Iran Journal of Computer Science (2025) 8:627–645
https://doi.org/10.1007/s42044-025-00240-0

RESEARCH

MK-SMOTE and M-SMOTE: enhanced techniques for handling class imbalance problem

Asifa Kanwal[1] · Nayyer Masood[2] · Ghulam Mustafa[1] · Maryam Abdul Ghafoor[3] · Samreen Ayaz[4]

Received: 6 December 2024 / Accepted: 3 February 2025 / Published online: 24 February 2025

Abstract

Class imbalance in datasets is a critical issue that leads to biased and misleading results in machine learning algorithms. To address this, two novel oversampling techniques are proposed: MK-SMOTE (Modified K-Means SMOTE) and M-SMOTE. MK-SMOTE minimizes noise by clustering minority class samples before generating synthetic data, while M-SMOTE balances the dataset through undersampling and iteratively creates synthetic instances based on calculated probabilities, retaining only those instances that are more likely to belong to the minority class. Both techniques were evaluated against K-Means SMOTE using G-Mean, $F1$-score, and accuracy as evaluation metrics across 22 benchmark datasets. The results show that MK-SMOTE and M-SMOTE consistently outperform K-Means SMOTE. For instance, with the KNN classifier, M-SMOTE achieved an average $F1$-score, G-Mean, and accuracy of 0.88, 0.89, and 0.89, respectively, compared to 0.87, 0.86, and 0.89 for MK-SMOTE, and 0.84, 0.83, and 0.86 for K-Means SMOTE. When using Logistic Regression, M-SMOTE yielded further improvements, with an average $F1$-score, G-Mean, and accuracy of 0.90, 0.90, and 0.91, respectively, demonstrating its effectiveness across multiple classifiers. Additionally, M-SMOTE achieved up to a 20% increase in $F1$-score and G-Mean for certain datasets, highlighting its robustness in addressing class imbalance. These results underscore the potential of the proposed methods to improve model performance and mitigate noise in imbalanced datasets.

Keywords Class imbalance · Clustering · Classification · Oversampling · Under-sampling · SMOTE

✉ Ghulam Mustafa
ghulam.mustafa.ssc@stmu.edu.pk

Asifa Kanwal
Asifa.ssc@stmu.edu.pk

Nayyer Masood
nayyer@cust.edu.pk

Maryam Abdul Ghafoor
maryam.ghafoor@lums.edu.pk

Samreen Ayaz
samreen.ayaz@gmail.com

[1] Department of Computing, Shifa Tameer-e-Millat University, Islamabad 44000, Pakistan

[2] Department of Computing, Capital University of Science and Technology, Islamabad 44000, Pakistan

[3] Department of Computer Science, Lahore University of Management Sciences, Lahore 54770, Pakistan

[4] Department of Computer Science, Mirpur University of Science and Technology (MUST), Mirpur 10250, Pakistan

1 Introduction

The proliferation of data across diverse domains has led to the accumulation of massive volumes of information [1–3]. Fields, such as medicine, education, and logistics, frequently handle extensive datasets in various formats [4]. Data science plays a pivotal role in managing and analyzing this information, enabling the extraction of meaningful insights through conventional machine learning techniques. However, the quality of the input data significantly impacts the performance of these algorithms, as raw and uncleaned data from multiple sources often hinder optimal results [5, 6].

An additional challenge arises from the class imbalance problem, a prevalent issue in real-world applications such as fraud detection, medical diagnosis, network intrusion detection, and fault detection [7–10]. In these domains, class imbalance occurs when data are unevenly distributed among classes, with a majority of instances belonging to one class (majority class) and far fewer instances to another (minority class). This imbalance is critical, because classifiers trained on imbalanced datasets tend to be biased toward the majority class, resulting in poor performance for the minority class. In applications like medical diagnosis or fraud detection, where the minority class represents rare but significant events (e.g., disease occurrences or fraudulent activities), misclassifying these events can lead to severe consequences [11–15]. Consequently, addressing class imbalance is not just an algorithmic challenge but a fundamental step in ensuring the effectiveness of these systems in critical real-world applications.

Existing solutions to the class imbalance problem typically fall into two main categories: data-level and algorithm-level approaches. Data-level techniques aim to balance the dataset by either oversampling the minority class or undersampling the majority class [16–19]. Oversampling involves generating synthetic instances of the minority class, while undersampling reduces instances of the majority class. However, undersampling may result in valuable data loss, making oversampling a more commonly preferred solution [11, 20–23]. Despite their advantages, traditional oversampling methods such as SMOTE and its variants exhibit limitations, especially when clusters contain a high proportion of majority class instances. In such cases, synthetic samples may resemble the majority class, leading to noise and degraded performance [11, 24]. Algorithm-level approaches, on the other hand, modify classifiers to better handle imbalanced datasets [25, 26]. While these techniques offer improvements, they often do not fully address the risk of noisy synthetic instances. This research focuses on data-level approaches, specifically proposing two novel oversampling techniques: Modified K-Mean SMOTE (MK-SMOTE) and Modified SMOTE (M-SMOTE). These methods aim to address the limitations of existing oversampling techniques. MK-SMOTE improves upon K-Mean SMOTE by clustering only the minority class, thus reducing the risk of generating synthetic samples that are mistakenly labeled as minority class when they are, in fact, more similar to the majority class. M-SMOTE introduces a noise-reduction mechanism by selecting minority class instances based on a defined threshold and iteratively computing the inclusion probability of newly generated samples, which helps ensure that synthetic instances are more accurate and representative.

The key contributions of this research are as follows:

- Modified K-Mean SMOTE (MK-SMOTE): This method addresses the limitations of K-Mean SMOTE by clustering only the minority class. This approach creates a safer environment for generating synthetic samples, reducing the risk of noisy or mislabeled instances.
- Modified SMOTE (M-SMOTE): To further mitigate the impact of noisy samples, this technique selects minority class instances based on a defined threshold and computes the inclusion probability of newly generated samples over multiple iterations before finalizing them.
- The proposed MK-SMOTE and M-SMOTE algorithms were rigorously evaluated using publicly available datasets, demonstrating superior performance compared to K-Mean SMOTE. Both techniques achieved over 90% accuracy in classification tasks, showcasing their effectiveness in addressing class imbalance.

The hypothesis underlying this research posits that the strategic modification of oversampling techniques, specifically through targeted clustering and noise-reduction mechanisms, can significantly improve classification performance on imbalanced datasets.

