CeC-SMOTE: A Clustering and Centroid-Based Adaptive Oversampling Method for Imbalanced Data

Xiaoling Gao[1], Marshima Mohd Rosli[2]*, Muhammad Izzad Ramli[3], Nursuriati Jamil[4]
Xinhua College of Ningxia University, Yinchuan, Ningxia, 750021, China[1,2]
College of Computing, Informatics and Mathematics
Universiti Teknologi MARA, Shah Alam 40450, Selangor, Malaysia[1,2,3,4]

***Abstract*—Class imbalance is a common challenge in real-world datasets, leading standard classifiers to perform poorly on underrepresented classes. Traditional oversampling techniques, such as SMOTE and its variants, often generate synthetic samples without fully considering the local data structure, resulting in increased noise and class overlap.This study introduces CeC-SMOTE, an adaptive oversampling method that integrates clustering and centroid-based strategies to enhance the quality of synthetic minority samples. By first partitioning minority instances using K-means clustering, CeC-SMOTE identifies safe and boundary regions, selectively generating new samples where they are most needed while filtering out noise. This targeted approach preserves the underlying distribution of the minority class and minimizes the risk of overfitting. Extensive experiments on artificial and benchmark UCI datasets demonstrate that CeC-SMOTE consistently delivers competitive or superior results compared to established oversampling techniques, particularly in cases with complex or ambiguous class boundaries. Sensitivity analysis confirms that the method is robust to parameter settings, enabling strong performance with minimal tuning.**

Keywords*—Imbalanced data classification; synthetic oversampling; k-means clustering; centroid-based neighbor

I. Introduction

Imbalanced data, where some classes are significantly underrepresented compared to others, is a widespread issue in real-world applications. This imbalance often causes models to favor majority classes while neglecting minority categories [1][2]. For example, in healthcare, class imbalance can hinder the early detection of rare diseases, leading to missed diagnoses and delayed treatment. Similarly, in financial systems, fraudulent transactions are often overshadowed by legitimate ones, making it difficult for conventional models to identify and prevent fraud [3].

Traditional classification algorithms, which typically assume a balanced class distribution or seek to minimize overall error, tend to underperform on imbalanced datasets. These models frequently exhibit a bias toward the majority class, resulting in high accuracy for common categories but poor performance for the minority class—often the primary focus in critical applications [4]. This bias leads to minority instances being overlooked or misclassified, sometimes even treated as noise, which compromises both the reliability and generalizability of predictive models. To address these limitations, data-level solutions such as oversampling have been developed to enhance the representation of minority classes by generating synthetic samples.

The introduction of the SMOTE algorithm marked a significant advancement in oversampling by creating synthetic minority samples through interpolation between existing neighbors. This approach has inspired many variants, each aiming to overcome specific limitations. For example, Borderline-SMOTE [5] generates synthetic samples near decision boundaries, while Safe-Level-SMOTE [6] focuses on safer regions to reduce noise amplification. Other methods, such as SVM combine SMOTE [7] and cluster-based approaches, refine the placement of synthetic samples using support vectors or cluster centroids [8]. Further advancements, like LD-SMOTE [9] and Simplicial SMOTE [10], leverage local density, information entropy, and topological structures to ensure that synthetic samples align more closely with the actual data distribution.

Despite these advances, imbalanced learning continues to evolve as researchers seek new methods that improve sample quality, minimize noise, and enhance classifier robustness, particularly in complex data environments. In this context, we propose CeC-SMOTE—an adaptive oversampling technique that combines clustering and centroid-based analysis. CeC-SMOTE systematically categorizes minority instances by utilizing both global and local cluster characteristics, applies a Nearest Centroid Neighbour strategy for more effective neighbor selection, and adaptively limits synthetic data generation to safe and boundary regions to reduce noise and overfitting. This approach addresses important gaps in existing oversampling methods. The primary contributions of this study are as follows:

- We introduce an adaptive oversampling algorithm that integrates centroid-based clustering with safety-aware sample selection.
- We propose a systematic method for filtering noisy samples and directing oversampling to the most informative minority regions.
- We present extensive experiments on multiple benchmark datasets, demonstrating that CeC-SMOTE outperforms established oversampling techniques.

Accordingly, the research question guiding this study is: How can synthetic minority sample generation be improved to better preserve the structure of imbalanced data, minimize noise, and enhance classification accuracy, especially in challenging datasets with complex class boundaries?

The remainder of this study is organized as follows: Section II reviews related work on imbalanced data learning and oversampling. Section III details the methodology of the proposed

*Corresponding authors.

CeC-SMOTE algorithm. Section IV describes the experimental setup and presents the results. Section V discusses the findings and their implications, and Section VI concludes the study with directions for future research.

II. Related Works

Imbalanced data remains a significant challenge in classification problems, often resulting in poor performance for minority classes. To address this issue, researchers have developed various techniques to improve the detection and representation of these underrepresented groups. This section reviews the most relevant methods, highlighting advances in oversampling and clustering-based strategies designed to enhance minority class representation.

A. SMOTE and Variants

Oversampling techniques are widely used to address class imbalance by generating additional synthetic instances of the minority class [11]. This strategy is generally preferred over undersampling, which reduces the number of majority class samples and may result in the loss of valuable information crucial for accurate learning [12] [13]. Among these approaches, the Synthetic Minority Over-sampling Technique (SMOTE) [14] is particularly well-known. SMOTE creates synthetic samples by linearly interpolating between existing minority instances and their nearest minority neighbors [15][16]. Its main goals are to expand the decision region of the minority class and reduce the risk of overfitting that often arises from simply duplicating existing minority data [17].

Despite its popularity, SMOTE's linear interpolation mechanism introduces several notable limitations, especially when applied to complex datasets. First, SMOTE often generates synthetic samples without considering the distribution of the majority class [18]. This can lead to overgeneralization, where new samples encroach into majority class regions or inadvertently bridge distinct minority sub-clusters, increasing class overlap and complicating subsequent classification tasks [19]. Second, SMOTE is susceptible to noise and outliers in the minority class. When outliers or borderline samples are present, the algorithm may create synthetic instances around them, unintentionally amplifying noise and reducing class separability [20]. Third, SMOTE assumes linear relationships among minority samples, which is not always appropriate for datasets with non-linear structures or multiple, distinct minority clusters. In such cases, synthetic samples may not accurately capture the true distribution of the minority class [21]. Additionally, SMOTE faces challenges in high-dimensional data [22]. The nearest neighbor concept, central to its function, becomes less meaningful and more computationally intensive in higher dimensions, which can result in less diverse or even misleading synthetic samples [23]. Finally, SMOTE can struggle with small, isolated minority regions [24]. The algorithm may fail to adequately represent these unique data pockets, leading to their persistent underrepresentation or mischaracterization [25].

