primary benefit is the introduction of variability in the synthetic sample generation process. By drawing weights from a Dirichlet distribution parameterized uniformly or by inverse distances, the method ensures that the weights can differ each time, producing a more diverse set of synthetic samples even for the same set of neighbours. This variability helps better approximate the true underlying distribution of the minority class, thereby significantly enhancing the generalization capability of the classifier. The Dirichlet distribution also ensures that the weights sum to one and are non-negative, promoting a balanced and realistic generation of synthetic instances. This method helps distribute synthetic samples more evenly across the feature space, avoiding the clustering problem seen in SMOTE.

Additionally, the Dirichlet distribution provides a probabilistic framework that naturally emphasizes closer neighbours while maintaining controlled randomness. This reduces the risk of overfitting to specific patterns in the minority class and mitigates the impact of outliers, which are assigned lower weights due to their larger distances. This also helps balance the bias–variance tradeoff, leading to synthetic instances that are more representative of the minority class distribution and ultimately improving the performance of classifiers on imbalanced datasets.

This method is ideal for datasets where diversity among synthetic samples is crucial and the minority class has a complex, variable structure. It is particularly useful when overfitting needs to be mitigated. However, the Dirichlet ExtSMOTE approach may struggle with extremely sparse datasets where the minority class instances are very few and far apart. In such cases, the variability in the weights might not be sufficient to generate meaningful synthetic samples, and the overall benefit of the method may be reduced.

6.2.3. FCRP SMOTE

Using the finite Chinese Restaurant Process (FCRP) in FCRP SMOTE offers a sophisticated approach to generating synthetic samples by introducing adaptive weighting through iterative probability adjustments. Initially, neighbours are selected based on inverse distance probabilities, which are dynamically adjusted with an alpha value over multiple iterations. This method respects the natural clustering of the minority class, enhances the diversity of synthetic samples, and better captures the true distribution and local density variations of the minority class while mitigating the influence of outliers. By iteratively refining the weights, the method balances the bias–variance tradeoff, leading to more representative synthetic instances and improved classifier performance on imbalanced datasets.

This method is well-suited for datasets where the minority class has varying densities and complex local structures. Its iterative adjustment mechanism is beneficial for applications requiring adaptive sampling that reflects the nuanced distribution of the minority class. However, this approach also has limitations. The iterative adjustment process can be computationally intensive, especially for large datasets, due to the repeated recalculations of probabilities. Additionally, in extremely sparse datasets where minority class instances are few and far apart, the clustering mechanism might struggle to form meaningful clusters, reducing the effectiveness of synthetic sample generation. Despite these challenges, FCRP SMOTE's adaptive mechanism provides significant advantages in generating high-quality synthetic samples.

6.2.4. BGMM SMOTE

Using BGMM SMOTE is advantageous because it leverages the probabilistic clustering capabilities of BGMMs to generate synthetic samples that closely reflect the true distribution of the minority class. By using mixing weights from the cluster where the minority median centroid belongs, the method ensures that synthetic instances are created in high-density regions of the minority class. This preserves the underlying structure and relationships among data points, leading to more accurate and diverse synthetic samples. This probabilistic framework also mitigates the impact of outliers, enhancing the classifier's ability to generalize from these synthetic instances.

This method is particularly suitable for datasets where the minority class can be effectively modelled using Gaussian mixtures, and where capturing the probabilistic distribution of the data is crucial. It is ideal for applications requiring detailed and accurate representation of the minority class distribution. However, the computational complexity of fitting a BGMM can be significant, particularly for large and high-dimensional datasets, making the process time-consuming. Additionally, in cases where minority class instances are sparse or do not form clear clusters, the model may struggle to capture the data distribution accurately.

In high-dimensional spaces, distance calculations can be computationally expensive and may not capture complex relationships. Using feature selection techniques beforehand can help mitigate the high-dimensionality issue.

The computations for this paper were performed using the resources provided by the Digital Research Alliance of Canada (Digital Research Alliance of Canada, 2024).

6.3. Future research directions

While our study introduces several promising methods, there are numerous avenues for future research that could further enhance these techniques and broaden their applications. Here are some suggestions:

- Extension to High-Dimensional Data: While our methods have shown promising results in bivariate scenarios, real-world applications often involve high-dimensional data. Future research should focus on adapting and optimizing these SMOTE extensions for high-dimensional datasets.
- Integration with Ensemble Learning Techniques: Combining the proposed SMOTE extensions with ensemble learning techniques, such as bagging, boosting, and stacking, could further improve classification performance. Future work could explore the synergy between these advanced sampling methods and ensemble classifiers to develop more robust and accurate models for imbalanced datasets.
- Ethical Considerations and Bias Mitigation: As highlighted in our discussion, the ethical implications of using synthetic data in sensitive applications need careful consideration. Future research should investigate techniques to identify and mitigate biases in synthetic data generation, ensuring that the models built on such data are fair and unbiased. This includes developing metrics and frameworks to evaluate the ethical impact of synthetic data on decision-making processes.

7. Conclusion

In summary, this manuscript serves as a foundational resource for comprehensively exploring solutions to enhance the applicability of SMOTE in the presence of abnormal instances within imbalanced datasets. By addressing this pivotal challenge, we aim to facilitate more robust and dependable classification in contexts where minority class instances hold paramount importance.

CRediT authorship contribution statement

Surani Matharaarachchi: Investigation, Methodology, Formal analysis, Data curation, Software, Validation, Visualization, Writing – original draft. **Mike Domaratzki:** Conceptualization, Supervision, Writing – review & editing. **Saman Muthukumarana:** Conceptualization, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Muthukumarana has been partially supported by research grants from the Natural Sciences and Engineering Research Council of Canada.

Appendix A. Assessing performance across different dataset shapes

To evaluate performance across various dataset shapes we used the ‘make_moons’ and ‘make_circles’ functions from the Python scikit-learn library to generate data (Pedregosa et al., 2011) synthetically. The visual representation of the resulting outcomes, inclusive of an additional outlier, is presented in Figs. A.1 and A.2, showcasing the final results for each dataset shape.

Appendix B. Simulation results for different accuracy measures

The box plots in Fig. B.1, Fig. B.2, and Fig. B.3 show the distribution of the F1 scores across 100 simulated datasets, each with two different imbalance ratios and an outlier ratio of 0.1. These F1 scores were computed for three different classifiers: Logistic Regression, k-Nearest Neighbors (k-NN), and Random Forest. These visual representations provide a clear comparison of the performance variability and central tendency of each classifier under the specified conditions.

Appendix C. Sensitivity analysis of the parameters

Sensitivity analysis was conducted to evaluate the impact of various parameter settings on the performance of the proposed methods. This provided insights into the stability of our methods, ensuring that they remain effective across a range of parameter values and dataset characteristics.

C.1. Parameters of Dirichlet ExtSMOTE

C.1.1. Impact of different multipliers (m)

Expected Values with Multipliers

Given a vector of parameters $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_k]$, the expected value for each weight w_i in the Dirichlet distribution $Dir(\alpha)$ is (Ng, Tang, & Tian, 2011):

$$E[w_i] = \frac{\alpha_i}{\sum_{j=1}^{k} \alpha_j}$$

Let the initial inverse distances be $d_1^{-1}, d_2^{-1}, \dots, d_k^{-1}$. We define the parameters of the Dirichlet distribution as:

$$\alpha_i = m \cdot d_i^{-1}$$

Where m is a multiplier.

Substituting $\alpha_i = m \cdot d_i^{-1}$ into the expected value formula:

$$E[w_i] = \frac{m \cdot d_i^{-1}}{\sum_{j=1}^{k} m \cdot d_j^{-1}}$$

Thus, the expected values $E[w_i]$ are independent of the multiplier m and depend only on the relative proportions of the inverse distances.

Expected Values with Variability and Concentration

The variance of each weight w_i in the Dirichlet distribution is given by Ng, Tang, and Tian (2011):

$$Var(w_i) = \frac{\alpha_i(\sum_{j=1}^{k} \alpha_j - \alpha_i)}{(\sum_{j=1}^{k} \alpha_j)^2(\sum_{j=1}^{k} \alpha_j + 1)}$$

Substituting $\alpha_i = m \cdot d_i^{-1}$ into the expected value formula:

$$\sum_{j=1}^{k} \alpha_j = m \cdot \sum_{j=1}^{k} m \cdot d_j^{-1}$$

$$Var(w_i) = \frac{m \cdot d_i^{-1}(m \cdot \sum_{j=1}^{k} d_j^{-1} - m \cdot d_i^{-1})}{(m \cdot \sum_{j=1}^{k} d_j^{-1})^2(m \cdot \sum_{j=1}^{k} d_j^{-1} + 1)}$$

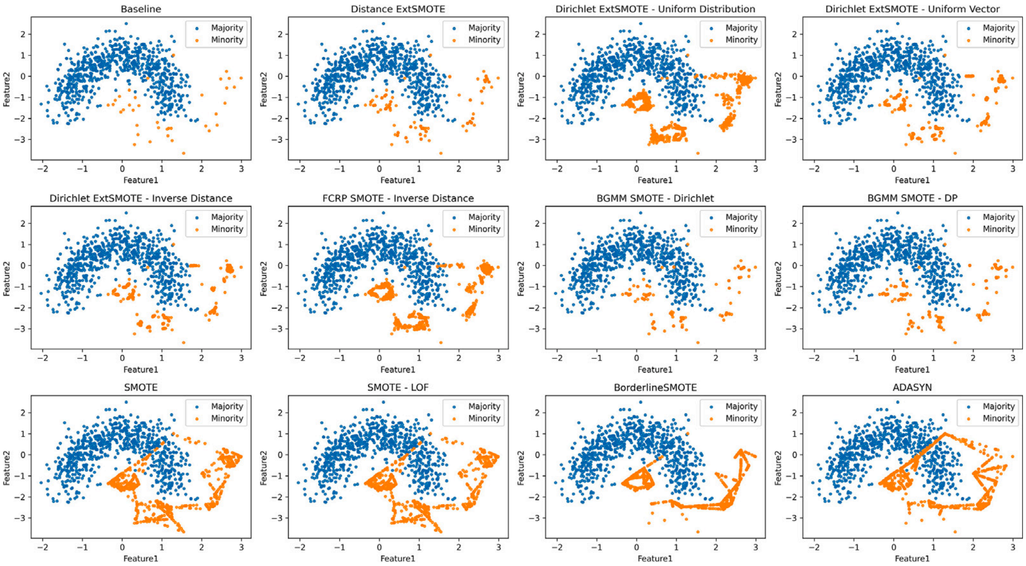


Fig. A.1. Results comparison with noisy moons dataset with one abnormal instance when noise = 0.2.

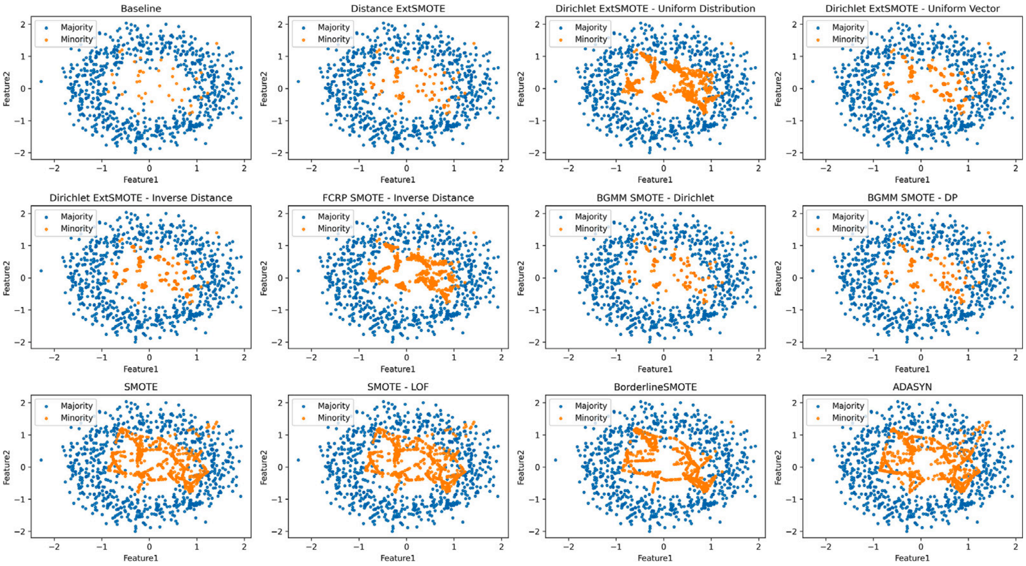


Fig. A.2. Results comparison with noisy circles dataset with one abnormal instance when noise = 0.2.

Simplifying:

$$Var(w_i) = \frac{m \cdot d_i^{-1} \cdot m(\sum_{j=1}^{k} d_j^{-1} - d_i^{-1})}{m^2(\sum_{j=1}^{k} d_j^{-1})^2(m \cdot \sum_{j=1}^{k} d_j^{-1} + 1)}$$

$$= \frac{m^2 \cdot d_i^{-1}(\sum_{j=1}^{k} d_j^{-1} - d_i^{-1})}{m^2(\sum_{j=1}^{k} d_j^{-1})^2(m \cdot \sum_{j=1}^{k} d_j^{-1} + 1)}$$

$$= \frac{d_i^{-1}(\sum_{j=1}^{k} d_j^{-1} - d_i^{-1})}{(\sum_{j=1}^{k} d_j^{-1})^2(m \cdot \sum_{j=1}^{k} d_j^{-1} + 1)}$$

From this expression, it is clear that as m increases, the term $(m \cdot \sum_{j=1}^{k} d_j^{-1} + 1)$ also increases, thereby reducing the overall variance. Conversely, for small values of m, this term does not significantly increase, leading to higher variance.