The experimental evaluation utilized 12 datasets from the UCI Machine Learning Repository, 8 datasets from the KEEL repository, and 2 artificial datasets generated using scikit-learn, as shown in Table 1. These datasets were selected to offer a diverse set of data types and characteristics, ensuring a comprehensive evaluation of the proposed oversampling techniques. The original multi-class labels of these datasets were transformed into binary labels using the one-versus-one approach, allowing for a consistent comparison across datasets. Evaluation metrics, including $F1$-score, G-Mean, and accuracy, demonstrated the effectiveness of the Modified K-Mean SMOTE (MK-SMOTE) and Modified SMOTE (M-SMOTE) approaches in enhancing prediction accuracy compared to K-Means SMOTE. The results revealed that both MK-SMOTE and M-SMOTE outperformed or matched K-Means SMOTE across datasets, with notable improvements in $F1$-score and accuracy. The synthetic instances generated by these methods reduced noise and bias toward the majority class, leading to better classifier performance. These results emphasize the potential of the proposed techniques in addressing the class imbalance problem across various domains.

The remainder of this paper is organized as follows: Sect. 2 reviews related work, Sect. 3 details the proposed methodology, Sect. 4 presents the experimental evaluation, Sect. 5 discusses the findings in the context of existing research, and Sect. 6 concludes the paper with future research directions.

2 Literature

The class imbalance problem has garnered significant attention in recent years, as it can severely impact the performance of machine learning models. Various techniques have been proposed to address this issue, with sampling-based approaches emerging as particularly effective. Sampling methods, particularly oversampling, have proven beneficial, as they allow traditional classifiers to be applied once the dataset is balanced. One widely used technique is the Synthetic Minority Oversampling Technique (SMOTE) [27, 28], which generates synthetic data by creating new instances of

Table 1 Datasets from UCI [11, 20, 31, 35] (Lin et al. [40]; Rayhan et al. [41]; Sikora and Raina [42]; Bellinger et al. [43]) and KEEL [21, 22, 27, 29, 32–34] (Han et al. [44]; Mullick et al. [45]; O'Neil and Schutt [46]; Thabtah et al. [47]; Yun et al. [48]) repository

Sr #	Dataset	No. of features	Total samples	Minority samples	Majority samples	Imbalance ratio
1	Breast tissue	9	106	36	70	1.94
2	cleveland-0	13	173	13	160	12.31
3	dermatology-6	34	358	20	338	16.90
4	Ecoli	7	336	52	284	5.46
5	Eucalyptus	8	642	98	544	5.55
6	Glass	9	214	70	144	2.06
7	Haberman	3	306	81	225	2.78
8	Heart	13	270	120	150	1.25
9	Iris	4	150	50	100	2.00
10	led7digit	7	443	37	406	10.97
11	Libra	90	360	72	288	4.00
12	Liver	6	345	145	200	1.38
13	new-thyroid1	5	215	35	180	5.14
14	new-thyroid2	5	215	35	180	5.14
15	page-blocks-1	10	472	28	444	15.8
16	Pima	8	768	268	500	1.87
17	Vehicle	18	846	199	647	3.25
18	vowel0	13	988	90	898	9.98
19	Wine	13	178	71	107	1.51
20	yeast1	8	1484	429	1055	2.46
21	Artificial 1	3	500	100	400	4
22	Artificial 2	3	500	100	400	4

the minority class rather than replicating existing ones. This is achieved by selecting a random instance from the minority class and one of its nearest-neighbor instances, generating new samples through linear interpolation. While SMOTE effectively mitigates the overfitting problem encountered with random sampling, it still suffers from the within-class imbalance issue. Additionally, studies have highlighted that SMOTE can lead to problems such as shrinkage of covariance [29, 30] and the risk of introducing synthetic instances that exhibit the characteristics of the majority class, which can bias model predictions in favor of the majority [31]. Despite these drawbacks, SMOTE remains a cornerstone in imbalance handling, with many advanced techniques built on top of it. These limitations can be addressed by combining SMOTE with other preprocessing methods.

An extension of SMOTE [32] aims to improve the selection of nearest neighbors by rejecting poorly chosen instances, which can lead to overfitting or underfitting. This modified approach generates synthetic data while assigning a rejection level based on the proximity of minority instances to their neighbors. If the rejection level is low, the synthetic instance is discarded, improving the overall quality of the synthetic samples and yielding improved G-Mean values.

Deterministic SMOTE (SMOTE-D) [31] builds upon SMOTE by repeatedly applying the oversampling procedure to generate more accurate synthetic samples. While random synthetic generation may introduce variability, SMOTE-D uses a deterministic approach, producing consistent and effective results. Another approach, the Automatic Neighborhood Size Determination (AND) [31], improves upon SMOTE by limiting the number of neighbors considered to better preserve the underlying data distribution. AND has shown superior performance compared to SMOTE, ADASYN, and Borderline-SMOTE, particularly when dealing with small sub-clusters and complex patterns in the data.

In addition to oversampling, other studies have integrated noise filtering techniques to improve the quality of the synthetic data. For instance, Kang et al. [33] proposed combining noise filtering with undersampling by focusing on the minority class for noise filtering while undersampling the majority class. However, their method still suffered from noise issues in the majority class, as it focused solely on minority class noise filtering. In another approach [22], a random forest classifier was used to remove noisy instances before incorporating them into the minority class. While effective under balanced conditions, this technique struggled with highly imbalanced datasets, resulting in misclassification due to bias.

Dina Elreedy et al. [29] conducted a mathematical survey of SMOTE, analyzing the distribution characteristics of synthetic samples and assessing how well they approximate the actual distribution. This foundational work provides insights into the quality of SMOTE-generated data, which is crucial for ensuring that synthetic instances are representative of the minority class. Another innovative technique, proposed by [34], utilizes an evolutionary algorithm (EA) to enhance cluster-based oversampling. This method optimizes the data generation process while reducing computational costs, making it more efficient for large-scale datasets.

The Adaptive Neighbor Synthetic Minority Oversampling Technique (ANS) [21] dynamically adjusts the number of neighbors used in oversampling, depending on the local density of the data. By refining the selection of neighbors, ANS aims to improve classifier accuracy. However, it may lead to overfitting when dealing with outliers or rare cases, which are typically referred to as outcaste minorities. These instances are surrounded by negative examples, making it difficult for the classifier to learn effectively, ultimately resulting in overfitting.