B. Enhancing Oversampling through Clustering

To overcome the limitations of traditional oversampling, researchers have incorporated clustering as a preparatory step. In clustering-based SMOTE variants, the minority class is first clustered, and then SMOTE is applied within each cluster [26]. This approach preserves local data structures and addresses imbalances within the minority class itself [27].

Combining clustering with oversampling offers several advantages. First, it reveals underlying subgroups within the minority class, enabling more targeted and nuanced oversampling. By recognizing these sub-clusters, oversampling can be focused on sparser or more critical regions rather than treating all minority samples equally, as standard SMOTE does. Second, clustering helps to identify and manage noise or outliers. Instances that do not fit well into any cluster can be excluded from synthetic sample generation or handled separately, reducing the risk of amplifying noise. Finally, clustering allows for differentiated oversampling strategies which more synthetic data can be generated in sparse but well-defined clusters, or efforts can be focused on safe regions to minimize overlap with the majority class.

Various clustering methods guide the oversampling process. K-means clustering [28]is a commonly used technique that partitions the data into distinct groups based on their centroids. K-Means SMOTE integrates this method by generating synthetic samples within each minority cluster, ensuring that new instances reflect the local distribution and preventing the artificial connection of unrelated groups [29][30] [31]. Other approaches, like LD-SMOTE, adjust the number of synthetic samples based on cluster density, focusing efforts on underrepresented regions and improving sample representativeness [9].

Density-based clustering algorithms, such as DBSCAN, are also employed for their ability to handle irregularly shaped clusters and detect noise [32]. For instance, DBSMOTE uses DBSCAN to cluster minority samples and place new synthetic points towards the center of each cluster, strengthening their core representation [33]. Similarly, adaptive clustering SMOT within clusters formed by DBSCAN [34], and clustering and optimization-based G-mean iteratively applies SMOTE in K-means clusters while optimizing performance metrics such as G-mean [35]. Some methods also use information entropy to monitor and control ambiguity in overlapping regions created by synthetic samples [36].

Despite significant advances in oversampling techniques, several common shortcomings remain across the reviewed methods. Many approaches, such as SMOTE and its variants, tend to generate synthetic samples without fully accounting for the underlying data distribution, leading to increased noise and class overlap. Methods that rely on linear interpolation may struggle with datasets exhibiting complex or nonlinear minority class structures. Additionally, many approaches are also sensitive to clustering quality and may perform poorly on high-dimensional or highly imbalanced datasets. These limitations underscore the need for more adaptive solutions.

III. CeC-SMOTE Methodology

This section delineates the technological process of CeC-SMOTE in greater detail. The flowchart illustrating imbalanced learning using CeC-SMOTE is presented in Fig. 1. The CeC-SMOTE algorithm is principally concerned with the following processes: the capture of the local structure of the minority

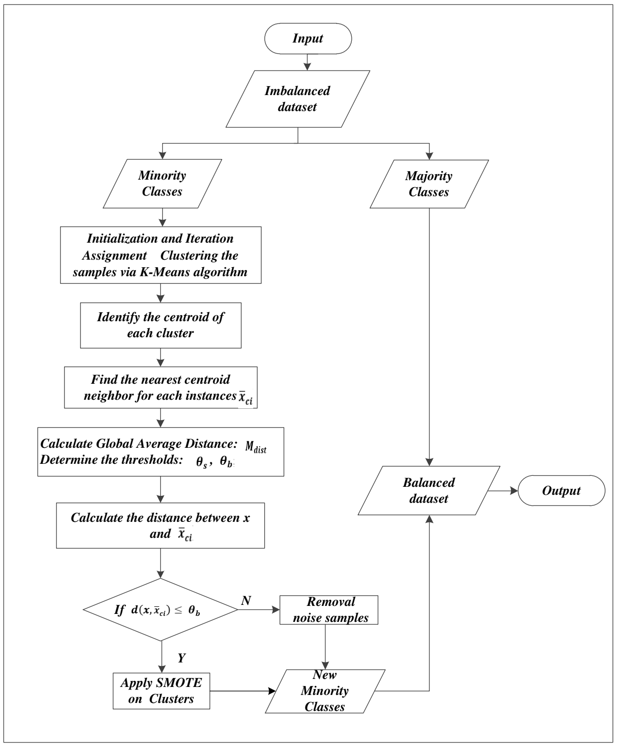

Input
Imbalanced dataset
Minority Classes
Majority Classes
Initialization and Iteration Assignment Clustering the samples via K-Means algorithm
Identify the centroid of each cluster
Find the nearest centroid neighbor for each instances $\bar{x}_{ci}$
Calculate Global Average Distance: M_{dist}
Determine the thresholds: θ_s, θ_b
Calculate the distance between x and $\bar{x}_{ci}$
If $d(x,\bar{x}_{ci}) \leq \theta_b$
N
Y
Removal noise samples
Apply SMOTE on Clusters
New Minority Classes
Balanced dataset
Output

Fig. 1. Block diagram of CeC-SMOTE.

class; the identification of "safe" minority points, with outliers that are too close to majority territory being discarded; and the expansion of the minority class, followed by the rebuilding of the full training set.

The process begins by applying K-means clustering to group the minority class samples, and each cluster's centroid is determined. The algorithm then calculates the global average distance within each cluster to categorize samples as "safe", "boundary", or "noise". Outliers and samples near the majority class are discarded to reduce potential overlap and noise. Next, SMOTE is selectively applied to the safe and boundary regions within each cluster, generating synthetic minority samples where they are most needed. Finally, the new synthetic samples are combined with the filtered original data, resulting in a more balanced dataset that is ready for training the classifier. While LD-SMOTE [9] and Simplicial SMOTE [10] have introduced local density estimation and topological techniques, CeC-SMOTE further advances the field by integrating centroid-based clustering with safety-aware filtering, ensuring that synthetic samples are generated primarily in well-defined, safe or boundary minority regions, as opposed to uniform interpolation.

A. Construct Minority Clusters

To capture the local structure of the minority class, this step involves initialising centroids and iteratively assigning samples to the nearest cluster based on their features. According to the step of the K-means clustering algorithm, until the change in the cluster centre is very small or the preset number of iterations is reached, $k$cluster centres are created to ensure that all samples are stable.

STEP 1: Randomly assign the initial point as the initial cluster centre point;

STEP 2: Calculate the Euclidean distance between all individuals in the sample set and the cluster centre, then assign the samples to the nearest cluster. The Euclidean distance between a point x and a centroid x_{c_i} in cluster c_i is given by Eq. (1):

$$\begin{aligned} d(x, x_{c_i}) = \|x - x_{c_i}\|_2 \\ = \sqrt{(x_1 - x_{c1})^2 + (x_2 - x_{c2})^2 + \cdots + (x_n - x_{cn})^2} \end{aligned} \tag{1}$$

STEP 3: Recalculate the cluster centre according to the samples in the class;

STEP 4: Iterate Steps 2 and 3 until the centroids no longer change significantly or the iteration limit is reached.