The expected values of the weights in the Dirichlet distribution remain constant across different multipliers because they depend on the relative proportions of the parameters. However, the variability of the weights differs significantly. Higher multipliers (e.g., 100) result in lower variability, leading to more stable and predictable synthetic samples. In contrast, lower multipliers (e.g., 0.01) introduce higher variability, producing a more diverse but less stable set of synthetic samples. This mathematical analysis underscores the importance

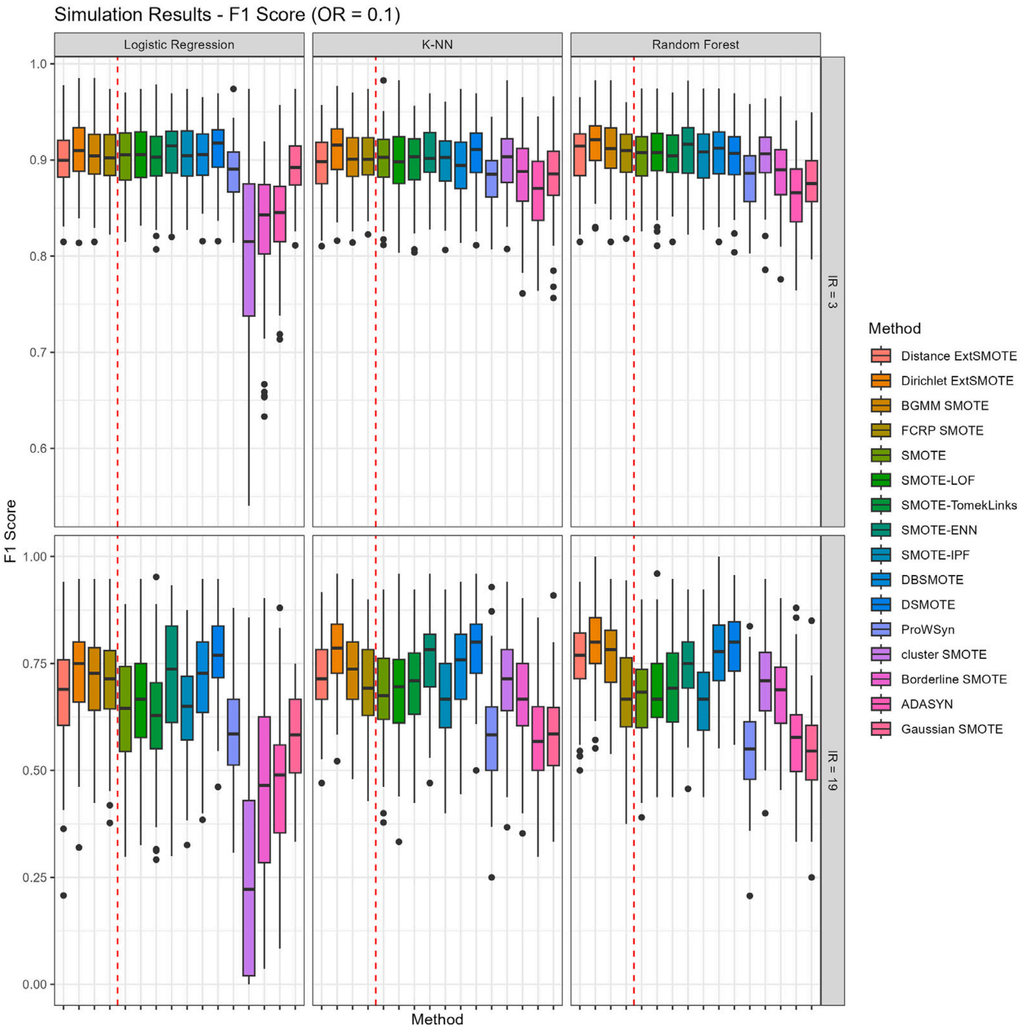


Fig. B.1. F1 Scores across 100 simulated datasets with two different imbalance ratios and an outlier ratio of 0.1 were computed for three different classifiers. On the left-hand side of the dashed line are the results obtained from the proposed methods, while on the right-hand side are the results from existing methods.

of choosing an appropriate multiplier based on the desired balance between diversity and stability in synthetic sample generation for addressing class imbalance.

Fig. C.1 and Fig. C.2 present the mean F1 scores for different multipliers of Dirichlet ExtSMOTE, considering outlier ratios of 0.05 and 0.1 respectively, across two different imbalance ratios for each type. It is important to note that the F1 scores vary significantly based on the selection of the classifier, the multiplier, and the number of outliers in the dataset, especially in scenarios with highly imbalanced data. This underscores the critical need for hyperparameter optimization to achieve higher performance. The figures illustrate how the careful tuning of parameters can lead to significant improvements in classifier effectiveness, demonstrating the necessity of customized approaches in handling imbalanced datasets with varying outlier ratios.

C.1.2. Impact of different types

Three different types are introduced in the Dirichlet ExtSMOTE method: Inverse Distance (D), Uniform Distribution (UD), and Uniform Vector (UV). Fig. C.3 illustrates the changes in Mean F1 Scores for these different types under two varying outlier ratios and two different imbalance ratios.

C.2. Parameters of FCRP SMOTE

C.2.1. Impact of different α values

We evaluated the performance changes of FCRP SMOTE as the parameter alpha varied from 0.1 to 1. Fig. C.4 shows the mean F1 scores for two imbalance rates and outlier ratios using three classifiers, highlighting the impact of alpha on the effectiveness of FCRP SMOTE under different conditions.

C.3. Parameters of BGMM SMOTE

C.3.1. Impact of different priors

Two priors are introduced with BGMM SMOTE. Fig. C.5 shows the mean F1 scores for two imbalance rates and outlier ratios using three classifiers, highlighting the impact of selecting the prior on the effectiveness of BGMM SMOTE under different conditions.

Appendix D. PCA plots of outliers detected by LOF

The Fig. D.1, Fig. D.2, Fig. D.3, and Fig. D.4 show the Principal Component Analysis (PCA) plots for the first two components, highlighting

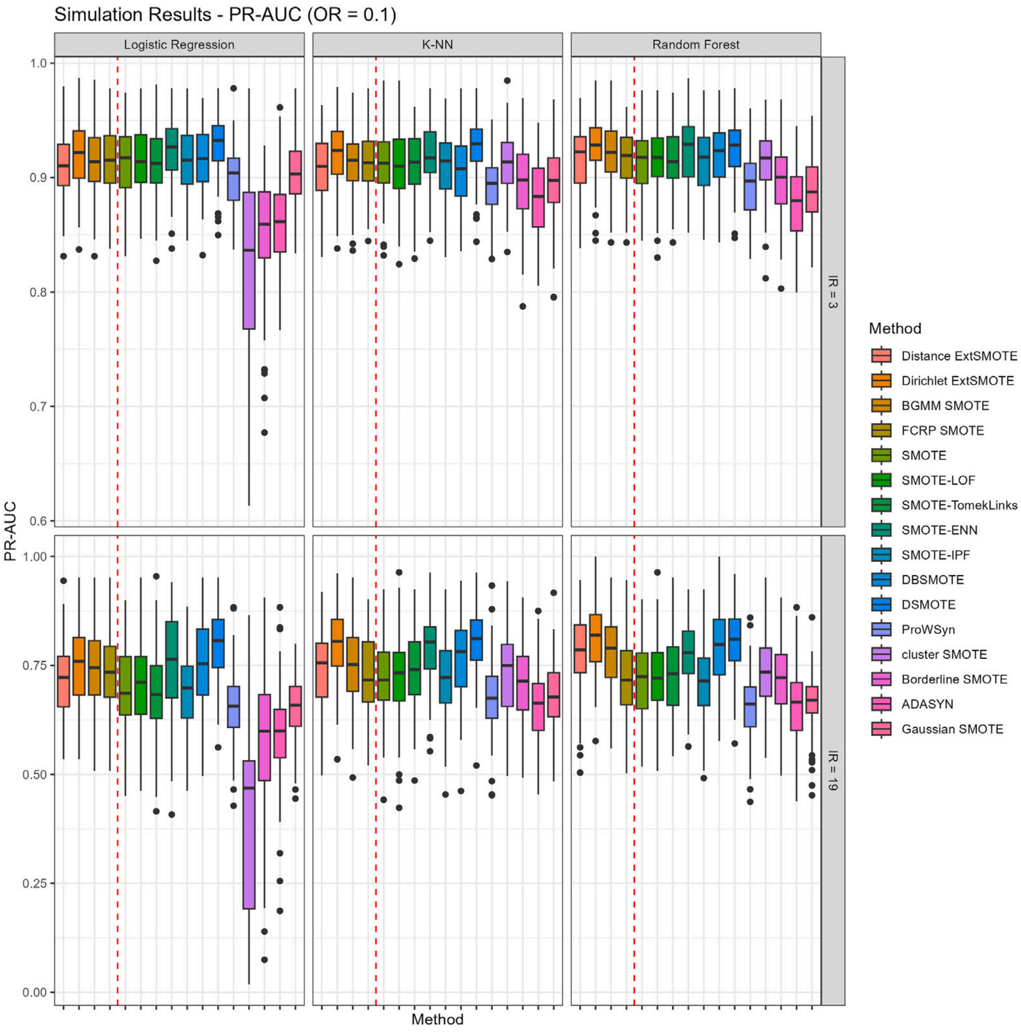


Fig. B.2. PR-AUCs across 100 simulated datasets with two different imbalance ratios and an outlier ratio of 0.1 were computed for three different classifiers. On the left-hand side of the dashed line are the results obtained from the proposed methods, while on the right-hand side are the results from existing methods.

the distribution of outliers within the minority class across 25 real-world datasets analysed in the experimental study. The outliers were detected by the Local Outlier Factor (LOF) method for each application dataset used in the analysis.

Appendix E. Comprehensive evaluation with additional accuracy measures

In addition to evaluating our methods based solely on the F1 score, we conducted a comprehensive assessment using a variety of performance metrics to ensure a robust evaluation of our classifiers. These metrics included recall, precision, PR AUC (Precision-Recall Area Under Curve), and MCC (Matthews Correlation Coefficient). By incorporating these additional measures, we aimed to capture different aspects of classifier performance, providing a more holistic understanding of their effectiveness.

Tables E.1, E.2, E.3, E.4, E.5, and E.6 present the detailed results for PR AUC and MCC for each of the three classifiers: Logistic Regression, k-Nearest Neighbors (k-NN), and Random Forest. These tables illustrate how each classifier performs under different conditions, allowing for a nuanced comparison of their capabilities. By analysing these metrics, we demonstrate the strengths and limitations of each classifier and the effectiveness of our methods across various scenarios.

E.1. MCC

Tables E.1–E.3.

E.2. PR-AUC

Tables E.4–E.6.

Data availability

Data used in this manuscript can be accessed through the UCI Machine Learning Repository at http://archive.ics.uci.edu/ml.

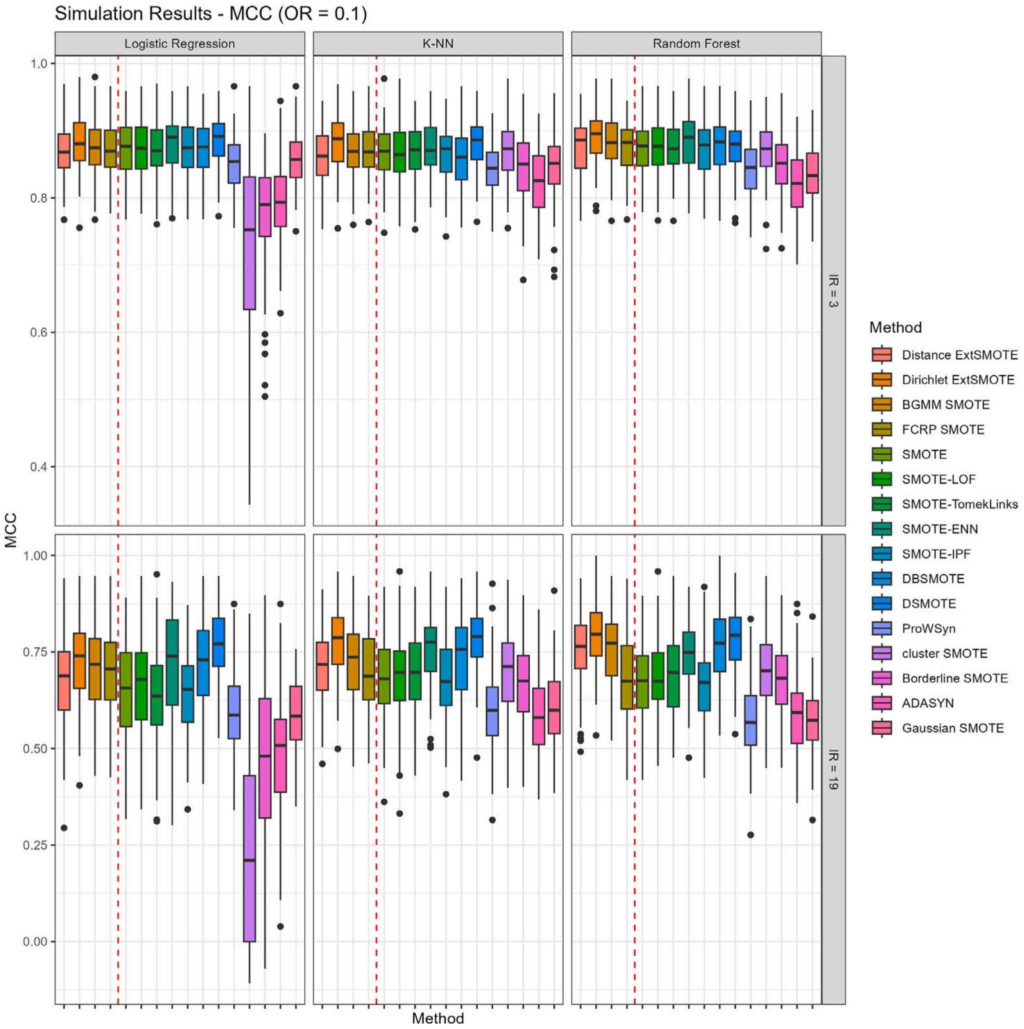


Fig. B.3. MCCs across 100 simulated datasets with two different imbalance ratios and an outlier ratio of 0.1 were computed for three different classifiers. On the left-hand side of the dashed line are the results obtained from the proposed methods, while on the right-hand side are the results from existing methods.