Sharma et al. [35] pointed out that existing oversampling methods typically focus only on the minority class, which may be ineffective when the imbalance ratio is extremely high. They proposed a technique that also incorporates information from the majority class, using Mahalanobis distances to generate synthetic instances. Their approach showed significant performance improvements when dealing with extreme imbalance ratios but is less effective for moderate imbalances.

The Noise Removal and Oversampling Method (NROMM) [36], introduced by Liu et al., incorporates noise removal within clusters of data points. Using adaptive embedding and secure boundaries, NROMM ensures that synthetic data are generated within well-defined clusters, improving the stability and quality of the oversampling process. Liu's approach, tested across 20 benchmark datasets, showed promising results in reducing noise and improving classification accuracy.

Wongvorachan et al. [37] compared various resampling techniques, including random oversampling, random undersampling, and SMOTE-NC, to address class imbalance in a dataset from the High School Longitudinal Study of 2009. They found that random oversampling performed well for moderately imbalanced data, while hybrid resampling methods, combining SMOTE-NC with random undersampling, were more effective for highly imbalanced datasets.

Feng et al. [38] proposed the Negative Binary General (NBG) technique, which combines oversampling with feature selection to improve classification performance. Using the Binary Ant Lion Optimizer to select relevant features and generating improved synthetic instances, this method demonstrated better results across seven datasets using various classifiers.

Finally, Douzas et al. [20] introduced K-means SMOTE, which combines clustering with SMOTE to prevent noisy instance generation. By clustering the data using K-means, filtering these clusters, and applying SMOTE to the selected clusters, this technique improves the quality of synthetic data and reduces the impact of imbalance within clusters. Their results, evaluated across 12 imbalanced datasets, showed superior performance compared to other oversampling methods.

2.1 Critical analysis

While a wide range of techniques has been developed to address the class imbalance problem, many of these methods still face significant limitations. Issues such as the risk of overfitting, the generation of synthetic instances that do not adequately represent the minority class, and the challenges of handling extreme class imbalances remain unresolved in several approaches. Although SMOTE and its extensions have proven effective in many scenarios, they are not without drawbacks, such as the potential for generating synthetic instances that disproportionately resemble the majority class or fail to capture complex data patterns. Furthermore, some techniques, such as ANS and NROMM, are prone to overfitting when dealing with outliers or rare instances. Despite these shortcomings, a universally effective solution that balances synthetic instance generation with noise reduction and classifier stability remains elusive. This work aims to address these challenges by proposing a novel approach that combines the strengths of existing techniques while mitigating their weaknesses, particularly for datasets exhibiting complex structures and extreme class imbalances.

3 Methodology

This study presents two approaches, MK-SMOTE and M-SMOTE, to address the data imbalance problem. The following subsections provide a detailed discussion of both methodologies.

3.1 MK-SMOTE

MK-SMOTE builds upon K-Means SMOTE, an oversampling technique that utilizes the K-means clustering algorithm. Unlike K-Means SMOTE, MK-SMOTE focuses specifically on creating clusters from the minority class, making it more targeted in addressing class imbalance. The method operates in three main stages: Clustering, Filtering, and Oversampling, as illustrated in Fig. 1.

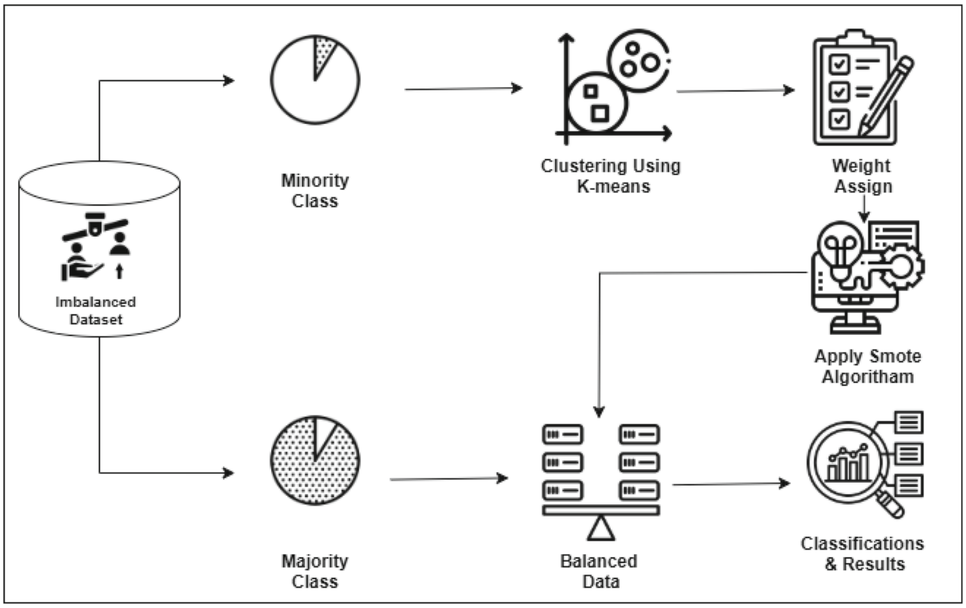


Fig. 1 Overview of the MK-SMOTE technique, illustrating the three main stages: clustering of minority class instances using K-means, filtering clusters based on their minority density, and oversampling selected clusters to generate synthetic instances. The method applies K-means clustering exclusively to the minority class and adjusts oversampling according to cluster density and sparsity, with the goal of improving class balance

3.1.1 Stage 1: clustering the minority class

The first step in MK-SMOTE is to apply K-means clustering to the minority class instances. This involves partitioning the minority class data into k clusters. K-means clustering works by iterating through the following two steps:

- **Assignment step:** Each sample is assigned to the nearest cluster center.
- **Update step:** The cluster centers are updated based on the new assignments, i.e., they are recalculated as the mean of the instances in the cluster.

In this case, only the minority class instances are considered, resulting in k clusters of minority class samples. The number of clusters, k, is an important hyperparameter that must be selected based on the dataset. This choice can be tuned by evaluating the algorithm's performance using cross-validation or other clustering evaluation metrics. A key novelty of MK-SMOTE is its exclusive application of clustering to the minority class, which prevents the introduction of noise from the majority class-a common issue in traditional SMOTE methods.