Step 2 is critical as it groups similar points together based on their proximity to the centroids. After all points have been assigned to clusters, the centroids are updated to reflect the means of their respective clusters.

B. Safety Assessment and Noise Cleaning

In order to improve class balance and accuracy while limiting noise and overfitting, After clusters the minority class and then identify "safe" minority points and discard outliers that sit too close to majority territory.

STEP 1: Apply the ensemble Nearest centroid neighborhood (NCN) strategy [37]for minority classes. For each data point, the algorithm identifies the nearest cluster centroid $\overline{x}_{ci}$. This involves calculating the distance from the point to each centroid and selecting the smallest one. This NCN helps to assess each data point's proximity to its cluster centre, which is pivotal for classifying the data points in later steps.

The idea of NCN is to find the nearest neighbor by the centroid. For a set of points $X_i = \{x_1, x_2, \cdots, x_n\}$, the centroid is calculated as Eq. (2):

$$\overline{X} = \frac{1}{n}(x_1 + x_2 + \cdots + x_n) \tag{2}$$

Neighbors should be distributed in geographical areas, and the idea of NCN requires that proximity be fully considered. First, NCN believes that the centroid neighbors are as close as possible to the test sample in terms of distance and are distributed as evenly as possible around the test sample. The NCN of sample p in X should be obtained by querying the following steps and the pseudocode in Algorithm 1.

STEP 2: Calculate the global average distance. Determine the global average distance by calculating the average distance of all samples to their respective centroids, which will serve as a benchmark for defining safe areas and boundary areas, as in Eq. (3):

$$M_{dist} = \frac{1}{m}\sum_{j=1}^{m} \left\|x_j - \overline{x}_{l_j}\right\|_2 \tag{3}$$

where, m is the total number of points in the dataset, $\overline{x}_{l_j}$ is the centroid of the cluster to which point x_j belongs, and l_j is the index pointing to the cluster to which point x_j belongs.

STEP 3: Classify regions within each cluster;

1) Define the thresholds for safe areas θ_s and boundary areas θ_b [see Eq. (4) and Eq. (5)];

$$\theta_s = \alpha \times M_{dist}, \quad (\alpha < 1) \tag{4}$$

$$\theta_b = \beta \times M_{dist}, \quad (\beta > \alpha) \tag{5}$$

2) Mark each sample as belonging to the safe zone, boundary area, or noise zone based on the comparison of the distance from the sample to the centroid $\mathbf{d}(x, \overline{x}_{c_i})$ and the global average distance M_{dist}:

- If $\mathbf{d}(x, \overline{x}_{c_i}) \leq \theta_s$, x is considered 'safe';
- if $\theta_s < \mathbf{d}(x, \overline{x}_{c_i}) \leq \theta_b$, x is considered 'boundary';
- and if $\mathbf{d}(x, \overline{x}_{c_i}) > \theta_b$, x is considered 'noise'.

By following these steps, the algorithm ensures that only points in well-defined, safe, or boundary regions are considered for oversampling, reducing the impact of noise and outliers.

```
Algorithm 1 Nearest Centroid Neighborhood (NCN)
Input: $X_i = \{x_1, x_2, \cdots, x_n\}$: input dataset; $k$: number of
    neighbors to search; $p$: query point
Output: $Q = \{q_1, q_2, \cdots, q_k\}$: set of $k$ nearest centroid
    neighbours
Initialize $Q = \emptyset$
Find the first NCN of $p$ as its nearest neighbour, $q_1$
$Q \leftarrow Q \cup \{q_1\}$
for $i = 2$ to $k$ do
    Select the $i$th neighbor $q_i$ such that the centroid of $q_i$
    and all previously selected neighbours $(q_1, q_2, \cdots, q_{i-1})$
    is the closest to $p$
    $Q \leftarrow Q \cup \{q_i\}$
end forreturn $Q$
```

C. Cluster-Aware Oversampling and Merge

After each sample is marked as belonging to the safe area, boundary area, or noise area. Within each cluster, apply SMOTE only to the safe points to create synthetic samples that stay inside the minority manifold.

STEP 1: First, for each sample $\mathbf{x}_i$ in the positive sample set, Euclidean distance is calculated between it and each other sample in the positive sample set, and k nearest neighbor samples are found, marked as $\mathbf{x}_i'$, $i \in \{1, 2, 3, \ldots, k\}$;

STEP 2: For each randomly selected neighbor sample $\mathbf{x}_i'$, a new sample is constructed according to Eq. (4) with $\mathbf{x}_i$, respectively [see Eq. (6)].

$$\mathbf{x}_{\text{new}} = \mathbf{x}_i + \text{rand}(0, 1) \times (\mathbf{x}_i' - \mathbf{x}_i) \tag{6}$$

STEP 3: Combine the generated samples to form the final dataset.

The CeC-SMOTE achieves a more balanced dataset through targeted oversampling of minority classes and removal of noisy, outlier samples. The pseudocode of the oversampling process of CeC-SMOTE is illustrated in Algorithm 2.

```
Algorithm 2 CeC-SMOTE
Input: minority imbalanced dataset $X^P$, $K$: maximum num-
    ber of clusters
Output: $D$: the dataset with augmented minority class sam-
    ples
Initialize;
Apply the K-means algorithm to the minority class;
Identify the centroid of each cluster: $C = \{c_1, c_2, \ldots, c_K\}$;
Nearest centroid neighbour: $x_{ci} = []$;
for each $x_i$ in $X^P$ do
    Initialize min_distance; Initialize $x_{ci} = []$;
    for each $c_j$ in $C$ do
        Calculate the Euclidean distance from $x_i$ to each
        centroid $c_j$;
        $d(x_i, c_j) = \|x_i - c_j\|_2$;
        Keep track of the centroid $c_j$ that has the minimum
        distance to $x_i$, storing the index of this nearest centroid
        $c_j$;
        if $d(x_i, c_j) <$ min_distance then
            min_distance $= d(x_i, c_j)$; update the index of
            the nearest centroid;
        end if
    end for
    Append the index of the nearest centroid to $x_{ci}$;
end forreturn $x_{ci}$
Calculate the global average distance between all samples
in $X^P$ and their respective nearest centroids.
    $$M_{dist} = \frac{1}{m}\sum_{j=1}^{m}\|x_j - x_{c_j}\|_2$$
Classify regions within each cluster:
    if $d(x, \overline{x}_{ci}) \leq \theta_s$, mark $x_i$ as "Safe"
    if $\theta_s < d(x, \overline{x}_{ci}) \leq \theta_b$, mark $x_i$ as "Boundary"
    else mark $x_i$ as "Noise"
Apply SMOTE to samples classified as "Safe" and
"Boundary" to generate synthetic samples $X_G^P$;
Merge the original minority class samples $X^P$ with the
newly generated synthetic samples $X_G^P$;
```

IV. Experimental Analysis

In this section, CeC-SMOTE is compared with four commonly used re-sampling methods on four metrics across seven UCI datasets and two artificial datasets under three quality evaluation measures, the G-mean, F1 score, AUC. We performed 5-fold cross-validation on all datasets to ensure the reliability of the results, averaging metrics across independent runs to reduce variance.