Table E.1
MCC results of 17 comparative algorithms on 25 imbalanced datasets using 100 × 5 fold cross-validation with a Logistic Regression classifier.

Dataset	Distance ExtSMOTE	Dirichlet ExtSMOTE	BGMM SMOTE	FCRP SMOTE	SMOTE	SMOTE LOF	SMOTE TomekLinks	SMOTE ENN	SMOTE IPF	DBSMOTE	ProWSyn	DSMOTE	cluster SMOTE	Gaussian SMOTE	Borderline SMOTE	ADASYN	Baseline
yeast6	0.4204	0.4229	**0.4547**	0.4075	0.3637	0.3646	0.3646	0.4434	0.3653	0.4379	0.3582	0.3265	0.3079	0.3294	0.4417	0.3248	0.4434
yeast5	0.4503	0.4462	0.465	0.4354	0.4161	0.4137	0.4124	0.5051	0.4112	**0.5508**	0.3978	0.4441	0.3688	0.379	0.4663	0.3652	0.5051
yeast-1289vs7	0.2013	0.2443	0.2301	0.1977	0.2054	0.2053	0.1965	0.2474	0.1949	0.0745	0.1939	0.206	**0.2522**	0.2329	0.2398	0.1955	0.2474
yeast4	0.2727	0.3136	0.3269	0.2927	0.2929	0.2948	0.2947	0.299	0.2976	**0.3846**	0.2987	0.2482	0.2903	0.3101	0.3618	0.2746	0.299
yeast-2vs8	0.2375	0.3891	0.2769	0.2636	0.4682	0.4669	0.4455	0.6263	0.4433	0.2867	0.4961	**0.6363**	0.6082	0.4954	0.2087	0.1447	0.6263
glass12357vs6	0.5223	0.6464	0.5718	0.5924	0.6321	0.6321	0.6321	0.4875	0.6321	**0.6806**	0.6321	0.4346	0.6268	0.4896	0.6321	0.6321	0.4875
yeast-1458vs7	0.1446	**0.1982**	0.1516	0.1732	0.1661	0.1637	0.1602	0	0.1576	0.0904	0.1302	0.1456	0.1233	0.1532	0.173	0.1573	0
oil	0.5422	0.5416	0.4974	0.5027	0.4975	0.4978	0.5033	**0.5534**	0.5023	0.5102	0.5132	0.4444	0.4897	0.415	0.4642	0.4966	**0.5534**
abalone9_18	0.4987	0.5005	0.4748	0.4798	0.4394	0.4417	0.4549	0.4929	0.4549	0.5189	0.4838	0.5067	0.471	0.364	**0.5277**	0.4108	0.4929
glass12367vs5	0.4423	0.5437	0.5447	0.4806	0.5267	0.5259	0.519	0.3422	0.5208	0.1989	0.5458	0.3158	0.408	0.4879	**0.563**	0.5375	0.3422
thyroid_sick	0.5797	0.5861	0.5583	0.5715	0.5571	0.5583	0.5548	**0.6935**	0.5542	0.6927	0.5838	0.5754	0.6225	0.3766	0.5373	0.5298	**0.6935**
yeast-1vs7	0.3385	0.3313	**0.3586**	0.3194	0.2887	0.2897	0.2928	0.1763	0.2941	0.2171	0.287	0.3135	0.3137	0.3303	0.3051	0.293	0.1763
us_crime	0.4564	0.4673	0.4689	0.47	0.4623	0.4607	0.4571	0.4744	0.4589	0.4615	0.4778	0.4972	0.4757	0.4706	**0.4988**	0.448	0.4744
glass12vs5	0.6241	0.785	0.687	0.7652	0.776	0.7743	0.7706	0.7162	0.7705	0.4701	0.7874	0.6046	0.7459	0.654	0.7329	**0.791**	0.7162
spectrometer	0.8451	0.8479	0.8513	0.846	0.8361	0.8336	0.8381	0.8534	0.8379	0.8375	0.8467	0.8284	0.8485	0.7818	0.8394	0.8209	**0.8607**
landsat_satellite	0.2371	0.245	0.2467	0.2485	**0.2661**	0.2654	0.2646	0.0945	0.2655	0.0592	0.261	0.0656	0.2568	0.257	0.2604	0.2579	0.0945
mfeatmor0	0.9916	0.9924	0.9916	0.9916	0.9908	0.9915	0.9901	0.99	0.9905	0.9916	**0.9939**	0.9709	0.9858	0.9741	0.9916	0.9889	0.9916
yeast3	0.6754	0.6713	0.6757	0.6658	0.6231	0.6244	0.6251	**0.7293**	0.6251	0.6789	0.6411	0.7031	0.6225	0.6304	0.6244	0.6183	**0.7293**
mfeatmor01	0.8966	0.8969	0.8917	0.8947	0.8853	0.8898	0.89	0.9053	0.8854	0.9049	0.8844	0.8855	0.8833	0.8291	0.797	0.7088	**0.9084**
glass123vs567	0.7872	0.8166	**0.8267**	0.8171	0.7965	0.7948	0.7973	0.737	0.7982	0.7993	0.7829	0.7978	0.7961	0.7807	0.7842	0.7892	0.7329
parkinsons	0.5496	0.5747	0.5587	0.557	0.5444	0.5469	0.5431	**0.6491**	0.5462	0.5895	0.5266	0.6076	0.5427	0.4987	0.5567	0.5584	**0.6491**
habermans_survival	0.262	0.3154	0.2858	0.2873	0.3012	0.304	0.301	0.2233	0.3061	0.2756	0.3109	0.145	**0.3241**	0.2923	0.2791	0.3111	0.2233
glass23567vs1	0.4348	0.454	0.4398	0.435	0.4203	0.4217	0.4267	**0.4957**	0.4179	0.4172	0.4348	0.4879	0.4496	0.4546	0.4507	0.4458	**0.4957**
breast_cancer	0.9462	0.9485	0.9493	0.9467	0.9438	0.9435	0.9422	**0.9496**	0.9444	0.9471	0.9408	0.8996	0.9431	0.9431	0.9223	0.9211	0.9484
banknote	0.9799	**0.9826**	0.9825	0.9815	0.9809	0.9808	0.981	0.981	0.9808	0.9805	0.9812	0.903	0.9811	0.95	0.981	NA	0.981

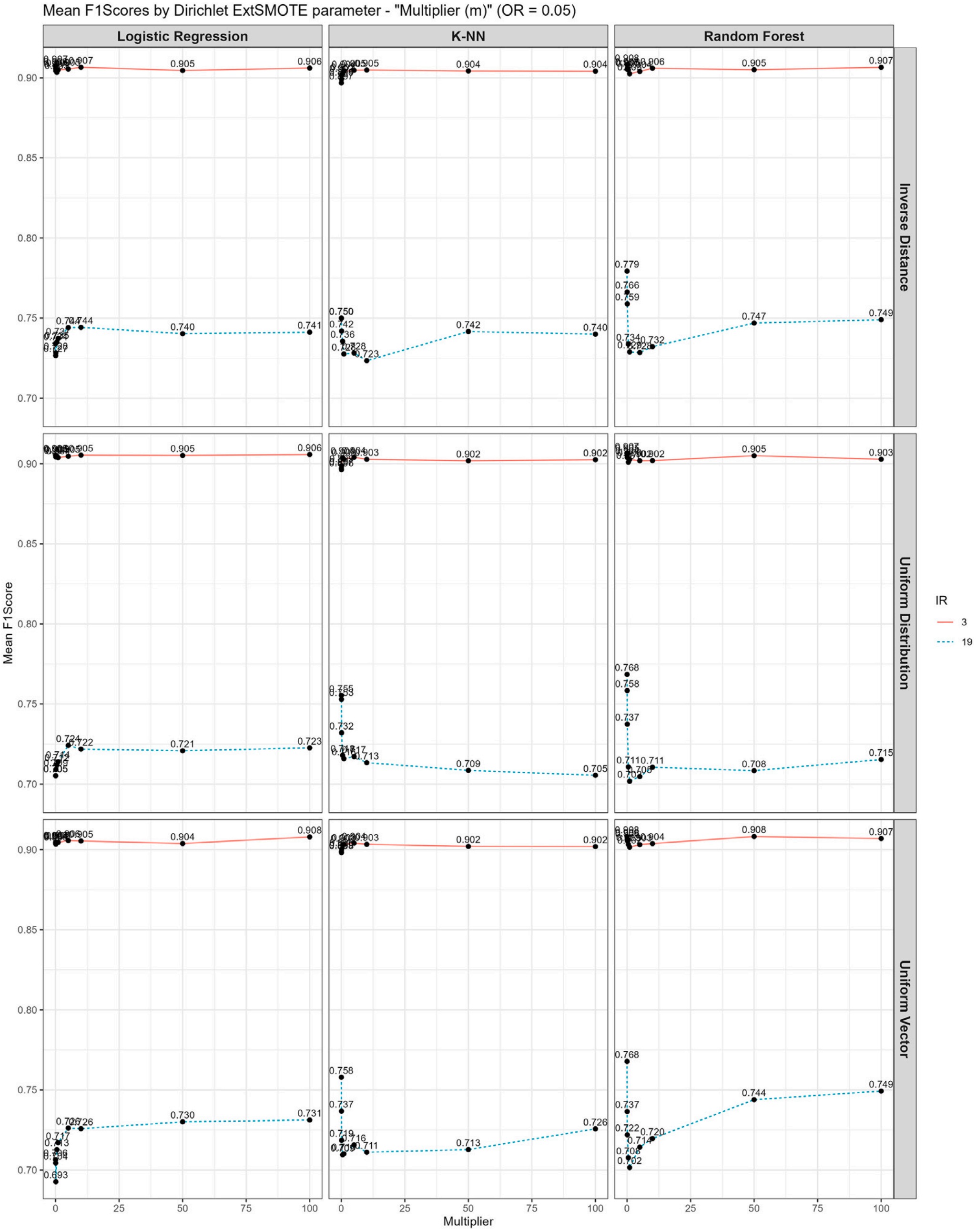


Fig. C.1. Mean F1 Scores for different multipliers of Dirichlet ExtSMOTE for an outlier ratio of 0.05 and two different imbalance ratios for each type.

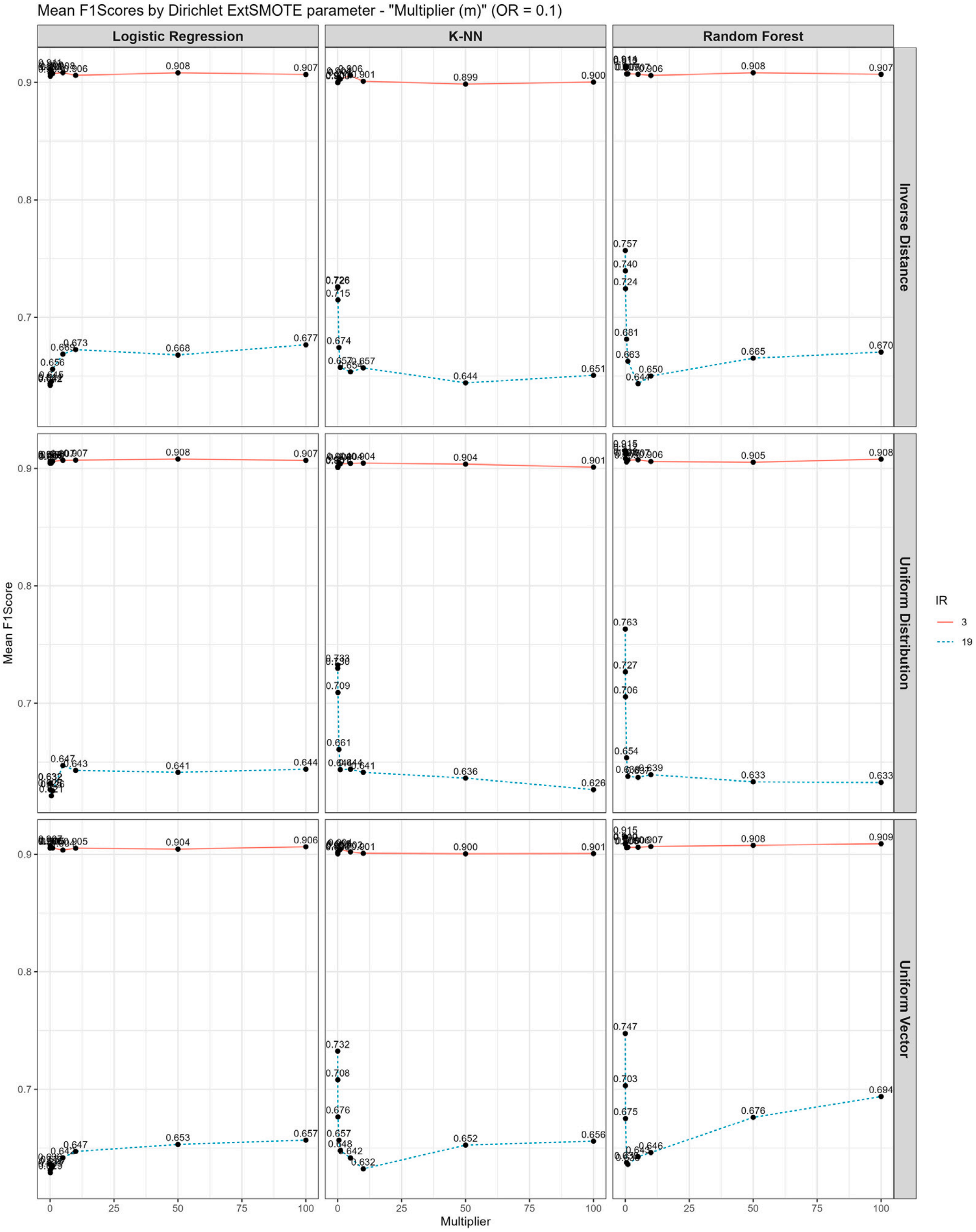


Fig. C.2. Mean F1 Scores for different multipliers of Dirichlet ExtSMOTE for an outlier ratio of 0.1 and two different imbalance ratios for each type.