3.1.2 Stage 2: filtering the clusters

Once clustering is complete, the next step is to filter the clusters. This process identifies which clusters should undergo oversampling based on their minority density. A key factor in this selection is the density of minority samples within each cluster. To quantify this, the average minority distance (AMD) is computed for each cluster, representing the average Euclidean distance between all pairs of instances within the cluster. Next, the density factor (DF) is calculated for each cluster. The density factor is inversely proportional to the average minority distance raised to a user-defined exponent (de) and accounts for the number of minority instances in the cluster. This formulation helps assess the sparsity of the minority class within each cluster, assigning higher weights to denser clusters for oversampling. The sparsity factor for each cluster is then derived as the inverse of its density factor, prioritizing clusters with fewer samples and greater inter-instance distances. Clusters for oversampling are selected based on this sparsity factor, with sparser clusters receiving higher priority. The novelty of this approach lies in its density-based selection of clusters for oversampling, ensuring that synthetic instances reflect the underlying minority class distribution while reducing the risk of generating noisy samples in denser regions where class imbalance is less severe.

3.1.3 Stage 3: oversampling using SMOTE

In the oversampling stage, synthetic data are generated using the Synthetic Minority Oversampling Technique (SMOTE). The number of synthetic samples for each cluster is determined by its sampling weight, which is computed based on the sparsity factors of all clusters. This ensures that clusters with higher sparsity (fewer minority samples) receive more synthetic instances. For each selected cluster, SMOTE is applied by generating synthetic samples along the line segments between existing minority instances and their nearest neighbors. The number of synthetic samples created is proportional to the cluster's sampling weight. This sparsity-based oversampling approach enhances the representativeness of the minority class while reducing noise by prioritizing regions where class imbalance is most pronounced. This marks a significant improvement over traditional SMOTE, which applies oversampling uniformly across the entire minority class without considering data density. At the end of this process, the dataset achieves a more balanced class distribution, with an increased number of minority class instances generated through synthetic sampling.

Algorithm 1 Oversampling with MK-SMOTE

```
begin
  Step 1: Cluster the minority instances
  clusters ← kmeans($X_{min}$)
  Step 2: For each cluster, compute the sampling weight based on its minority density.
  for each $f \in$ all clusters do
    amd($f$) ← mean(euclidean distances($f$))
    df($f$) ← $\frac{\text{minority count}(f)}{(\text{average minority distance}(f))^{de}}$
    sparsity factor($f$) ← $\frac{1}{\text{density factor}(f)}$
  end for
  $ss \leftarrow \sum_{f \in \text{filtered clusters}}$ sparsity factor($f$)
  sampling weight($f$) ← $\frac{\text{sparsity factor}(f)}{\text{sparsity sum}}$
  Step 3: Oversample each cluster using SMOTE. The number of samples to be generated is computed using the sampling weight.
  generated samples ← ∅
  for each $f \in$ clusters do
    $ns \leftarrow \lfloor n \times \text{sampling weight}(f)k \rfloor$
    generated samples ← generated samples ∪ {SMOTE($f$, number of samples, knn)}
  end for
  return generated samples
end
```

The equations used in MK-SMOTE are designed to evaluate and guide the oversampling process. The key equations involve calculating the average minority distance (amd), the density factor (df), and the sparsity factor for each cluster. These equations serve as essential components for determining which clusters will be prioritized for oversampling.

(1) Average minority distance (amd):

$$amd(f) \leftarrow mean(euclidean\_distances(f)). \tag{1}$$

The average minority distance quantifies the compactness of a cluster. If the distance between instances in a cluster is large, it indicates that the cluster is sparse, meaning that it has fewer samples that are far apart. This measure helps assess the density of the minority class in each cluster. A smaller value for amd implies a denser cluster.

(2) Density factor (df):

$$df(f) \leftarrow \frac{minority\_count(f)}{(average\_minority\_distance(f)).} \tag{2}$$

The density factor helps us understand how "dense" a cluster is in terms of the minority class. It is based on the number of minority instances within the cluster and the distance between these instances. A lower amd leads to a higher df, indicating a denser cluster. The exponent de is used to tune the importance of distance in the density calculation, allowing us to control how sensitive the density is to spatial distribution.

(3) Sparsity factor (sparsity factor):

$$sparsity\_factor(f) \leftarrow \frac{1}{density\_factor(f)}. \tag{3}$$

The sparsity factor inversely reflects the density of the cluster. If a cluster has a high *df*, it means that it is dense and requires fewer synthetic samples. Conversely, a lower df (sparser cluster) results in a higher sparsity factor, signaling that more synthetic samples should be generated to balance the class distribution.

The sampling weight for each cluster is calculated based on the sparsity factor relative to the sum of all sparsity factors across the selected clusters. The larger the sparsity factor, the greater the weight, meaning that the algorithm will prioritize oversampling for sparser clusters.

Furthermore, the key parameters in MK-SMOTE are k, the number of clusters; de, the exponent for density calculation; knn, the number of nearest neighbors in SMOTE; and irt, the imbalance ratio threshold. The number of clusters (k) determines the granularity of clustering, with higher values leading to finer granularity but potentially increased computational cost and overfitting. de influences the density factor's sensitivity to average minority distances, where higher values emphasize the spatial distribution of sparse clusters, and it is typically tuned empirically to improve class balance. knn defines how many nearest neighbors are considered in SMOTE, balancing diversity, and noise in synthetic sample generation, with common values ranging from 5 to 10. Although it is not directly used in the algorithm, it serves as a guideline for determining when oversampling should be applied based on the class imbalance ratio, with its value depending on the dataset.

3.2 M-SMOTE

The M-SMOTE (Oversee SMOTE) method proposes a novel approach to oversampling by integrating a Naïve Bayes classifier into the resampling process, distinct from traditional methods that rely on clustering (as presented in Fig. 2). Instead of using clustering to generate synthetic data, M-SMOTE focuses on creating synthetic samples based on probability estimates derived from the Naïve Bayes classifier. This method leverages the classifier to assess how likely a synthetic instance is to belong to the minority class, ensuring that the generated instances are more representative and less biased. The steps of the M-SMOTE method involve training a Naïve Bayes model on a balanced dataset, generating synthetic instances, and then checking their probability of belonging to the minority class. This ensures that only instances most likely to be classified as the minority class are added. The probability of each instance is checked against predefined thresholds, thus reducing the risk of overfitting or bias caused by the majority class. Furthermore, training the Naïve Bayes model iteratively on newly generated synthetic instances allows the model to integrate these instances into the learning process, reducing bias and enhancing overall performance.