A. Dataset

To showcase the efficacy of CeC-SMOTE, the experimental data in this section are divided into three parts. The first are the two artificially generated two-dimensional (2D) datasets as shown in Table I. According to the definition of the imbalanced degree, dataset A is highly imbalanced, while dataset B is almost extremely imbalanced. Table II provides a detailed overview of 7 benchmark binary datasets. The IR of binary datasets selected from the UCI data repository varies from a minimum of 1.9 to a maximum of 129.5, with an average of 23.13. According to the imbalance ratio, Pima, ecoli3, Cleveland, vehicle, page-blocks, and Breast datasets are lowly imbalanced. The alone is extremely imbalanced.

TABLE I. Artificial Datasets

Dataset	Imbalance Ratio	Number of Datasets	Number of Minority Classes	Number of Majority Classes	Dimension
A	48.47	1484	30	1454	2
B	94.46	2673	28	2645	2

B. Experimental Metrics

In this study, the performance of CeC-SMOTE was validated using several metrics commonly used in imbalanced data learning: F1-score, geometric mean(G-Mean), and the area under the receiver operating characteristic curve (AUC) to systematically compare and analyze the efficacy of various methodologies. These metrics were selected due to their ability to balance the precision-recall trade-off and capture the model's discriminative power across both classes. The G-Mean is calculated to assess the balance between classification accuracies of the positive and negative classes. It is given by the following Eq. (7):

$$G\text{-}Mean = \sqrt{Specificity \times Recall} \tag{7}$$

where, recall (True Positive Rate), specificity (True Negative Rate) and precision (successfully identified positives out of all positives predicted) are respectively defined as Eq. (8) to Eq. (10):

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{8}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{9}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{10}$$

TABLE II. Details of UCI Imbalanced Dataset

Dataset	Instances	Positive	IR	Attributes
Pima	758	258	1.9	8
ecoli3	336	35	8.6	7
Cleveland	297	35	7.49	13
vehicle	846	199	3.25	19
page-blocks	5472	28	8.79	10
Breast	286	85	2.36	9
Abalone	4176	32	129.5	8

Here, TP denotes the true positives, TN denotes the true negatives, FP denotes the false positives, and FN denotes the false negatives. The G-mean evaluates how well the model performs across both positive and negative classes by considering their accuracies. Furthermore, F1-score is determined as the harmonic mean of the method's precision and recall [see Eq. (11)].

$$\text{F1}_{\text{score}} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{11}$$

AUC quantifies the overall ability of the model to discriminate between negative and positive classes, and is defined as the area under the receiver operating characteristic curve (ROC), which plots the True Positive Rate (TPR) against the False Positive Rate (FPR). AUC is used to evaluate the model performance. A higher AUC indicates a model with better discriminatory ability.

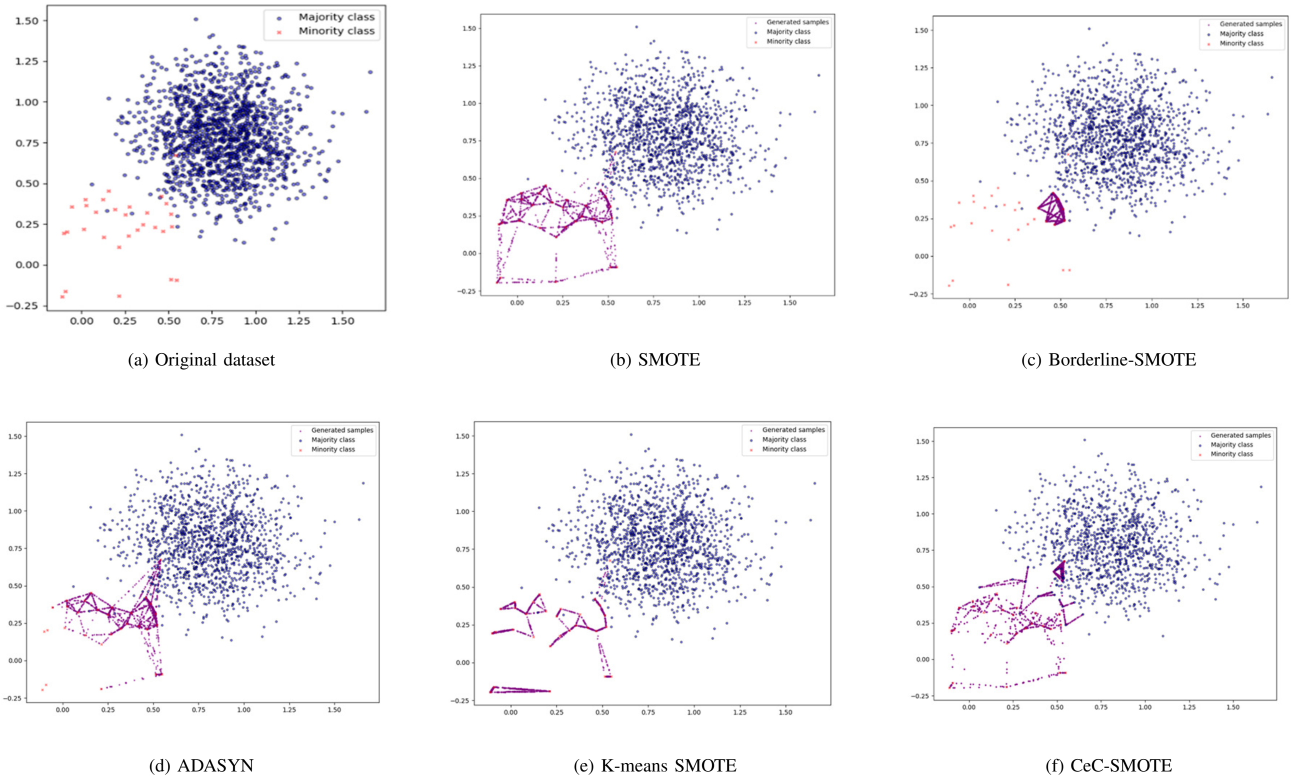
C. Comparison Methodology

In all experiments, CeC-SMOTE is compared with SMOTE and three variants of SMOTE, Borderline-SMOTE, ADASYN and K-Means SMOTE. The comparative methods involved in the experiments are all run with default parameters. The MLP is chosen as the classifier to evaluate the effectiveness of rebalancing on the balanced datasets that have been adjusted using the different comparison methods. The MLP implementation is available in the SciKit-Learn package is provided by the Python implementation, and configured with various layers and neurons using the ReLU activation function. The number of nearest neighbours K involved in all the compared algorithms is five in all experiments.