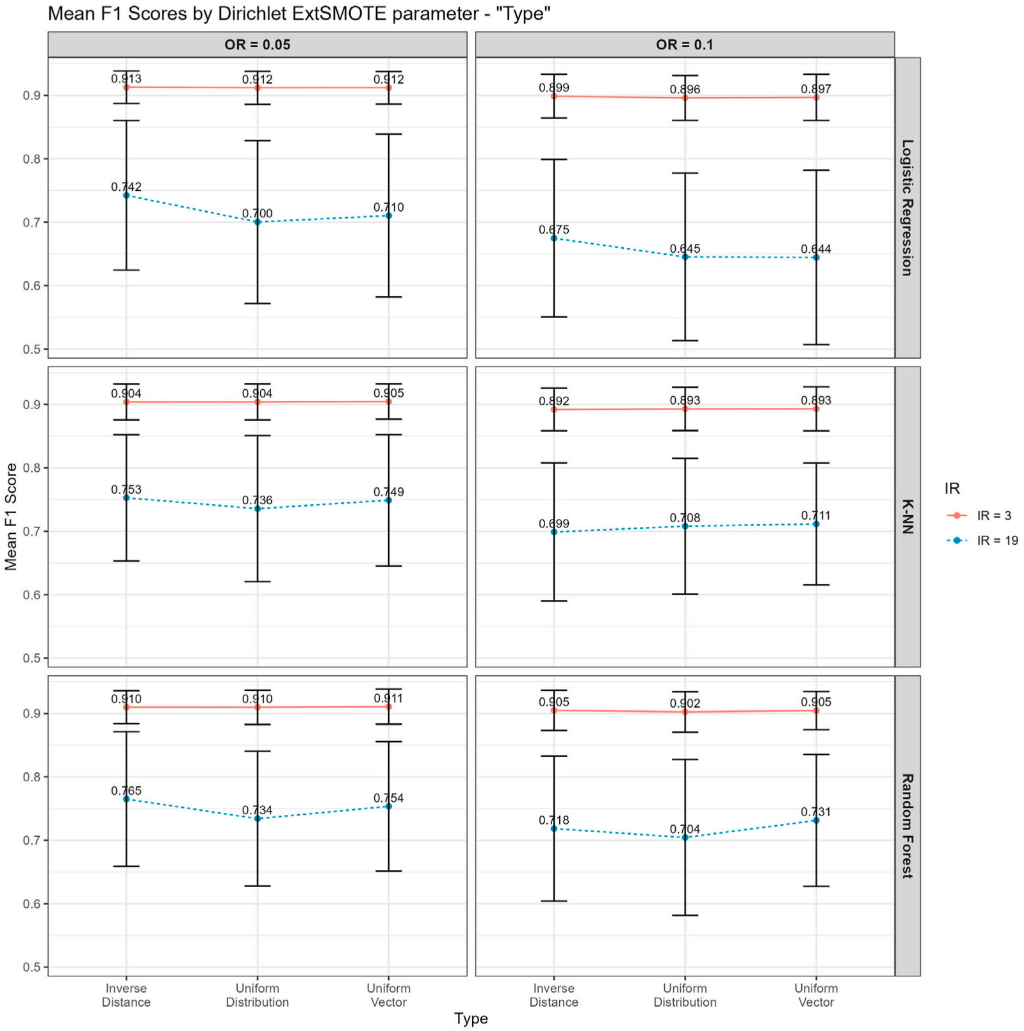


Fig. C.3. Mean F1 Scores for different types of Dirichlet ExtSMOTE for two different outlier ratios and two different imbalance ratios.

Table E.2
MCC results of 17 comparative algorithms on 25 imbalanced datasets using 100 × 5 fold cross-validation with a K-NN classifier.

Dataset	Distance ExtSMOTE	Dirichlet ExtSMOTE	BGMM SMOTE	FCRP SMOTE	SMOTE	SMOTE LOF	SMOTE TomekLinks	SMOTE ENN	SMOTE IPF	DBSMOTE	ProWSyn	DSMOTE	cluster SMOTE	Gaussian SMOTE	Borderline SMOTE	ADASYN	Baseline
yeast6	0.4748	0.506	**0.5154**	0.4983	0.4	0.4	0.3952	0.499	0.3979	0.442	0.3741	0.4653	0.4022	0.3962	0.3947	0.3571	0.499
yeast5	0.512	0.5941	0.549	0.57	0.4954	0.4924	0.4868	**0.6553**	0.4905	0.6444	0.4487	0.6018	0.5092	0.4226	0.5322	0.4464	**0.6553**
yeast-1289vs7	0.111	0.2443	0.2216	0.2048	0.2224	0.2279	0.2225	0.2337	0.2251	0.11	0.2074	**0.3196**	0.2506	0.221	0.2172	0.2207	0.2337
yeast4	0.3298	**0.3957**	0.3675	0.3705	0.334	0.3328	0.3455	0.3066	0.3398	0.3087	0.3304	0.363	0.3296	0.3689	0.3431	0.3272	0.3066
yeast-2vs8	0.5133	0.5437	0.4461	0.4333	0.4019	0.4019	0.3746	**0.6461**	0.3799	0.6009	0.4167	0.6287	0.4332	0.4122	0.3266	0.3644	**0.6461**
glass12357vs6	**0.8174**	0.7875	0.6857	0.7665	0.7864	0.7897	0.7945	0.7571	0.7955	0.7571	0.7352	0.6961	0.7462	0.5423	0.7862	0.7859	0.7571
yeast-1458vs7	0.0044	**0.1616**	0.0069	0.0444	0.0982	0.0985	0.1007	0.0162	0.1055	0.1257	0.1277	0.1354	0.1132	0.0761	0.1024	0.0899	0.0162
oil	0.492	0.5941	0.551	0.5492	0.5559	0.5586	0.5583	0.5435	0.5594	0.5917	0.5561	0.3559	0.5431	0.4199	**0.6071**	0.5478	0.5435
abalone9_18	0.3351	0.4082	**0.4286**	0.4273	0.3791	0.378	0.3825	0.4267	0.3831	0.4058	0.3823	0.4239	0.3404	0.3716	0.4025	0.3533	0.4267
glass12367vs5	0.6008	0.6697	0.6234	0.6628	0.6548	0.6624	0.6739	0.4325	**0.6836**	0.2077	0.6465	0.3754	0.562	0.5671	0.6442	0.6261	0.4325
thyroid_sick	0.5266	0.5422	0.5458	0.5464	0.5561	0.5537	0.5514	0.5378	0.5554	0.4877	0.5192	0.487	0.5458	0.489	0.538	**0.5566**	0.5378
yeast-1vs7	0.303	0.392	0.3067	0.3512	0.3773	0.3753	0.3652	**0.4295**	0.3663	0.2494	0.3277	0.3994	0.4096	0.3211	0.4094	0.3529	**0.4295**
us_crime	0.3367	0.3667	0.3649	0.3711	0.3583	0.3579	0.3613	0.3951	0.3565	0.3051	0.3941	**0.3997**	0.3675	0.3673	0.3607	0.3453	0.3951
glass12vs5	0.2675	0.5669	0.5185	0.5487	0.6376	0.6218	0.6553	0.293	0.6513	0.2017	0.6474	0.2765	0.6	**0.6847**	0.6416	0.6435	0.293
spectrometer	0.7894	**0.8678**	0.7576	0.8235	0.8332	0.8384	0.8401	0.8273	0.8374	0.8071	0.8135	0.7958	0.8185	0.8667	0.789	0.8134	0.8388
landsat_satellite	0.6246	**0.6736**	0.6314	0.6549	0.6702	0.6702	0.6663	0.6723	0.6674	0.6703	0.6531	0.4775	0.6614	0.5948	0.6559	0.6667	0.6723
mfeatmor0	0.9916	**0.9926**	0.9916	0.9916	0.9832	0.9877	0.9846	0.9889	0.9855	0.9916	0.9874	0.9462	0.9864	0.9611	0.9916	0.9862	0.9916
yeast3	0.6914	0.7123	0.7216	0.7161	0.685	0.685	0.6904	**0.7322**	0.688	0.66	0.7308	0.7191	0.6572	0.6851	0.5983	0.6508	**0.7322**
mfeatmor01	0.9367	0.9383	0.9389	0.9352	0.9321	0.9319	0.9332	0.939	0.933	0.9222	0.932	0.9223	0.9277	0.928	0.8995	0.8603	**0.9453**
glass123vs567	0.8258	**0.84**	0.8355	0.8363	0.8118	0.8103	0.8139	0.8165	0.8099	0.8095	0.8331	0.7882	0.8045	0.8329	0.8017	0.8024	0.8189
parkinsons	0.8288	**0.8291**	0.8071	0.821	0.7827	0.7847	0.7825	0.7039	0.7897	0.6765	0.7437	0.6935	0.79	0.7914	0.8068	0.8255	0.7039
habermans_survival	0.0453	0.1874	0.1529	0.149	0.1839	0.1827	0.1704	−0.003	0.1854	0.0551	0.2153	0.1122	0.1753	**0.2307**	0.1848	0.178	−0.003
glass23567vs1	0.603	**0.6387**	0.6167	0.6148	0.5771	0.5764	0.5776	0.5419	0.5755	0.5309	0.5351	0.531	0.6139	0.5251	0.5875	0.5954	0.5419
breast_cancer	0.9175	**0.9286**	0.9211	0.9202	0.9174	0.9186	0.9145	0.9128	0.9186	0.9112	0.9146	0.8842	0.9149	0.9124	0.9038	0.9042	0.9125
banknote	0.9971	**0.9976**	0.9971	0.9971	0.9971	0.9971	0.9971	0.9971	0.9971	0.9971	0.9967	0.9126	0.9971	0.9747	0.9971	NA	0.9971

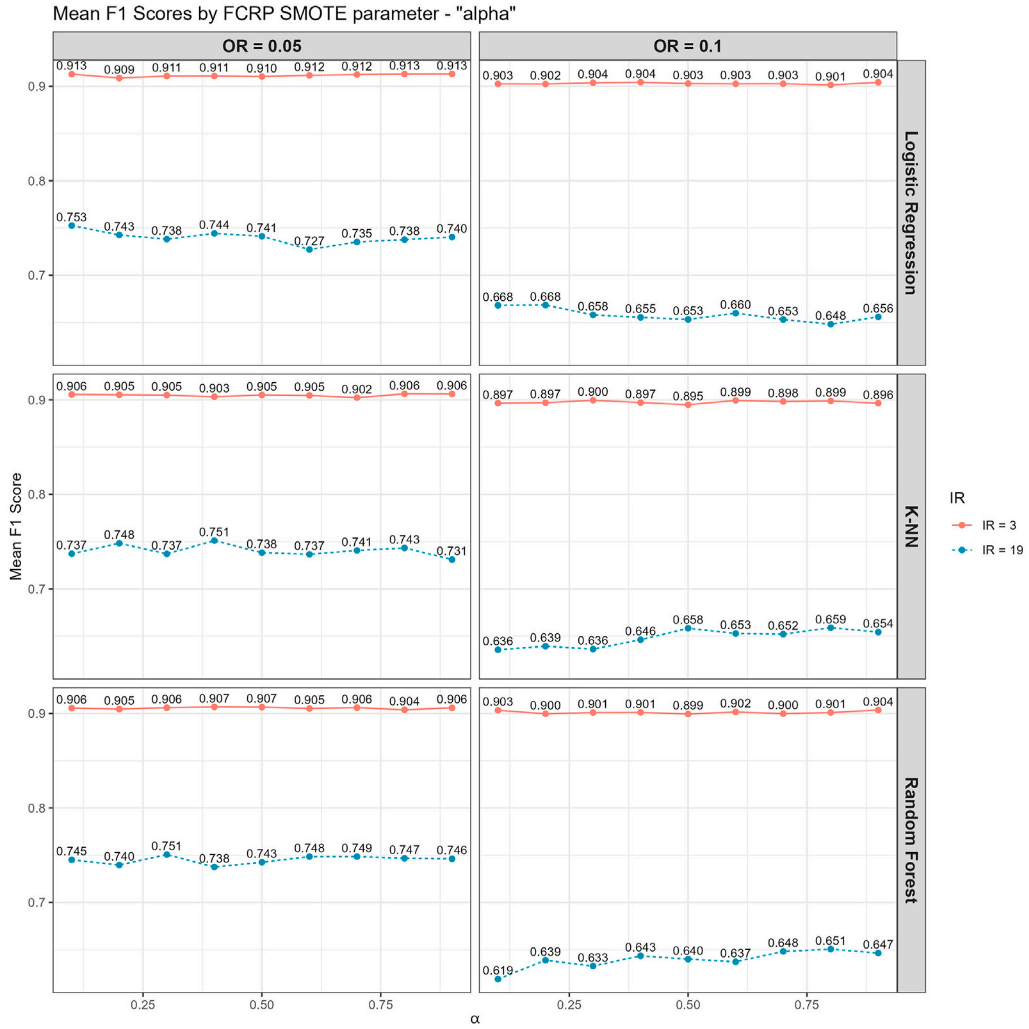


Fig. C.4. Mean F1 Scores for different alpha values of FCRP SMOTE for two different outlier ratios and two different imbalance ratios.