Tuning the M-SMOTE method: The performance of M-SMOTE depends on several key parameters that can be tuned for each dataset:

- Threshold range for minority class ((T_{low}, T_{high})): The thresholds determine which synthetic instances are retained and added to the minority class. A range between 0.5 and 1.0 is commonly used to ensure that the synthetic instances have a high probability of belonging to the minority class, thus ensuring fairness. The exact values for these thresholds can be adjusted based on the imbalance ratio and the specific characteristics of the dataset. For instance, datasets with severe imbalance may require tighter thresholds to avoid generating synthetic instances that overly resemble the majority class.
- Number of iterations for probability calculation ((N)): The method involves iterating the training and probability calculation N times. By default, N is set to 5, meaning that the Naïve Bayes model is retrained on the new data five times to minimize the bias introduced by any one iteration. This helps create more diverse and accurate synthetic instances. The number of iterations can be fine-tuned based on dataset size and complexity. For larger datasets, fewer iterations might suffice, while for highly imbalanced datasets, more iterations may help improve the accuracy of synthetic data.
- Random selection of majority class instances ((S_{maj})): The selection of majority class instances should be done randomly to ensure a diverse training set. The number of majority class instances selected should equal the number of minority class instances, which ensures that the dataset is balanced after oversampling.

The key advantage of the M-SMOTE method is its dynamic and adaptive nature. Using Naïve Bayes to generate synthetic instances and iterating the process, M-SMOTE ensures that the newly generated data are as representative of the minority class as possible. This process reduces the risk of generating synthetic instances that are biased toward the majority class or are outliers.

Algorithm 2 Oversampling with M-SMOTE

```
Require:
$X$ : Matrix of all observations
$AX$ : Matrix of observations after oversampling
$y$ : Random value ranging from 0 to 1
$RS$ : Number of required samples to be generated
$X_{min}$ : Number of minority samples
$AX_{min}$ : Number of minority samples after oversampling
$S_{maj}$ : Number of selected majority samples
$T_{low}, T_{high}$ : Lower and upper threshold values
$x_i^{min}, x_j^{min}$ : Randomly selected minority instances from $X_{min}$
$X_{newmin}$ : New generated instances
$C_{model}$ : Classifier model
$AP$ : Average probability
begin
while $|AX_{min}| < RS$ do
    for $i = 1$ to $N$ do
        $S_{maj} \leftarrow$ Random selection($X$) ▷ Selection of majority class
      equal to minority class
        $C_{model} \leftarrow$ Train($S_{maj} \cup X_{min}$)
        $X_{newmin} \leftarrow x_i^{min} + y \cdot (x_i^{min} - x_j^{min})$   ▷ Synthesize new
      instance
        $P(X_{newmin}) \leftarrow P(X_{newmin}) +$ Predict($C_{model}, X_{newmin}$)
    end for
    AP of $X_{newmin} \leftarrow \frac{P(X_{newmin})}{N}$   ▷ Compute average probability
    if AP of $X_{newmin} \in [T_{low}, T_{high}]$ then
        $AX_{min} \leftarrow AX_{min} \cup \{X_{newmin}\}$
    end if
end while
$AX_{min} \leftarrow AX_{min} \cup X_{min}$
$AX \leftarrow AX_{min} \cup S_{maj}$
return $AX$
end
```

The intuition behind the key equations used in the M-SMOTE algorithm stems from the need to generate synthetic samples for minority classes in a controlled manner while preserving class boundaries. Specifically, the equation $X_{newmin} \leftarrow x_i^{min} + y \cdot (x_i^{min} - x_j^{min})$ inspired by the Synthetic Minority Oversampling Technique (SMOTE). It ensures the generation of new instances X_{newmin} along the line segment connecting two randomly selected minority samples x_i^{min} and x_j^{min} where y is a random value in the range [0, 1]. This approach maintains diversity among generated samples while ensuring they remain within the minority class distribution. Additionally, the use of a classifier model (C_{model}) to predict the probability of the generated samples being classified as the minority class is crucial for ensuring that the synthetic samples are valid. The average probability (AP) T_{low} and T_{high} are added to the final dataset.

The values for key parameters in the algorithm, such as the lower and upper thresholds (T_{low} and T_{high}), were chosen based on experimental validation to ensure a balance between diversity and validity of the generated samples. For instance, T_{low} is set to ensure that generated samples are not too close to the decision boundary, reducing the risk of misclassification, while T_{high} avoids including overly conservative samples that limit diversity. The number of majority samples selected (S_{maj}) is kept equal to the number of minority samples (X_{min}) to maintain class balance during training,