D. Experimental Results

1) The results on artificial datasets: To demonstrate the efficacy of CeC-SMOTE, we conduct a visualization experiment on artifical datasets. Fig. 2 and Fig. 3 presents the visualization outcomes of samples after being respectively rebalanced by Cec-SMOTE and other comparative methods, followed by dimensionality reduction using t-SNE. The blue, red, and purple points correspond to the negative, positive, and synthetic samples, respectively.

The figures show that the SMOTE algorithm treats all sample points equally. This results in new sample points being generated that overlap with the majority class. CeC-SMOTE first clusters the minority class samples, and then generates new samples near the cluster centroids. This results in the generation of concentrated, pink synthetic samples that adhere closely to the existing cluster structures of the minority class. The sampled data is more consistent with the original data distribution than data sampled using the SMOTE algorithm. Fig. 2(f) shows more minority class data points closer to the boundary than Fig. 2(e). This is because CeC-SMOTE oversample each cluster and the data at the boundary are more likely to be assigned to one cluster after clustering. Thus, the synthesised data are concentrated near the boundary, creating a clear distinction between the minority and majority classes.



(a) Original dataset

(b) SMOTE

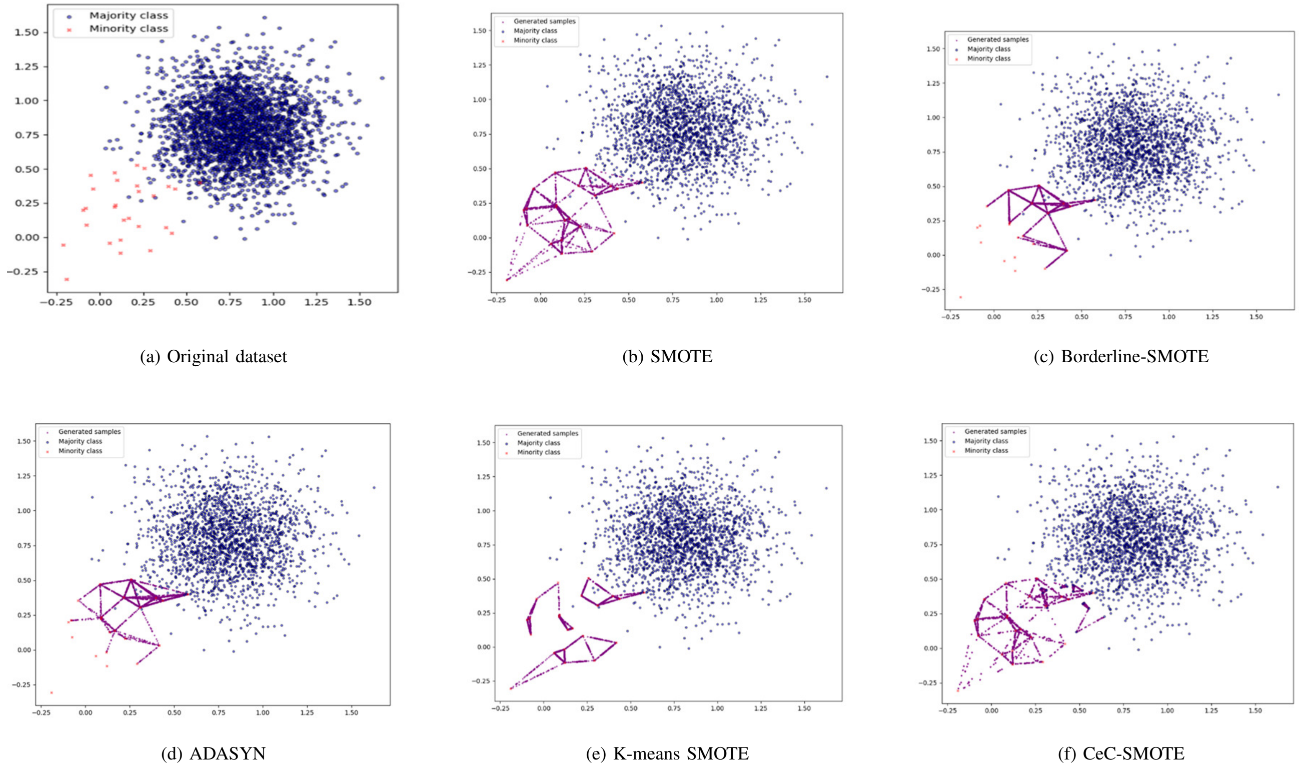
(c) Borderline-SMOTE

(d) ADASYN

(e) K-means SMOTE

(f) CeC-SMOTE

Fig. 2. Sampling effects of different methods on dataset A.



(a) Original dataset

(b) SMOTE

(c) Borderline-SMOTE

(d) ADASYN

(e) K-means SMOTE

(f) CeC-SMOTE

Fig. 3. Sampling effects of different methods on dataset B.

2) The results on UCl datasets: To evaluate the classification performance of the CeC-SMOTE algorithm, the experimental setup consists of seven datasets that present different imbalance ratios, which can be found in the UCI Machine Learning repository. The UCI datasets are used to conduct experiments on the MLP classifier and compared with the other four algorithms.

Table III shows the classification performance of various synthetic oversampling algorithms across multiple UCI datasets. The effectiveness of each method varies by dataset. For instance, in the vehicle dataset, all methods show high G-Mean and AUC, indicating robust performance. However, in datasets like Cleveland and Abalone, the variations in performance metrics are more pronounced, suggesting challenges related to specific characteristics of these datasets, such as feature distributions or class separability. In most cases, AUC scores are high, indicating good discriminative ability of the models post-oversampling. However, the F1-score and G-Mean sometimes show significant variation, highlighting the impact of these methods on precision-recall balance and class-specific accuracy. The performance of the CeC-SMOTE algorithm demonstrates a substantial enhancement in metric values compared to K-means SMOTE. Across the datasets, CeC-SMOTE often shows competitive or superior performance in terms of F1-score and AUC, suggesting its effectiveness in handling border cases and dataset specificities.

Fig. 4 presents a radar chart that compares the experimental results of CeC-SMOTE with those of the other four methods, providing more intuitive experimental results. The CeC-SMOTE line in the chart extends further in most datasets for the G-Mean and AUC metrics, suggesting that this technique might be more effective in distinguishing between classes.

E. Sensitivity Analysis of Critical Parameters

CeC-SMOTE involves several key parameters, including the number of clusters (k) in K-means, and the threshold coefficients α and β used to define "safe", "boundary", and "noise" regions within clusters. The number of clusters k influences the granularity of the minority class partitioning; too few clusters may overlook local data structure, while too many can result in overfitting or fragmented synthetic sampling. The parameters α and β directly affect which samples are classified as safe or boundary points, thereby controlling where new synthetic samples are generated.