Table E.3
MCC results of 17 comparative algorithms on 25 imbalanced datasets using 100 × 5 fold cross-validation with a Random Forest classifier.

Dataset	Distance ExtSMOTE	Dirichlet ExtSMOTE	BGMM SMOTE	FCRP SMOTE	SMOTE	SMOTE LOF	SMOTE TomekLinks	SMOTE ENN	SMOTE IPF	DBSMOTE	ProWSyn	DSMOTE	cluster SMOTE	Gaussian SMOTE	Borderline SMOTE	ADASYN	Baseline
yeast6	0.5556	0.6105	**0.63**	0.6039	0.5424	0.5487	0.5426	0.5173	0.5485	0.5212	0.4931	0.5172	0.5608	0.5364	0.5231	0.5281	0.5157
yeast5	0.6115	0.6269	0.6228	0.6273	0.5235	0.5273	0.5194	0.5986	0.5144	0.5904	0.5131	**0.6549**	0.5472	0.6157	0.5408	0.5068	0.5988
yeast-1289vs7	0.0584	0.1397	0.0719	0.097	0.1534	0.1504	0.1599	0.2947	0.1505	0.1165	0.1292	0.3183	0.2688	**0.3463**	0.2162	0.119	0.2924
yeast4	0.2814	0.3899	0.3638	0.366	0.392	0.3896	0.3779	0.3551	0.3729	0.3637	0.3808	0.3413	0.4245	0.3983	**0.4299**	0.3814	0.359
yeast-2vs8	0.5919	0.6001	0.5337	0.4813	0.4381	0.4372	0.4298	0.6131	0.429	0.6306	0.5114	0.6227	0.5445	**0.659**	0.3427	0.2806	0.6139
glass12357vs6	0.7149	**0.7347**	0.6629	0.6485	0.6165	0.6222	0.616	0.5771	0.609	0.6192	0.6369	0.5408	0.6209	0.6143	0.6219	0.5968	0.5826
yeast-1458vs7	−0.0037	0.1539	0.0094	0.0573	0.0987	0.0924	0.0924	−0.0032	0.0841	−0.0117	0.0937	**0.1614**	0.0727	0.0327	0.1103	0.0889	0.0005
oil	0.4578	0.5413	0.4939	0.5658	0.5014	0.5066	0.5128	0.4817	0.5136	0.3278	**0.5916**	0.4637	0.4484	0.5656	0.5372	0.4928	0.4819
abalone9_18	0.3663	0.3945	0.3683	0.3724	0.3395	0.3387	0.3454	0.3763	0.3436	**0.4026**	0.3681	0.3966	0.3452	0.3482	0.3739	0.3219	0.3821
glass12367vs5	0.659	**0.8418**	0.6662	0.7232	0.7437	0.7417	0.7552	0.6031	0.7555	0.7286	0.7788	0.6221	0.682	0.7265	0.7619	0.7412	0.6024
thyroid_sick	0.8521	**0.8769**	0.8625	0.8515	0.8673	0.8626	0.8646	0.853	0.8652	0.85	0.8402	0.8061	0.871	0.8631	0.8609	0.8633	0.8521
yeast-1vs7	0.1528	0.3019	0.1863	0.2536	0.322	0.3199	0.303	0.3373	0.3171	0.1233	0.2151	**0.3931**	0.3295	0.3575	0.3265	0.2724	0.3373
us_crime	0.4449	0.4574	0.4364	0.4577	0.462	0.4618	0.4588	0.4768	0.4621	0.3822	0.4815	0.4711	0.4538	**0.4851**	0.4611	0.451	0.4799
glass12vs5	0.671	**0.8069**	0.7643	0.7802	0.7811	0.7884	0.7845	0.692	0.7824	0.7485	0.7843	0.5826	0.7265	0.7773	0.7821	0.7786	0.6922
spectrometer	0.8188	**0.8466**	0.798	0.8201	0.8003	0.8094	0.8018	0.8314	0.8002	0.8306	0.7939	0.8062	0.8013	0.8102	0.8011	0.7839	0.825
landsat_satellite	0.6254	0.6456	0.6401	0.6394	0.6529	0.6483	0.6503	0.6454	0.6505	0.6441	0.6456	0.6262	**0.6572**	0.5997	0.6511	0.6523	0.6459
mfeatmor0	0.9916	**0.9918**	0.9915	0.9916	0.9878	0.9905	0.9875	0.988	0.9871	0.9901	0.9903	0.9498	0.9864	0.9867	0.9916	0.9858	0.9916
yeast3	0.7402	**0.7618**	0.7522	0.757	0.7584	0.7599	0.7592	0.7339	0.7599	0.7429	0.7617	0.731	0.7574	0.7424	0.7516	0.7496	0.7352
mfeatmor01	0.9413	0.9477	0.9453	0.9406	0.9384	0.9393	0.938	0.9419	0.9379	0.9436	0.9463	0.9241	0.9358	0.9426	0.9154	0.8663	**0.9481**
glass123vs567	0.86	**0.8762**	0.8598	0.8707	0.856	0.8559	0.854	0.8548	0.8627	0.8559	0.8659	0.7772	0.8549	0.8586	0.8415	0.8387	0.8542
parkinsons	0.7022	0.7584	0.7391	0.7437	0.7538	0.7564	0.7564	0.7372	0.7576	0.7434	0.7239	0.669	**0.7608**	0.7277	0.7427	0.7585	0.7369
haber-mans_survival	0.1934	0.2194	0.1988	0.2272	0.1767	0.1728	0.1933	0.137	0.1928	0.1333	0.2234	0.1244	0.181	**0.2525**	0.1649	0.1698	0.1354
glass23567vs1	0.7274	**0.7765**	0.747	0.7408	0.7343	0.7349	0.7307	0.7372	0.7284	0.75	0.6901	0.736	0.7322	0.6838	0.7263	0.7228	0.7437
breast_cancer	0.9172	**0.9246**	0.9183	0.92	0.9163	0.9166	0.9186	0.9132	0.9166	0.9169	0.9148	0.8856	0.917	0.9147	0.9115	0.9206	0.9136
banknote	0.9873	**0.9891**	0.9873	0.9883	0.9867	0.9864	0.9866	0.986	0.9868	0.9864	0.9864	0.8965	0.9867	0.9822	0.986	NA	0.9861

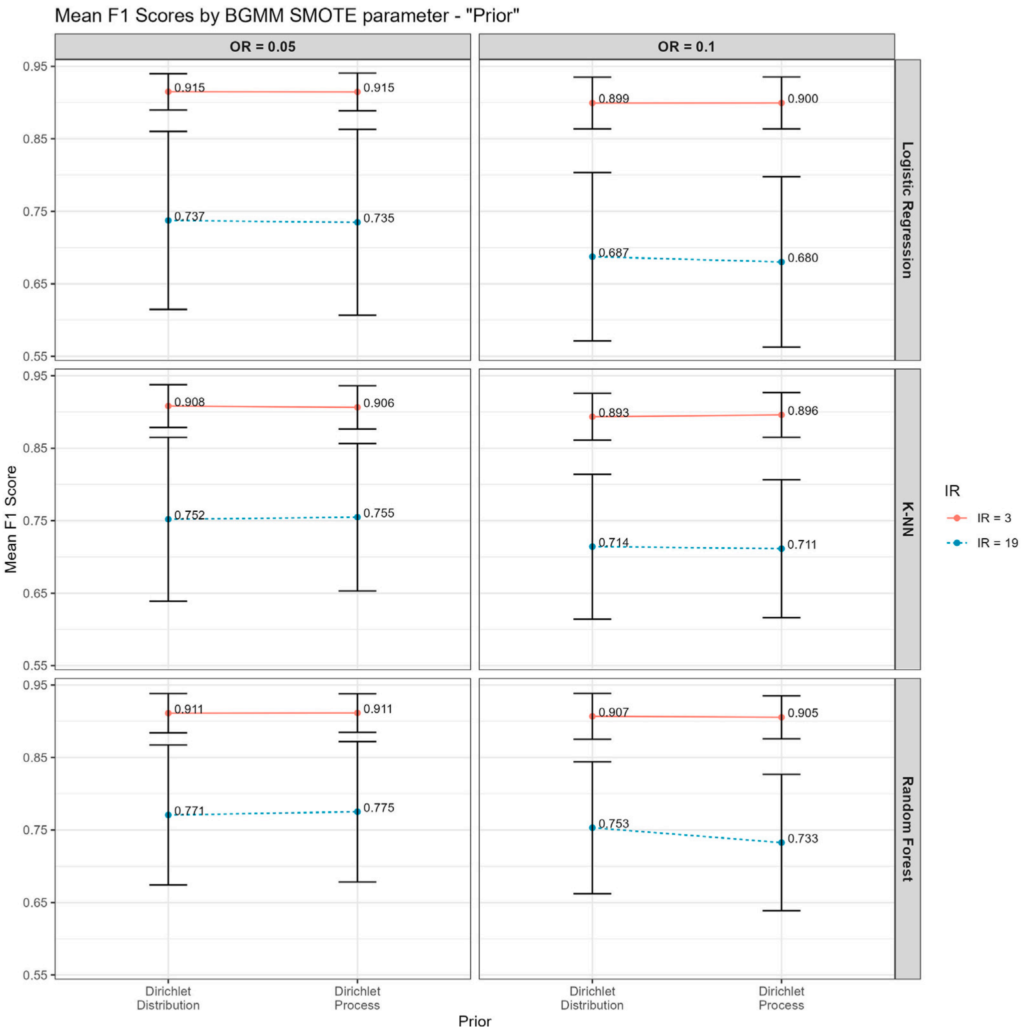


Fig. C.5. Mean F1 Scores for different priors of BGMM SMOTE for two different outlier ratios and two different imbalance ratios.