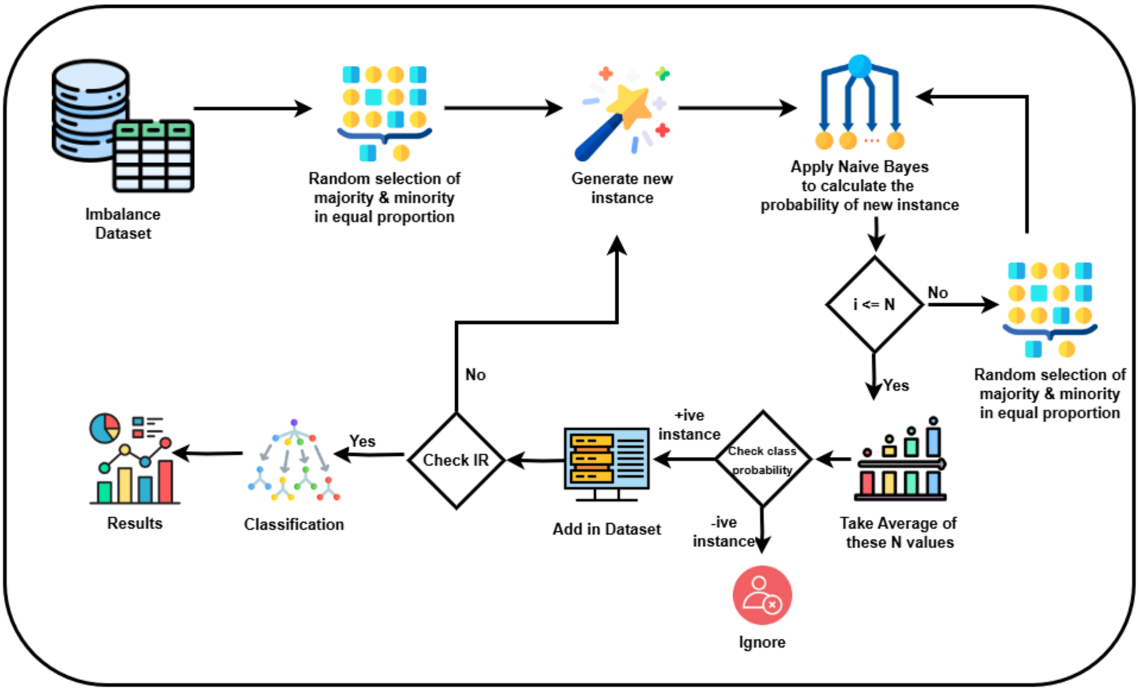

Imbalance Dataset
Random selection of majority & minority in equal proportion
Generate new instance
Apply Naive Bayes to calculate the probability of new instance
i <= N
No
Random selection of majority & minority in equal proportion
Yes
Take Average of these N values
Check class probability
+ive instance
-ive instance
Ignore
Add in Dataset
Check IR
No
Yes
Classification
Results

Fig. 2 Illustration of the M-SMOTE technique for minority class oversampling. The process involves selecting two minority samples ($x_i^{min} - x_j^{min}$) and generating synthetic samples (X_{newmin}) along the line segment between them using a random value (y). A classifier model (C_{Model}) is trained on a balanced dataset of majority (S_{maj}) and minority samples (X_{min}) to validate the generated samples. Synthetic samples are included in the augmented dataset AX_{min} if their average probability (AP) of belonging to the minority class falls within predefined threshold values (T_{low} and T_{high}). This ensures diversity and validity of the generated samples while preserving class boundaries

which is a critical assumption for the proper functioning of the classifier. Furthermore, the choice of random values (y) within [0, 1] aligns with SMOTE's strategy to distribute new samples uniformly between the selected instances.

Furthermore, in concluding this algorithm, it is important to highlight that M-SMOTE introduces a probabilistic approach to further reduce noise and enhance the effectiveness of synthetic oversampling. Unlike traditional SMOTE, which generates synthetic instances based solely on nearest-neighbor information, M-SMOTE calculates the inclusion probability of each new synthetic sample over multiple iterations. This allows for a more refined selection of synthetic instances, ensuring that the newly generated data points are more likely to represent meaningful minority class instances. In M-SMOTE, a thresholding mechanism is applied to filter out potentially noisy synthetic samples based on a user-defined probability threshold. This probabilistic filtering improves the quality of the generated samples, retaining only those with a high likelihood of being true minority class instances. As a result, the synthetic instances produced by M-SMOTE are less likely to introduce noise and better represent the true minority class distribution. The novelty of M-SMOTE lies in its probabilistic validation of synthetic instances, ensuring that the generated data are more reliable and positively contributes to balancing the dataset.

4 Experimental settings

The primary objective of oversampling is to improve the prediction accuracy of the classifier. All experiments were conducted on a Core i5 system running Ubuntu with 8GB of RAM. Well-defined evaluation metrics, datasets, classifiers, and oversampling techniques were employed. To ensure unbiased results, fivefold cross-validation was applied to each dataset. For each evaluation metric, the mean of the five values obtained across the iterations of the fivefold process was calculated. This section provides a detailed description of these components.

4.1 Evaluation metrics

The choice of evaluation metrics is primarily driven by the specific objective, especially when dealing with imbalanced datasets. In certain applications, such as medical diagnosis, false negatives may be more critical than false positives. However, when assessing oversampling techniques, such prioritization should not be applied. Therefore, four key evaluation metrics were selected: G-Mean, $F1$-score, Accuracy, and Area Under the Curve (AUC). Each metric was chosen for its relevance and ability to provide insights into different aspects of model performance in the context of class imbalance.

- **G-Mean:** This metric is particularly valuable for imbalanced datasets as it balances sensitivity and specificity by computing their geometric mean. It ensures that the model performs well across both the minority and majority classes, preventing a bias toward the majority class.
- **F1-score:** The $F1$-score, being the harmonic mean of precision and recall, was chosen for its utility in evaluating models where class imbalance is significant. It provides a balanced evaluation of the model's ability to correctly identify positive instances without generating excessive false positives.
- **Accuracy:** While accuracy is commonly used as a general performance metric, its usefulness in imbalanced datasets is often limited. However, it still serves as a useful baseline metric to evaluate overall classification performance, though it should be interpreted with caution in such contexts.
- **Area under the curve (AUC):** AUC is an important metric for imbalanced datasets, because it evaluates the model's ability to distinguish between classes at various decision thresholds. It captures the trade-off between true-positive and false-positive rates, offering a more nuanced view of model performance than accuracy alone.

Additionally, the approach was compared with the existing SMOTE technique, which also utilizes these metrics, ensuring a fair and consistent comparison. Classification metrics assess predicted outcomes against actual class labels, and a confusion matrix can be constructed to summarize these results, including True Positives, False Positives, False Negatives, and True Negatives.

4.1.1 Classifiers

Two classifiers, Logistic Regression (LR) and K-Nearest Neighbors (KNN), were selected to evaluate the performance of the oversampling techniques, MK-SMOTE and M-SMOTE. This choice ensures that the effectiveness of the oversampling methods is not restricted to a specific classifier. LR is a statistical model used for binary classification, where it models the probability of the target class. It serves as a baseline for many other approaches and provides easily reproducible results. KNN, on the other hand, classifies instances based on the class of their K-nearest neighbors, assigning the label of the majority class among them. For instance, when classifying an instance 'a' as either malignant or benign, KNN determines its class by examining the labels of 'a's K-neighbors. If the majority of the neighbors are malignant, KNN will classify 'a' as malignant. Since these classifiers are commonly used with SMOTE, employing them allows for a fair comparison of the methods with existing techniques.

4.1.2 Dataset

Twelve datasets were selected from the UCI Machine Learning Repository, eight from the KEEL repository, and two artificial datasets generated using scikit-learn, as shown in Table 1. To ensure an unbiased and fair evaluation of the methods, the same 20 datasets used by K-means SMOTE [20, 39] were employed. The original class labels of these datasets were not binary; they were transformed into binary labels using the one-versus-one approach. All the modified datasets are publicly available on GitHub.[1]

4.1.3 Over-samplers

K-Means SMOTE was used in addition to the over-samplers MK-SMOTE and M-SMOTE. The details of these over-samplers, along with their hyperparameters, are provided below

- K-Mean SMOTE - KNN $\in \{3, 5, 20\}$
- MK-SMOTE - KNN $\in \{3, 5, 20\}$
- M-SMOTE - KNN $\in \{3, 5, 20\}$.