Fig. 5 presents the effects of varying the parameters α and β on the F1 and G-mean scores across four datasets: Pima, ecoli3, vehicle, and page-blocks. Each plot illustrates how different (α, β) pairs influence classification performance, with F1 and G-mean used as evaluation metrics.

Across all datasets, the results show that the choice of α and β significantly impacts both F1 and G-mean. The most favorable performance is consistently observed when α is set between 0.65 and 0.80, and β is chosen to be 0.20 to 0.30 higher than α. In this range, both F1 and G-mean reach their peak or maintain stable, high values. This trend suggests that these parameter settings provide an optimal balance between generating a sufficient number of synthetic samples and maintaining the quality of those samples by targeting "safe" and "boundary" regions in the data.

When α is set too low, the algorithm becomes overly permissive, allowing many borderline points to be treated as "safe". This results in more synthetic data but can reduce overall precision due to the inclusion of ambiguous samples. Conversely, when α approaches 0.90, especially in datasets with small minority classes, recall tends to drop. This is likely because the "safe" region becomes too restrictive, limiting the creation of synthetic minority samples and, therefore, failing to sufficiently balance the dataset.

Notably, the (α, β) pair of (0.80, 1.20) performs among the best across all tested datasets, making it a reliable default setting when dataset-specific tuning is not practical. This combination provides a robust trade-off, consistently supporting both strong recall and precision.

For situations involving very scarce minorities or a high number of borderline samples, lowering α to 0.65 and increasing β by 0.10 to 0.20 further improves recall with only a minor reduction in precision. In contrast, for large and highly imbalanced datasets where high precision is more important than recall, setting α near 0.90 and β above 1.30 can be more advantageous, even if recall drops slightly.

In summary, Fig. 5 demonstrates that CeC-SMOTE is robust across a range of α and β values, with optimal performance achieved in moderate settings. The method's flexibility allows practitioners to tailor the balance between recall and precision according to the needs of specific datasets, while also providing a strong default setting for general use.

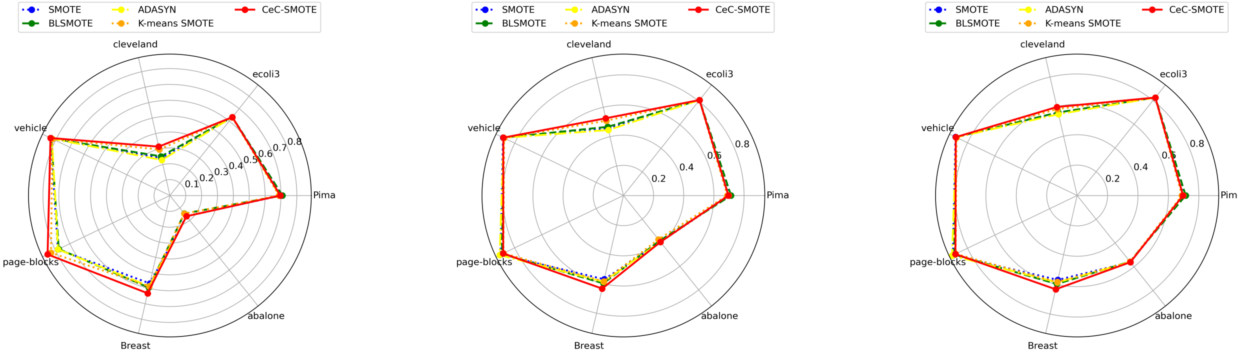
V. Discussion

The comparative results vary across datasets due to differences in characteristics such as imbalance ratio, feature count, class overlap, and minority class distribution. For example, datasets with well-separated classes and moderate imbalance, like "vehicle" or "page-blocks", typically show strong, consistent performance across most oversampling methods, including CeC-SMOTE. However, CeC-SMOTE excels in datasets with smaller minority classes or unclear class boundaries, such as "Cleveland" and "Abalone". Its clustering and noise filtering strategies help preserve meaningful minority samples while reducing the impact of outliers and class overlap.

CeC-SMOTE is particularly effective for datasets, where the minority class is well-clustered or where class boundaries are ambiguous. This observation is consistent with previous findings in the literature, where advanced oversampling techniques that incorporate clustering or density estimation have shown improved performance in handling complex imbalanced datasets. Recent works such as LD-SMOTE [9] and Cluster-Based Reduced Noise SMOTE [15] demonstrate that considering local density or clustering information can enhance the representativeness of synthetic samples and minimize noise. CeC-SMOTE builds upon these principles by combining centroid-guided clustering with adaptive sampling, resulting in synthetic data that better preserves local data structure and reduces class overlap. As shown in Table III and Fig. 4, CeC-SMOTE consistently achieves competitive or superior F1 and AUC scores, particularly in datasets with higher complexity, demonstrating its robustness and adaptability in real-world applications.

TABLE III. Metric Values for Different Imbalance Ratios

Dataset	Metric	SMOTE	BLSMOTE	ADASYN	K-means SMOTE	CeC-SMOTE
Pima	F1	0.6994	**0.7088**	0.6922	0.6881	0.6959
	G-Mean	0.7015	**0.7133**	0.6972	0.6891	0.6962
	AUC	0.7014	**0.7133**	0.6971	0.6891	0.6961
ecoli3	F1	0.6316	0.6316	0.6316	0.6300	**0.6316**
	G-Mean	0.8094	0.8094	0.8094	0.8084	**0.8094**
	AUC	0.8245	0.8245	0.8245	0.8234	**0.8245**
Cleveland	F1	0.2410	0.2529	0.2295	0.2975	**0.3158**
	G-Mean	0.4557	0.4670	0.4457	0.5085	**0.5522**
	AUC	0.5570	0.5609	0.5489	0.5841	**0.5976**
vehicle	F1	0.8340	0.8298	0.8252	0.8225	**0.8352**
	G-Mean	**0.8859**	0.8832	0.8806	0.8778	0.8836
	AUC	**0.8876**	0.8850	0.8825	0.8798	0.8855
page-blocks	F1	0.7833	0.7776	0.7827	0.8314	**0.8556**
	G-Mean	0.9043	0.8931	**0.9105**	0.8828	0.8848
	AUC	0.9068	0.8961	**0.9125**	0.8864	0.8881
Breast	F1	0.5685	0.5955	0.5865	0.5872	**0.6319**
	G-Mean	0.5699	0.5974	0.5865	0.5840	**0.6333**
	AUC	0.5698	0.5974	0.5865	0.5839	**0.6333**
abalone	F1	0.1672	0.1456	0.1672	0.1460	**0.1672**
	G-Mean	0.3940	0.3844	0.3940	0.3721	**0.3940**
	AUC	0.5612	0.5567	0.5612	0.5578	**0.5613**