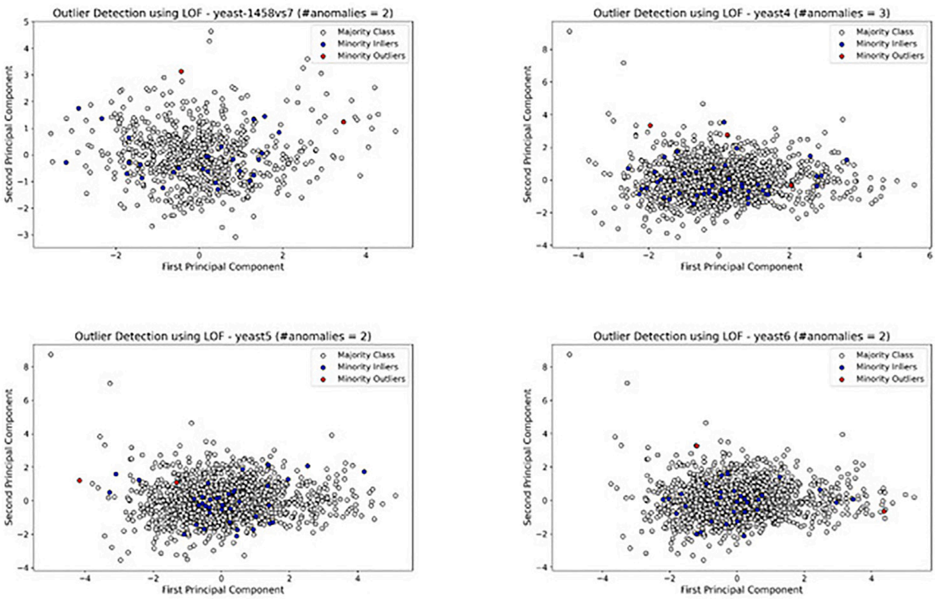
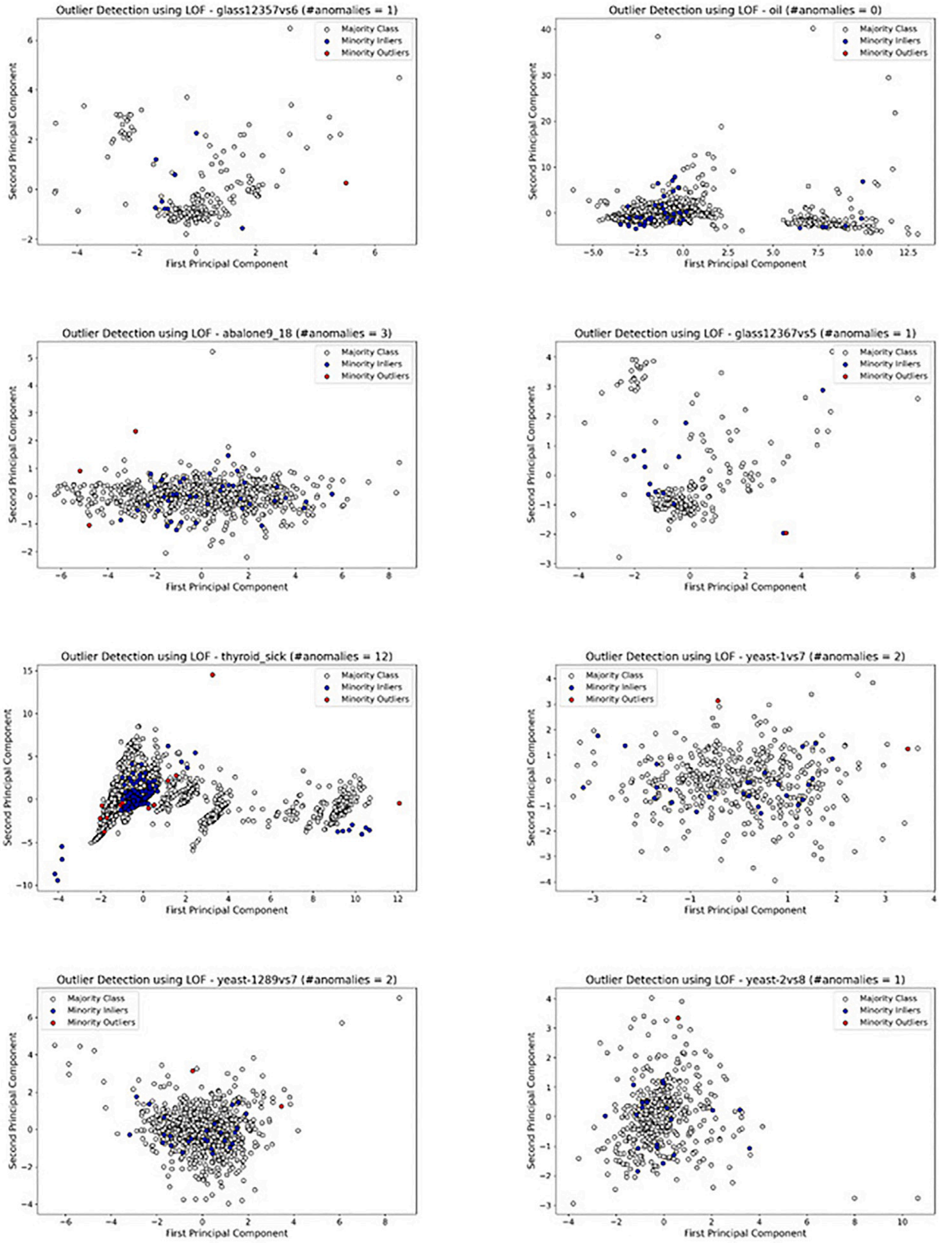


Fig. D.1. Principal component plots for the first two components, showing the distribution of outliers within the minority class across 4 datasets detected by the Local Outlier Factor (LOF) method.



Outlier Detection using LOF - glass12357vs6 (#anomalies = 1)

Outlier Detection using LOF - oil (#anomalies = 0)

Outlier Detection using LOF - abalone9_18 (#anomalies = 3)

Outlier Detection using LOF - glass12367vs5 (#anomalies = 1)

Outlier Detection using LOF - thyroid_sick (#anomalies = 12)

Outlier Detection using LOF - yeast-1vs7 (#anomalies = 2)

Outlier Detection using LOF - yeast-1289vs7 (#anomalies = 2)

Outlier Detection using LOF - yeast-2vs8 (#anomalies = 1)

Fig. D.2. Principal component plots for the first two components, showing the distribution of outliers within the minority class across 8 datasets detected by the Local Outlier Factor (LOF) method.

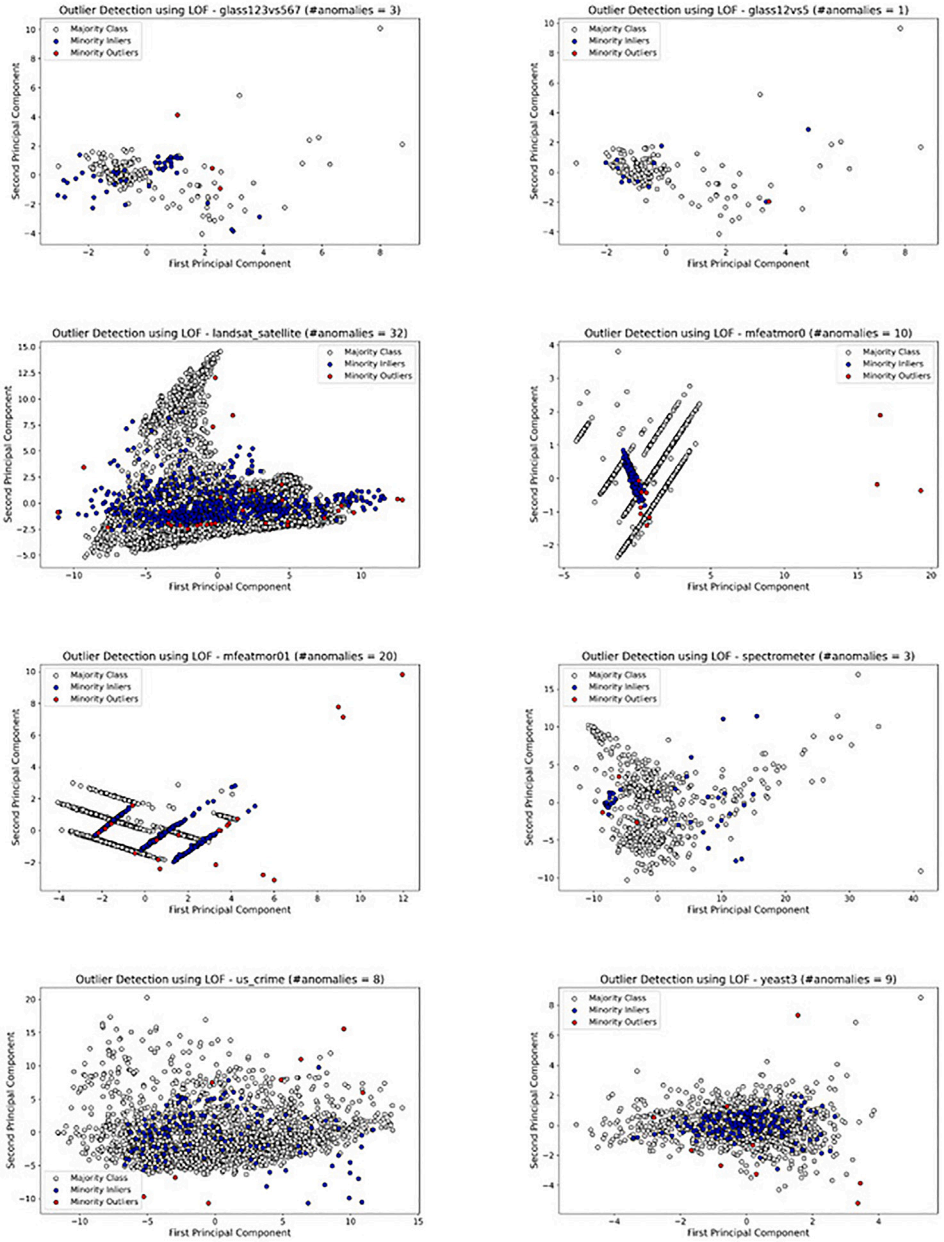


Fig. D.3. Principal component plots for the first two components, showing the distribution of outliers within the minority class across 8 datasets detected by the Local Outlier Factor (LOF) method.

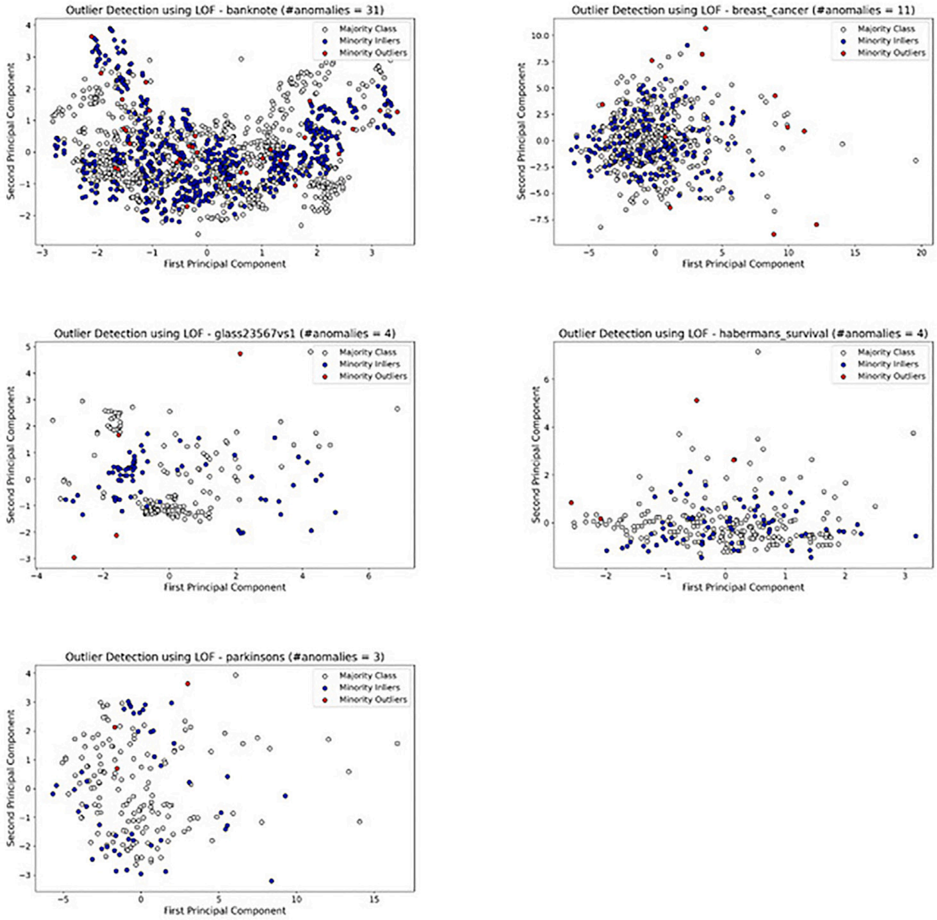


Fig. D.4. Principal component plots for the first two components, showing the distribution of outliers within the minority class across 6 datasets detected by the Local Outlier Factor (LOF) method.

Table E.4
PR-AUC results of 17 comparative algorithms on 25 imbalanced datasets using 100 × 5 fold cross-validation with a Logistic Regression classifier.