5 Evaluation

This section presents the results obtained from applying the proposed methodologies, providing quantitative evidence to support the findings. It highlights the effectiveness of the oversampling techniques through a detailed analysis of performance metrics.

5.1 Results and discussion

The performance of the proposed MK-SMOTE and M-SMOTE approaches was evaluated against K-Means SMOTE [20] using identical datasets and hyperparameters. The evaluation, based on G-Mean, F1-measure, and accuracy, involved running each experiment seven times and calculating the mean values to ensure robustness. The results, presented in Fig. 3, demonstrate that MK-SMOTE and M-SMOTE consistently outperformed or matched K-Means SMOTE across most datasets when using the KNN classifier. This suggests that the synthetic data generated by the proposed approaches significantly improved model training and prediction.

[1] https://github.com/ghulammustafacomsat/dataimbalance.

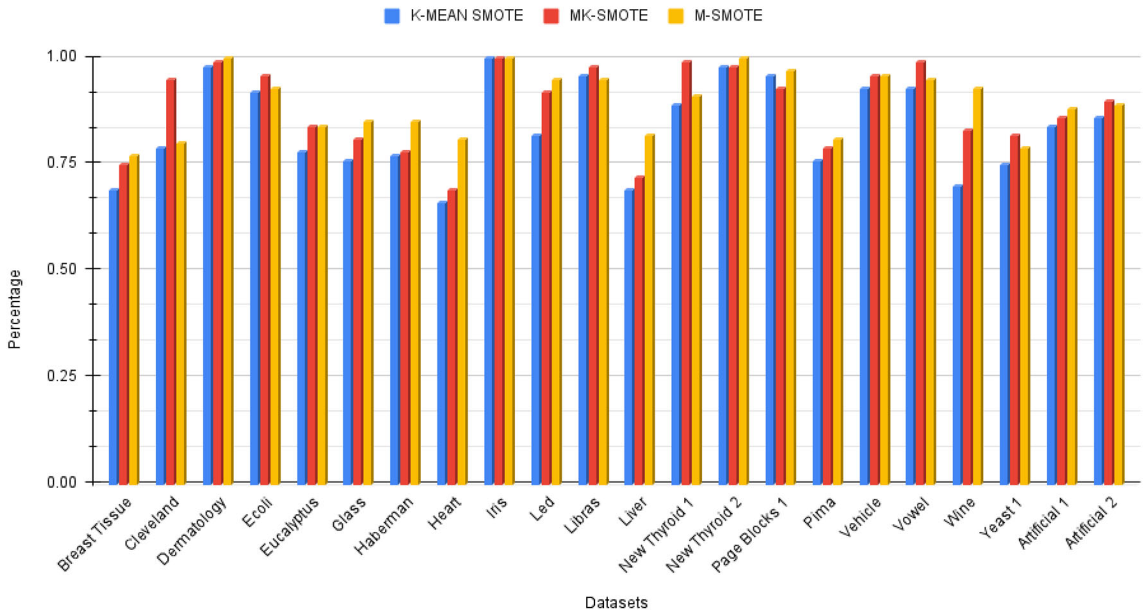


Fig. 3 Accuracy of K-Mean SMOTE, MK-SMOTE, and M-SMOTE for all data sets for KNN classifier

Table 2 Comparison of M-SMOTE and MK-SMOTE with K-Mean SMOTE

Algorithm Name	Clustering	Clustering of minority instances	Classifier used	Iterative approach	Remarks
K-mean Smote	Yes	No	No	No	Both minority and majority class used
M-Smote	No	No	Yes	Yes	Naïve Bayes used to calculate the probability of newly generated instance
MK-Smote	Yes	Yes	No	No	Only minority class used to reduce the effect of majority class on smote generated instance

5.2 Results using KNN classifier

MK-SMOTE achieved notable performance gains across all metrics for the majority of datasets, except for 'Iris' and 'Page Blocks 1', where its performance was comparable to K-Means SMOTE or showed no improvement. In contrast, M-SMOTE outperformed both MK-SMOTE and K-Means SMOTE in most cases, achieving maximum improvements of 0.14 for G-Mean, 0.12 for $F1$-score, and 0.15 for accuracy. Moreover, M-SMOTE demonstrated exceptional performance by achieving 100% accuracy, $F1$-score, and G-Mean for two datasets, underscoring its robustness in handling imbalanced data. The average metric values for MK-SMOTE−0.87 for $F1$-score, 0.86 for G-Mean, and 0.89 for accuracy reflect its consistent effectiveness. However, M-SMOTE slightly surpassed these averages with values of 0.88, 0.89, and 0.89, respectively. Dataset-specific analysis revealed strong performance by MK-SMOTE and M-SMOTE on datasets such as 'Breast Tissue', 'Ecoli', 'Haberman', and 'Vehicle', where class imbalance is more pronounced. However, neither approach improved performance on the 'Page Blocks 1' dataset, suggesting that its characteristics may limit the effectiveness of oversampling techniques. Table 2 outlines the methodological differences between the three approaches, which explain their respective performances. M-SMOTE incorporates a classification step during resampling, enabling it to generate more realistic synthetic samples and consistently outperform the other methods. MK-SMOTE, by clustering minority instances, mitigates the dominance of the majority class, leading to significant improvements for most datasets. Overall, these results confirm that MK-SMOTE and M-SMOTE are effective solutions for handling imbalanced datasets, with M-SMOTE emerging as the superior method due to its classification-based approach (Fig. 4).