(a) F1-score

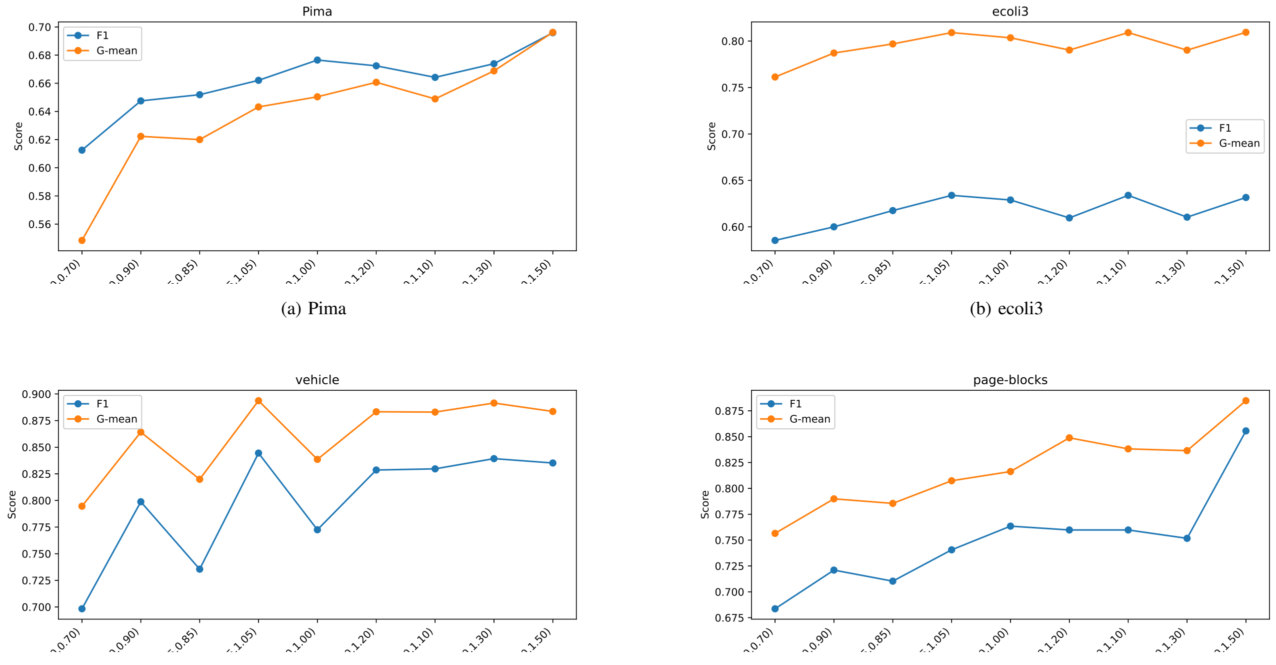
(b) G-Mean

(c) AUC

Fig. 4. Radar chart of experimental results of various algorithms.

CeC-SMOTE's robust performance can be attributed to its combination of centroid-guided clustering and adaptive sampling within safe and boundary regions. By focusing on generating synthetic data near cluster centroids, the method preserves local data structures and avoids excessive overlap with the majority class, which is a common pitfall in standard SMOTE. The visualizations on artificial datasets further demonstrate that CeC-SMOTE produces synthetic samples that align well with the minority class distribution, resulting in clearer class separations.

Parameter sensitivity analysis shows that CeC-SMOTE remains stable across a wide range of (α, β) values. The method performs best when α is between 0.65 and 0.80, with β exceeding α by 0.20 to 0.30. These results suggest that CeC-SMOTE can be deployed with minimal parameter tuning and still deliver reliable improvements in both recall and precision. Moreover, the recommended default setting $(\alpha = 0.80,\ \beta = 1.20)$ consistently ranked among the top configurations for most datasets. For datasets with extremely rare minorities, lowering α and widening β further enhances recall, while highly imbalanced datasets with abundant data



(a) Pima

(b) ecoli3

(c) vehicle

(d) page-blocks

Fig. 5. Effects of different α and β on performance.

benefit from higher α and β, favoring precision.

VI. Conclusion

In summary, CeC-SMOTE introduces a robust approach to class-imbalance learning by integrating K-means clustering, centroid-based neighbor selection, and safety-aware sampling. This combination enhances the representativeness of synthetic minority samples, preserves the geometric structure of the data, and reduces the risk of amplifying noise. Across multiple real-world and synthetic datasets, CeC-SMOTE demonstrates competitive or superior performance relative to established oversampling techniques, particularly in challenging scenarios with complex class boundaries. The method's stability across different parameter settings makes it practical for real-world deployment, requiring minimal fine-tuning for strong results.

While CeC-SMOTE demonstrates robust performance across multiple real-world and synthetic datasets, several limitations should be acknowledged to provide a more comprehensive review. First, the algorithm's effectiveness relies on the quality of clustering, which may be sensitive to the choice of cluster number and initial centroid selection. Inappropriate clustering could lead to suboptimal synthetic sample generation or noise amplification. Second, this study is limited to binary classification tasks. The adaptation of CeC-SMOTE to multi-class imbalanced datasets and its performance in high-dimensional feature spaces require further investigation. Finally, the current experiments utilize a limited set of benchmark datasets. Future research could extend CeC-SMOTE to multi-class and high-dimensional datasets, and explore automated parameter selection based on data characteristics to fully establish the generalizability of the approach. Overall, CeC-SMOTE offers a valuable and reliable solution for improving classifier performance on imbalanced data.

Acknowledgment

This research was supported by the Natural Science Foundation of Ningxia (grant number: 2024AAC03326).

References

[1] S. Alahyari and M. Domaratzki, Local distribution-based adaptive oversampling for imbalanced regression, vol. 1, no. 1. Association for Computing Machinery, 2025. [Online]. Available: http://arxiv.org/abs/2504.14316

[2] D. Annie, M. Swetha, and S. Sarawagi, “Synthetic Tabular Data Generation for Imbalanced Classification: The Surprising Effectiveness of an Overlap Class”.

[3] K. Kitova, “Improving Financial Distress Prediction through Clustered SMOTE,” vol. 5, no. 2, pp. 3663–3680, 2025.

[4] M. Altalhan, A. Algarni, and M. Turki-Hadj Alouane, “Imbalanced Data Problem in Machine Learning: A Review,” IEEE Access, vol. 13, no. January, pp. 13686–13699, 2025, doi: 10.1109/ACCESS.2025.3531662.

[5] H. Han, W. Y. Wang, and B. H. Mao, “Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning,” Lecture Notes in Computer Science, vol. 3644, no. PART I, pp. 878–887, 2005, doi: 10.1007/11538059_91.

[6] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, “Safe-level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem,” Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 5476 LNAI, pp. 475–482, 2009, doi: 10.1007/978-3-642-01307-2_43.

[7] D. T. Utari, ”Integration of SVM and SMOTE-NC for classification of heart failure patients,” BAREKENG: Jurnal Ilmu Matematika dan Terapan, vol. 17, no. 4, pp. 2263-2272, 2023.