Dataset	Distance ExtSMOTE	Dirichlet ExtSMOTE	BGMM SMOTE	FCRP SMOTE	SMOTE	SMOTE LOF	SMOTE TomekLinks	SMOTE ENN	SMOTE IPF	DBSMOTE	ProWSyn	DSMOTE	cluster SMOTE	Gaussian SMOTE	Borderline SMOTE	ADASYN	Baseline
yeast6	0.5159	0.5168	0.5392	0.5122	0.4988	0.4987	0.5027	**0.5758**	0.5039	0.5056	0.5092	0.3705	0.5016	0.5076	0.5129	0.4936	**0.5758**
yeast5	0.5521	0.5454	0.5507	0.5384	0.5373	0.5373	0.536	0.5626	0.5343	**0.5789**	0.5425	0.4867	0.4964	0.5347	0.5456	0.523	0.5626
yeast-1289vs7	0.3738	0.3786	0.4231	0.3798	0.406	0.407	0.4043	**0.4304**	0.4036	0.2161	0.4043	0.251	0.4188	0.4151	0.3916	0.4111	**0.4304**
yeast4	0.4274	0.4452	0.4407	0.4307	0.4477	0.4469	0.4452	0.3931	0.4485	0.4396	0.4598	0.2966	0.4327	**0.4851**	0.473	0.4364	0.3931
yeast-2vs8	0.4275	0.4863	0.4661	0.4439	0.5404	0.5417	0.5248	**0.7204**	0.525	0.4547	0.5573	0.6943	0.6402	0.5531	0.4242	0.4311	**0.7204**
glass12357vs6	0.696	**0.7865**	0.732	0.7472	0.7797	0.7797	0.7797	0.632	0.7797	0.7499	0.7797	0.5878	0.7752	0.6396	0.7797	0.7797	0.632
yeast-1458vs7	0.3133	0.3817	0.3218	0.3543	0.3794	0.3745	0.379	**0.5216**	0.3784	0.2888	0.3687	0.2279	0.3626	0.3081	0.3559	0.374	**0.5216**
oil	0.5758	0.5766	0.5395	0.5439	0.565	0.5652	0.5698	**0.5931**	0.5687	0.5798	0.5816	0.4934	0.5585	0.5317	0.5112	0.5646	**0.5931**
abalone9_18	0.5728	0.5782	0.5687	0.5679	0.5662	0.5665	0.5726	0.5728	0.5727	0.5662	0.5833	0.5971	0.5812	0.4768	**0.6093**	0.556	0.5728
glass12367vs5	0.4898	0.5948	0.5927	0.5296	0.5915	0.5902	0.5837	0.402	0.5849	0.3566	0.612	0.3815	0.48	0.5983	**0.617**	0.6031	0.402
thyroid_sick	0.6374	0.6412	0.6272	0.634	0.6358	0.6325	0.6345	**0.725**	0.6344	0.7214	0.653	0.6204	0.6704	0.5581	0.6267	0.6276	**0.725**
yeast-1vs7	0.4576	0.4577	**0.4868**	0.4493	0.4667	0.468	0.4681	0.345	0.4697	0.329	0.4607	0.4357	0.4574	0.4794	0.4394	0.4806	0.345
us_crime	0.535	0.5477	0.5472	0.5499	0.5601	0.5579	0.555	0.5428	0.5567	0.5663	0.5697	0.561	0.5618	**0.5893**	0.5704	0.5537	0.5428
glass12vs5	0.6741	0.8139	0.7321	0.7969	0.8054	0.8038	0.8002	0.7511	0.8001	0.5316	0.8174	0.661	0.7839	0.724	0.7677	**0.82**	0.7511
spectrometer	0.8691	0.871	0.872	0.8697	0.8593	0.8575	0.861	0.8764	0.8607	0.8623	0.8679	0.8624	0.8696	0.8159	0.8615	0.846	**0.8835**
landsat_satellite	0.4919	0.5039	0.5098	0.5144	0.5463	0.5467	0.5447	0.3316	0.5455	0.201	0.5281	0.198	0.5225	**0.5722**	0.5374	0.5408	0.3316
mfeatmor0	0.9932	0.9939	0.9932	0.9932	0.9922	0.9929	0.9916	0.9918	0.9919	0.9932	**0.995**	0.9767	0.9878	0.9777	0.9932	0.9906	0.9932
yeast3	0.7372	0.7343	0.7379	0.7306	0.712	0.7131	0.7132	**0.7729**	0.7131	0.7372	0.7141	0.7537	0.7085	0.7174	0.7157	0.7143	**0.7729**
mfeatmor01	0.9237	0.924	0.92	0.9224	0.9159	0.9191	0.9192	0.9316	0.9159	0.9305	0.9157	0.9215	0.9135	0.8753	0.8546	0.7935	**0.9361**
glass123vs567	0.8608	0.877	**0.8847**	0.8781	0.8625	0.8612	0.8629	0.829	0.8635	0.8682	0.8529	0.8719	0.8621	0.8517	0.8532	0.8565	0.8265
parkinsons	0.7053	0.7281	0.7111	0.7124	0.7103	0.7117	0.7088	**0.795**	0.7112	0.7295	0.7003	0.7707	0.7108	0.6817	0.719	0.7228	**0.795**
habermans_survival	0.5785	**0.5973**	0.5882	0.5844	0.5699	0.5721	0.5713	0.4906	0.5726	0.5649	0.571	0.4425	0.5823	0.5519	0.5612	0.583	0.4906
glass23567vs1	0.6938	0.7081	0.7053	0.6997	0.69	0.6912	0.6924	**0.7168**	0.6891	0.6884	0.6935	0.7115	0.7005	0.6998	0.7108	0.7105	**0.7168**
breast_cancer	0.9764	0.9772	0.9783	0.9766	0.9748	0.9747	0.9741	0.9782	0.9751	0.9776	0.9728	0.9565	0.9746	0.9727	0.9601	0.9585	**0.9785**
banknote	0.9907	0.9923	0.9933	0.9918	0.9937	0.9936	0.9937	**0.9941**	0.9936	0.9935	0.9935	0.9675	0.9938	0.9764	**0.9941**	NA	**0.9941**

Table E.5
PR-AUC results of 17 comparative algorithms on 25 imbalanced datasets using 100 × 5 fold cross-validation with a K-NN classifier.

Dataset	Distance ExtSMOTE	Dirichlet ExtSMOTE	BGMM SMOTE	FCRP SMOTE	SMOTE	SMOTE LOF	SMOTE TomekLinks	SMOTE ENN	SMOTE IPF	DBSMOTE	ProWSyn	DSMOTE	cluster SMOTE	Gaussian SMOTE	Borderline SMOTE	ADASYN	Baseline
yeast6	0.5039	0.5293	**0.5404**	0.5235	0.4336	0.4336	0.4301	0.526	0.4319	0.4658	0.4363	0.499	0.4366	0.4871	0.4244	0.396	0.526
yeast5	0.5413	0.614	0.5726	0.595	0.5339	0.5318	0.5282	**0.6832**	0.5304	0.6702	0.518	0.6309	0.5404	0.5096	0.557	0.4986	**0.6832**
yeast-1289vs7	0.1504	0.2851	0.2661	0.2706	0.2844	0.2896	0.2896	0.4304	0.2921	0.1815	0.3623	**0.4317**	0.2999	0.2992	0.263	0.285	0.4304
yeast4	0.381	0.4329	0.4089	0.4083	0.3732	0.3724	0.386	0.3728	0.3807	0.3596	0.4296	0.4267	0.366	**0.438**	0.3795	0.3684	0.3728
yeast-2vs8	0.5964	0.5803	0.5145	0.5066	0.4722	0.473	0.4452	**0.726**	0.453	0.6604	0.496	0.7193	0.4792	0.4881	0.3818	0.4328	**0.726**
glass12357vs6	**0.8547**	0.8181	0.7446	0.8073	0.8103	0.8137	0.8175	0.807	0.8185	0.8047	0.7708	0.8071	0.7822	0.6488	0.8103	0.8092	0.807
yeast-1458vs7	0.0543	0.2236	0.072	0.1114	0.165	0.1663	0.1714	0.1626	0.1774	0.3035	0.2638	**0.3803**	0.1763	0.1703	0.1606	0.1577	0.1626
oil	0.5408	0.63	0.5915	0.5815	0.5886	0.5908	0.5909	0.5833	0.5921	0.6324	0.5919	0.4367	0.5805	0.4946	**0.6395**	0.582	0.5833
abalone9_18	0.397	0.4714	0.4815	0.492	0.4502	0.4502	0.454	**0.5028**	0.4563	0.4858	0.4685	0.4912	0.4054	0.4372	0.4675	0.4254	**0.5028**
glass12367vs5	0.683	0.715	0.6674	0.7034	0.7042	**0.7271**	0.715	0.522	0.7183	0.559	0.6779	0.5861	0.6926	0.6525	0.7104	0.6589	0.522
thyroid_sick	0.5893	0.591	0.5941	0.5925	0.5984	0.5975	0.5944	0.5946	**0.5985**	0.583	0.5644	0.5491	0.5902	0.5738	0.582	0.5983	0.5946
yeast-1vs7	0.3686	0.4652	0.376	0.427	0.4576	0.4543	0.454	0.4896	0.4578	0.4211	0.449	**0.5223**	0.4791	0.4136	0.4824	0.428	0.4896
us_crime	0.4205	0.4505	0.4434	0.4513	0.4314	0.431	0.4357	**0.4939**	0.4315	0.3813	0.4605	0.4819	0.4379	0.439	0.4322	0.4216	**0.4939**
glass12vs5	0.448	0.6247	0.5843	0.6096	0.6885	0.6734	0.704	0.3668	0.7005	0.4678	0.697	0.4901	0.6559	**0.7454**	0.7015	0.6936	0.3668
spectrometer	0.8196	**0.8879**	0.7993	0.8499	0.8584	0.8631	0.8634	0.8543	0.8608	0.8423	0.8398	0.8287	0.8486	0.8854	0.8263	0.8425	0.8635
landsat_satellite	0.6895	0.7197	0.695	0.712	0.7202	0.7203	0.7179	0.7189	0.7187	0.7171	0.7045	0.5535	0.714	0.6931	0.71	**0.7213**	0.7189
mfeatmor0	0.9932	**0.9941**	0.9932	0.9932	0.9857	0.9898	0.987	0.9908	0.9878	0.9932	0.9895	0.9572	0.9886	0.9661	0.9932	0.9885	0.9932
yeast3	0.7475	0.7633	0.7685	0.7645	0.7396	0.7393	0.7441	**0.7776**	0.7419	0.7263	0.7752	0.7685	0.7184	0.7415	0.6655	0.7076	**0.7776**
mfeatmor01	0.9555	0.9564	0.9573	0.9534	0.9508	0.9506	0.9516	0.9559	0.9514	0.9439	0.9501	0.9482	0.9474	0.9469	0.9264	0.8968	**0.9613**
glass123vs567	0.8877	**0.8949**	0.8936	0.8932	0.8751	0.8744	0.877	0.8783	0.8744	0.873	0.8874	0.8642	0.8703	0.8861	0.8718	0.8708	0.88
parkinsons	**0.8967**	0.8962	0.8793	0.8884	0.8625	0.8634	0.8624	0.8116	0.8667	0.7986	0.8323	0.8139	0.8681	0.8608	0.8765	0.8883	0.8116
habermans_survival	0.3881	0.4952	0.4722	0.4706	0.4947	0.4945	0.4899	0.3079	0.4958	0.3867	0.5228	0.4188	0.4755	**0.5378**	0.4953	0.4993	0.3079
glass23567vs1	0.7802	**0.799**	0.7864	0.7851	0.7643	0.7641	0.7649	0.7477	0.7637	0.7409	0.7425	0.739	0.7865	0.7394	0.7709	0.7745	0.7477
breast_cancer	0.9597	**0.9649**	0.961	0.961	0.9585	0.9591	0.9567	0.956	0.9591	0.9539	0.9568	0.9491	0.9576	0.9545	0.9527	0.9523	0.9561
banknote	0.9984	**0.9987**	0.9984	0.9984	0.9984	0.9984	0.9984	0.9984	0.9984	0.9984	0.9982	0.9709	0.9984	0.9862	0.9984	NA	0.9984

Table E.6
PR-AUC results of 17 comparative algorithms on 25 imbalanced datasets using 100 × 5 fold cross-validation with a Random Forest classifier.

Dataset	Distance ExtSMOTE	Dirichlet ExtSMOTE	BGMM SMOTE	FCRP SMOTE	SMOTE	SMOTE LOF	SMOTE TomekLinks	SMOTE ENN	SMOTE IPF	DBSMOTE	ProWSyn	DSMOTE	cluster SMOTE	Gaussian SMOTE	Borderline SMOTE	ADASYN	Baseline
yeast6	0.5985	0.6371	**0.6574**	0.6272	0.5672	0.5727	0.5671	0.5509	0.572	0.5547	0.524	0.5489	0.5812	0.5589	0.5464	0.5544	0.5486
yeast5	0.6451	0.6567	0.6519	0.6536	0.5511	0.5539	0.5479	0.6262	0.543	0.6217	0.5529	**0.6884**	0.5738	0.636	0.5865	0.5361	0.6254
yeast-1289vs7	0.2089	0.1979	0.1151	0.1416	0.1963	0.1934	0.2029	0.3847	0.1935	0.3645	0.2027	0.3992	0.3096	**0.408**	0.2573	0.1633	0.3803
yeast4	0.3972	0.4346	0.4023	0.4019	0.4275	0.4246	0.4139	0.4329	0.4094	0.443	0.4489	0.4008	0.4639	0.4465	**0.4701**	0.4173	0.4364
yeast-2vs8	0.6874	0.6841	0.6127	0.5519	0.5006	0.5016	0.4952	0.7086	0.492	0.7182	0.5856	0.7161	0.6208	**0.7374**	0.4123	0.3399	0.7098
glass12357vs6	0.7966	**0.8043**	0.7837	0.767	0.7786	0.7764	0.774	0.744	0.7742	0.7533	0.7903	0.7324	0.7865	0.6644	0.7834	0.7579	0.7459
yeast-1458vs7	0.3463	0.2426	0.0645	0.1196	0.1564	0.1509	0.1521	0.3748	0.1439	0.1689	0.1794	**0.3982**	0.1256	0.2082	0.1653	0.1471	0.3806
oil	0.5564	0.6061	0.5507	0.6258	0.5707	0.5748	0.5795	0.5672	0.5807	0.5084	**0.6505**	0.5644	0.5265	0.6207	0.5882	0.5718	0.5615
abalone9_18	0.4409	0.4563	0.4393	0.4374	0.4048	0.4039	0.4099	0.4582	0.4083	0.4755	0.4471	**0.4875**	0.4204	0.4243	0.4375	0.3885	0.4638
glass12367vs5	0.7892	**0.8722**	0.7957	0.8054	0.8031	0.804	0.8141	0.7448	0.8131	0.797	0.8284	0.7601	0.7729	0.7668	0.8156	0.7997	0.7461
thyroid_sick	0.8688	**0.8892**	0.8773	0.8669	0.8799	0.8758	0.8775	0.8692	0.878	0.8678	0.8557	0.8284	0.8839	0.8778	0.8748	0.8762	0.8684
yeast-1vs7	0.3842	0.4086	0.2619	0.3309	0.3883	0.3881	0.3724	0.4836	0.3853	0.3734	0.3121	**0.5329**	0.4009	0.4374	0.3938	0.3446	0.4886
us_crime	0.5185	0.5242	0.5059	0.521	0.5231	0.5227	0.5203	0.5476	0.5232	0.506	0.5421	0.5432	0.5162	0.5441	0.5231	0.513	**0.55**
glass12vs5	0.7239	**0.845**	0.8024	0.8171	0.8193	0.8254	0.8211	0.7539	0.8198	0.7976	0.821	0.7089	0.7718	0.8134	0.8199	0.8173	0.7556
spectrometer	0.8507	**0.8742**	0.8301	0.8482	0.8284	0.8375	0.8292	0.8603	0.828	0.8639	0.8219	0.845	0.8307	0.8381	0.8306	0.8144	0.8601
landsat_satellite	0.6831	0.698	0.6932	0.6916	0.7035	0.6994	0.7009	0.7044	0.7012	0.7033	0.6972	0.6857	**0.7074**	0.6576	0.7013	0.7023	0.7051
mfeatmor0	0.9932	**0.9934**	0.9932	0.9932	0.9898	0.9923	0.9896	0.9901	0.9892	0.9919	0.9921	0.9603	0.9887	0.9889	0.9932	0.9881	0.9932
yeast3	0.7836	**0.8006**	0.7929	0.7968	0.7977	0.799	0.7984	0.779	0.799	0.7862	**0.8006**	0.7797	0.7968	0.785	0.7925	0.7907	0.7801
mfeatmor01	0.9585	0.9629	0.961	0.9572	0.9555	0.9562	0.9551	0.9584	0.955	0.96	0.9616	0.9505	0.9533	0.9588	0.9377	0.901	**0.9637**
glass123vs567	0.9076	**0.9172**	0.907	0.9127	0.9034	0.903	0.9016	0.9035	0.907	0.9044	0.9085	0.8573	0.9025	0.9024	0.8943	0.8909	0.9032
parkinsons	0.8064	0.844	0.8315	0.833	0.8426	0.8433	0.8432	0.836	0.844	0.8353	0.821	0.7942	**0.849**	0.8222	0.8349	0.847	0.8361
habermans_survival	0.4767	0.5015	0.4834	0.506	0.4704	0.4676	0.485	0.428	0.4882	0.4356	0.5086	0.4228	0.468	**0.5287**	0.4624	0.4677	0.4263
glass23567vs1	0.8478	**0.8756**	0.8583	0.8544	0.8516	0.8519	0.8493	0.8547	0.8483	0.861	0.827	0.8561	0.8507	0.8237	0.8471	0.8456	0.8583
breast_cancer	0.9564	**0.9598**	0.9566	0.9574	0.9556	0.9558	0.9568	0.956	0.9558	0.9574	0.9541	0.9502	0.9561	0.9547	0.953	0.9575	0.9561
banknote	0.9936	**0.9944**	0.9937	0.9939	0.9936	0.9935	0.9935	0.9933	0.9936	0.9934	0.9932	0.964	0.9936	0.9906	0.9933	NA	0.9933

References

Asniar, Maulidevi, N. U., & Surendro, K. (2022). SMOTE-LOF for noise identification in imbalanced data classification. *Journal of King Saud University. Computer and Information Sciences*, *34*(6), 3413–3423.

Barua, S., Islam, M. M., Yao, X., & Murase, K. (2014). MWMOTE–majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Transactions on Knowledge and Data Engineering*, *26*(2), 405–425.

Barua, S., Islam Md, M., & Murase, K. (2013). ProWSyn: Proximity weighted synthetic oversampling technique for imbalanced data set learning. In *Advances in knowledge discovery and data mining* (pp. 317–328). Springer Berlin Heidelberg.

Batista, G. E. A. P. A., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations*, *6*(1), 20–29.

Bej, S., Davtyan, N., Wolfien, M., Nassar, M., & Wolkenhauer, O. (2021). LoRAS: an oversampling approach for imbalanced datasets. *Machine Learning*, *110*(2), 279–301.

Bela, A. F., Amol, K., & Maya, R. G. (2010). *Introduction to the Dirichlet distribution and related processes*: *Technical report*, University of Washington Department of Electrical Engineering.

Blagus, R., & Lusa, L. (2013). SMOTE for high-dimensional class-imbalanced data. *BMC Bioinformatics*, *14*(1), 106.

Blei, D. M., & Jordan, M. I. (2006). Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, *1*(1).

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

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: identifying density-based local outliers. In *International conference on management of data: proceedings of the 2000 ACM SIGMOD international conference on management of data : Dallas, Texas, United states; 15-18 May 2000* (pp. 93–104). New York, NY, USA: ACM.

Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2012). DBSMOTE: Density-based synthetic minority over-sampling technique. *Applied Intelligence (Dordrecht, Netherlands)*, *36*(3), 664–684.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *The Journal of Artificial Intelligence Research*, *16*, 321–357.

Chawla, N. V., Lazarevic, A., Hall, L. O., & Bowyer, K. W. (2003). SMOTEBoost: Improving prediction of the minority class in boosting. In *Knowledge discovery in databases: PKDD 2003* (pp. 107–119). Berlin, Heidelberg: Springer Berlin Heidelberg.

Cieslak, D., Chawla, N., & Striegel, A. (2006). Combating imbalance in network intrusion datasets. In *2006 IEEE international conference on granular computing* (pp. 732–737).

Couronné, R., Probst, P., & Boulesteix, A.-L. (2018). Random forest versus logistic regression: a large-scale benchmark experiment. *BMC Bioinformatics*, *19*(1), 270.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21–27.

Digital Research Alliance of Canada (2024). Digital Research Alliance of Canada. https://alliancecan.ca/.

Duan, L., Xu, L., Guo, F., Lee, J., & Yan, B. (2007). A local-density based spatial clustering algorithm with noise. *Information Systems (Oxford)*, *32*(7), 978–986.

Duin, R. (2024). *Multiple Features*. UCI Machine Learning Repository, http://dx.doi.org/10.24432/C5HC70.

Dunn, O. J. (1964). Multiple comparisons using rank sums. *Technometrics*, *6*(3), 241–252.

Feng, S., Keung, J., Zhang, P., Xiao, Y., & Zhang, M. (2022). The impact of the distance metric and measure on SMOTE-based techniques in software defect prediction. *Information and Software Technology*, *142*, 106742–.

Fernández, A., del Jesus, M. J., & Herrera, F. (2010). On the 2-tuples based genetic tuning performance for fuzzy rule based classification systems in imbalanced data-sets. *Information Sciences*, *180*(8), 1268–1291.

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, *32*(200), 675–701.

Gao, M., Hong, X., Chen, S., & Harris, C. J. (2011). On combination of SMOTE and particle swarm optimization based radial basis function classifier for imbalanced problems. In *The 2011 international joint conference on neural networks* (pp. 1146–1153). IEEE.

German, B. (1987). *Glass identification*. UCI Machine Learning Repository, http://dx.doi.org/10.24432/C5WW2P.

Haberman, S. (1999). *Haberman's survival*. UCI Machine Learning Repository, http://dx.doi.org/10.24432/C5XK51.

Han, H., Wang, W.-Y., & Mao, B.-H. (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In *Advances in intelligent computing* (pp. 878–887). Berlin, Heidelberg: Springer Berlin Heidelberg.

Hawkins, D. (1980). Identification of outliers. In *Monographs on applied probability and statistics*, Dordrecht: Springer Netherlands, SpringerLink (Online service).

He, H., Bai, Y., Garcia, E., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*: *vol. 10*, (pp. 1322–1328). IEEE.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, *6*(2), 65–70.

Hu, S., Liang, Y., Ma, L., & He, Y. (2009). MSMOTE: Improving classification performance when training data is imbalanced. In *2009 second international workshop on computer science and engineering*: *vol. 2*, (pp. 13–17). IEEE.

Jain, N., Manikonda, L., Hernandez, A. O., Sengupta, S., & Kambhampati, S. (2018). *Imagining an engineer: On GAN-based data augmentation perpetuating biases*. Ithaca: Cornell University Library, arXiv.org.

Kovács, G. (2019). An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets. *Applied Soft Computing*, *83*, 105662–.

Lee, H., Kim, J., & Kim, S. (2017). Gaussian-based smote algorithm for solving skewed class distributions. *International Journal og Fuzzy Logic Intelligent Systems*, *17*, 229–234.

Lemaitre, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, *18*, 1–5.

Little, M. (2008). *Parkinsons*. UCI Machine Learning Repository, http://dx.doi.org/10.24432/C59C74.

Lohweg, V. (2013). *Banknote authentication*. UCI Machine Learning Repository, http://dx.doi.org/10.24432/C55P57.

(1988). *Low resolution spectrometer*. UCI Machine Learning Repository, http://dx.doi.org/10.24432/C5B02R.

Mahmoudi, S., Moradi, P., Akhlaghian, F., & Moradi, R. (2014). Diversity and separable metrics in over-sampling technique for imbalanced data classification. In *2014 4th international conference on computer and knowledge engineering* (pp. 152–158).

Matharaarachchi, S., Domaratzki, M., & Muthukumarana, S. (2021). Assessing feature selection method performance with class imbalance data. *Machine Learning with Applications*, *6*, 100170–.

Mathew, J., Luo, M., Pang, C. K., & Chan, H. L. (2015). Kernel-based SMOTE for SVM classification of imbalanced datasets. In *IECON 2015 - 41st annual conference of the IEEE industrial electronics society* (pp. 001127–001132).

Mazurowski, M. A., Habas, P. A., Zurada, J. M., Lo, J. Y., Baker, J. A., & Tourassi, G. D. (2008). Training neural network classifiers for medical decision making: the effects of imbalanced datasets on classification performance. *Neural Networks : The Official Journal of the International Neural Network Society*, *21*(2–3), 427—436.

Nakai, K. (1996). *Yeast*. UCI Machine Learning Repository, http://dx.doi.org/10.24432/C5KG68.

Nash, W., Sellers, T., Talbot, S., Cawthorn, A., & Ford, W. (1995). *Abalone*. UCI Machine Learning Repository, http://dx.doi.org/10.24432/C55C7W.

Newman, C., Blake, D., & Merz, C. (1998). *UCI repository of machine learning databases*. University of California, Irvine, Dept. of Information and Computer Sciences.

Ng, K. W., Tang, M.-L., & Tian, G.-L. (2011). Dirichlet and related distributions: Theory, methods and applications, In *Wiley series in probability and statistics*: *vol. 895*, (1. Aufl.). Newark: Wiley.

Ng, K. W., Tian, G.-L., & Tang, M.-L. (2011). *Wiley series in probability and statistics*, *Dirichlet and related distributions : Theory, methods and applications*. Chichester, England: Wiley.

Pastaltzidis, I., Dimitriou, N., Quezada-Tavarez, K., Aidinlis, S., Marquenie, T., Gurzawska, A., & Tzovaras, D. (2022). Data augmentation for fairness-aware machine learning: Preventing algorithmic bias in law enforcement systems. In *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency* (pp. 2302–2314). New York, NY, USA: ACM.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pitman, J., & Picard, J. (2006). *Lecture notes in mathematics ; 1875*, *Combinatorial stochastic processes: Ecole d'ete de probabilities de Saint-Flour XXXII - 2002* (1st ed. 2006). Berlin: Springer.

Roberts, S., Husmeier, D., Rezek, I., & Penny, W. (1998). Bayesian approaches to Gaussian mixture modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(11), 1133–1142.

Sáez, J. A., Luengo, J., Stefanowski, J., & Herrera, F. (2015). SMOTE–IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Information Sciences*, *291*, 184–203.

Shah, K., Patel, H., Sanghvi, D., & Shah, M. (2020). A comparative analysis of logistic regression, random forest and KNN models for the text classification. *Augmented Human Research*, *5*(1).

Sharma, S., Zhang, Y., Ríos Aliaga, J. M., Bouneffouf, D., Muthusamy, V., & Varshney, K. R. (2020). Data augmentation for discrimination prevention and bias disambiguation. In *Proceedings of the AAAI/ACM conference on AI, ethics, and society* (pp. 358–364). New York, NY, USA: ACM.

Srinivasan, A. (1993). *Statlog (Landsat satellite)*. UCI Machine Learning Repository, http://dx.doi.org/10.24432/C55887.

Suh, Y., Kim, C., Song, L., Yu, J., & Mo, J. (2017). A comparison of oversampling methods on imbalanced topic classification of Korean news articles. *Journal of Cognitive Science*, *18*, 391–437.

Tomalin, M., Byrne, B., Concannon, S., Saunders, D., & Ullmann, S. (2021). The practical ethics of bias reduction in machine translation: why domain adaptation is better than data debiasing. *Ethics and Information Technology*, *23*(3), 419–433.

Turlapati, V. P. K., & Prusty, M. R. (2020). Outlier-SMOTE: A refined oversampling technique for improved detection of COVID-19. *Intelligence-Based Medicine*, *3–4*, 100023.

Weisberg, S. (2005). *Applied Linear Regression*. Hoboken: John Wiley & Sons, Incorporated.

Wolberg, W., Mangasarian, O., Street, N., & Street, W. (1995). *Breast cancer wisconsin (Diagnostic)*. UCI Machine Learning Repository, http://dx.doi.org/10.24432/C5DW2B.

Yang, Z., Tang, W., Shintemirov, A., & Wu, Q. (2009). Association rule mining-based dissolved gas analysis for fault diagnosis of power transformers. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, *39*, 597–610.

Zhu, B., Baesens, B., Backiel, A., & vanden Broucke, S. K. L. M. (2018). Benchmarking sampling techniques for imbalance learning in churn prediction. *Journal of the Operational Research Society*, *69*(1), 49–65.