[8] B. Fish, “SMOTE Variants for Imbalanced Binary Classification: Heart Disease Prediction,” vol. 2507, no. February, pp. 1–9, 2020.

[9] J. Lyu, J. Yang, and Z. Su, “LD-SMOTE: A Novel Local Density Estimation-Based Oversampling Method for Imbalanced Datasets,” pp. 1–21, 2025.

[10] O. Kachan, A. Savchenko, and G. Gusev, “Simplicial SMOTE: Oversampling Solution to the Imbalanced Learning Problem,” 2025, doi: 10.1145/3690624.3709268.

[11] S. He, H., Bai, Y., Garcia, E., & Li, “ADASYN: Adaptive synthetic sampling approach for imbalanced learning.,” IJCNN 2008.(IEEE World Congress on Computational Intelligence) (pp. 1322– 1328), no. 3, pp. 1322– 1328, 2008.

[12] G. Kovács, “An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets,” Applied Soft Computing Journal, vol. 83, p. 105662, 2019, doi: 10.1016/j.asoc.2019.105662.

[13] G. Douzas and F. Bacao, “Geometric SMOTE a geometrically enhanced drop-in replacement for SMOTE,” Inf Sci (N Y), vol. 501, pp. 118–135, 2019, doi: 10.1016/j.ins.2019.06.007.

[14] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique,” Journal of Artificial Intelligence Research, vol. 16, no. Sept. 28, pp. 321–357, 2002, [Online]. Available: https://arxiv.org/pdf/1106.1813.pdf

[15] J. Hemmatian, R. H. Id, and F. Nazari, “Addressing imbalanced data classification with Cluster-Based Reduced Noise SMOTE,” pp. 1–20, 2025, doi: 10.1371/journal.pone.0317396.

[16] D. Lee and H. Kim, “Adaptive Oversampling via Density Estimation for Online Imbalanced Classification,” Information , vol. 16, no. 1, pp. 1–17, 2025, doi: 10.3390/info16010023.

[17] S. Rezvani and X. Wang, ”A broad review on class imbalance learning techniques,” Applied Soft Computing, vol. 143, p. 110415, 2023.

[18] D. Elreedy, A. F. Atiya, and F. Kamalov, ”A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning,” Machine Learning, vol. 113, no. 7, pp. 4903–4923, 2024.

[19] Y. Chen, W. Pedrycz, and J. Yang, “A new boundary-degree-based oversampling method for imbalanced data,” pp. 26518–26541, 2023, doi: 10.1007/s10489-023-04846-4.

[20] X. Yuan, S. Chen, H. Zhou, C. Sun, and L. Yuwen, “CHSMOTE: Convex hull-based synthetic minority oversampling technique for alleviating the class imbalance problem,” Inf Sci (N Y), vol. 623, pp. 324–341, 2023, doi: 10.1016/j.ins.2022.12.056.

[21] S. Barua, M. M. Islam, X. Yao, and K. Murase, “MWMOTE - Majority weighted minority oversampling technique for imbalanced data set learning,” IEEE Trans Knowl Data Eng, vol. 26, no. 2, pp. 405–425, 2014, doi: 10.1109/TKDE.2012.232.

[22] A. Fern and S. Garc, “SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary,” vol. 61, pp. 863–905, 2018.

[23] W. Chen, K. Yang, Z. Yu, Y. Shi, and C. P. Chen, ”A survey on imbalanced learning: latest research, applications and future directions,” Artificial Intelligence Review, vol. 57, no. 6, p. 137, 2024.

[24] P. Soltanzadeh and M. Hashemzadeh, “RCSMOTE: range-controlled synthetic minority over- sampling technique for handling the class imbalance problem,” Inf Sci (N Y), 2020, doi: 10.1016/j.ins.2020.07.014.

[25] D. Elreedy, A. F. Atiya, and F. Kamalov, “A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning,” Mach Learn, no. 0123456789, 2023, doi: 10.1007/s10994-022-06296-4.

[26] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming, “K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data,” Inf Sci (N Y), vol. 622, pp. 178–210, 2023, doi: 10.1016/j.ins.2022.11.139.

[27] G. Douzas, F. Bacao, and F. Last, “Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE,” Inf Sci (N Y), vol. 465, pp. 1–20, 2018, doi: 10.1016/j.ins.2018.06.056.

[28] A. Alam, M. Muqeem, M. K. Ahamad, and K. O. Mohammed Aarif, “K-means clustering hybridized with nature inspired optimization algorithm: A review,” AIP Conf Proc, vol. 2935, no. 1, 2024, doi: 10.1063/5.0202041.

[29] F. Last, G. Douzas, and F. Bacao, “Oversampling for Imbalanced Learning Based on K-Means and SMOTE,” pp. 1–19, 2017, doi: 10.1016/j.ins.2018.06.056.

[30] S. Bhavani and N. Subhash Chandra, Histogram Based Initial Centroids Selection for K-Means Clustering, vol. 137. Springer Nature Singapore, 2023. doi: 10.1007/978-981-19-2600-6_38.

[31] J. Fonseca and G. Douzas, “Improving Imbalanced Land Cover Classification with K-Means SMOTE: Detecting and Oversampling Distinctive Minority Spectral Signatures,” 2021.

[32] A. Arafa, N. El-Fishawy, M. Badawy, and M. Radad, “RN-SMOTE: Reduced Noise SMOTE based on DBSCAN for enhancing imbalanced data classification,” Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 8, pp. 5059–5074, 2022, doi: 10.1016/j.jksuci.2022.06.005.

[33] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, “DB-SMOTE: Density-based synthetic minority over-sampling technique,” Applied Intelligence, vol. 36, no. 3, pp. 664–684, 2012, doi: 10.1007/s10489-011-0287-y.

[34] C. Bunkhumpornpat, E. Boonchieng, V. Chouvatut, and D. Lipsky, “FLEX-SMOTE: Synthetic over-sampling technique that flexibly adjusts to different minority class distributions,” Patterns, vol. 5, no. 11, p. 101073, 2024, doi: 10.1016/j.patter.2024.101073.

[35] J. G. B, A Novel Oversampling Technique for Imbalanced Learning Based on SMOTE and Genetic Algorithm. Springer International Publishing, 2021. doi: 10.1007/978-3-030-92238-2.

[36] L. Tao et al., “Newton cooling theorem-based local overlapping regions cleaning and oversampling techniques for imbalanced datasets,” Neurocomputing, vol. 616, no. November 2024, p. 128959, 2025, doi: 10.1016/j.neucom.2024.128959.

[37] L. J. Cohen, “On the use of neighbourhood-based non-parametric classifiers,” Philosophy, vol. 30, no. 112, pp. 7–14, 1955, doi: 10.1017/S0031819100036329.