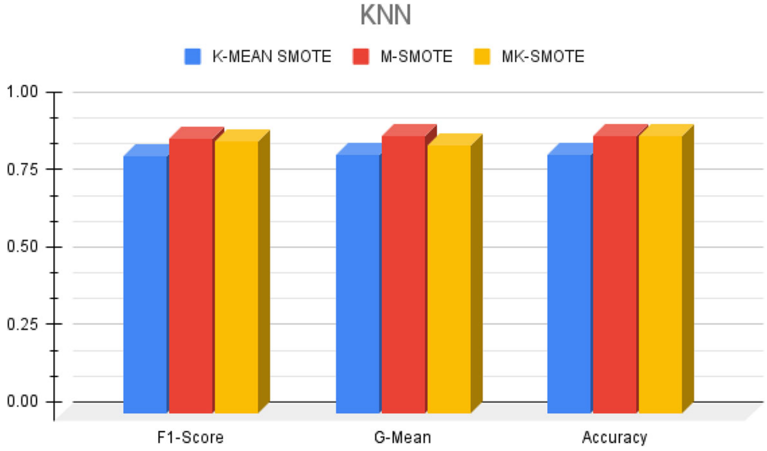


Fig. 4 Comparison of K-Mean SMOTE, MK-SMOTE, and M-SMOTE (fivefold) using KNN

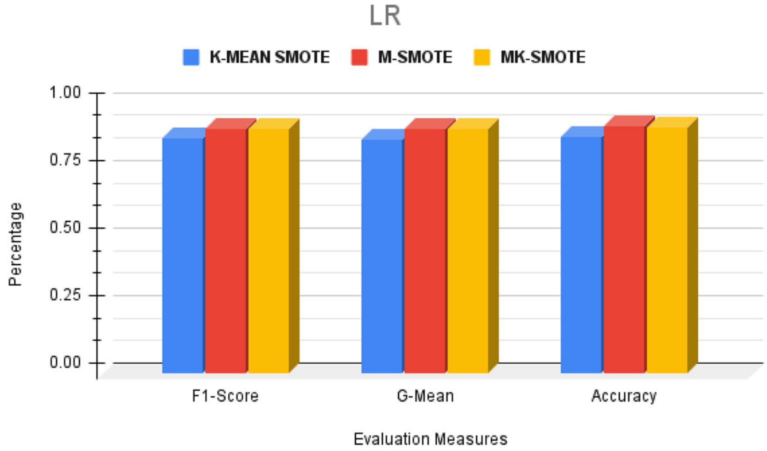


Fig. 5 Comparison of K-Mean SMOTE, MK-SMOTE, and M-SMOTE (fivefold) using LR

5.3 Results using LR classifier

The results from using the Logistic Regression (LR) classifier reveal that both MK-SMOTE and M-SMOTE outperform K-Means SMOTE, with M-SMOTE showing the most consistent improvements across all datasets. For MK-SMOTE using LR, no improvement was observed in three datasets; however, the overall average result across all measures was 90%. Notably, the largest gains with LR were 0.2 for F1-measure, 0.19 for G-Mean, and 0.2 for accuracy. M-SMOTE demonstrated superior performance, achieving an average $F1$-score of 90%, G-Mean of 90%, and accuracy of 91%. These results suggest that M-SMOTE generates synthetic instances that are less noisy compared to clustering-based techniques like K-Means SMOTE.

Additionally, the evaluation of the impact of varying threshold values for M-SMOTE revealed that as the threshold increased (from 3 to 7), all three evaluation metrics-G-Mean, $F1$-score, and accuracy-gradually improved. Specifically, accuracy and G-Mean showed a 2% increase when the threshold was raised from 3 to 7, while the $F1$-score improved by 1%. The best performance for M-SMOTE was observed when the threshold values were increased to 6 and 7, particularly on the "Artificial 1" dataset. On the "Artificial 2" dataset, improvements were noted across all threshold values.

It is also important to note that while M-SMOTE demonstrated enhanced performance, its iterative nature resulted in longer computation times. Figure 5 further illustrates the comparison, showing that LR achieved superior results with M-SMOTE across all three evaluation metrics (F1-measure, G-Mean, and accuracy) when compared to both K-Means SMOTE and MK-SMOTE.

5.4 Results with area under the curve (AUC)

The results in Fig. 6 clearly demonstrate the performance improvements achieved by the proposed MCC SMOTE and Oversee SMOTE approaches compared to the baseline K-means SMOTE, in terms of AUC values. For both KNN and LR classifiers, MCC SMOTE and Oversee SMOTE consistently outperform K-means SMOTE in most cases. Notably, Oversee SMOTE achieves the highest AUC values in challenging datasets, such as Dermatology, Glass, and Libras, where both classifiers achieve perfect scores of 1. In contrast, K-means SMOTE occasionally struggles, as seen in

Table 3 Comparison of M-SMOTE with K-Mean SMOTE using F1-Score, G-Mean, and Accuracy with a different number of iterations

Sr#	*K*-mean SMOTE			M-SMOTE											
	F1-Score	G-Mean	Accuracy	F1-Score (# iterations)				*G*-Mean(# iterations)				Accuracy(# iterations)			
				3	4	6	7	3	4	6	7	3	4	6	7
1	0.67	0.7	0.69	0.72	0.72	0.74	0.76	0.75	0.77	0.77	0.77	0.77	0.77	0.79	0.79
2	0.8	0.78	0.79	0.82	0.82	0.82	0.82	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8
3	0.93	0.98	0.98	1	1	1	1	1	1	1	1	1	1	1	1
4	0.92	0.92	0.92	0.8	0.8	0.92	0.92	0.81	0.81	0.94	0.94	0.81	0.81	0.93	0.93
5	0.79	0.78	0.78	0.82	0.82	0.82	0.82	0.83	0.83	0.83	0.83	0.84	0.84	0.84	0.84
6	0.77	0.76	0.76	0.84	0.84	0.84	0.84	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85
7	0.76	0.77	0.77	0.89	0.89	0.89	0.89	0.84	0.84	0.84	0.84	0.85	0.85	0.85	0.85
8	0.64	0.65	0.66	0.8	0.81	0.8	0.8	0.81	0.81	0.81	0.81	0.81	0.83	0.81	0.81
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	0.79	0.8	0.82	0.95	0.95	0.95	0.95	0.94	0.94	0.94	0.94	0.95	0.95	0.95	0.95
11	0.96	0.96	0.96	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
12	0.67	0.69	0.69	0.78	0.78	0.78	0.78	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82
13	0.92	0.9	0.89	0.87	0.9	0.93	0.93	0.89	0.89	0.94	0.94	0.89	0.91	0.91	0.91
14	0.98	0.98	0.98	1	1	1	1	1	1	1	1	1	1	1	1
15	0.95	0.96	0.96	0.96	0.96	0.96	0.96	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97
16	0.77	0.76	0.76	0.8	0.8	0.82	0.82	0.81	0.81	0.85	0.85	0.81	0.81	0.85	0.85
17	0.93	0.91	0.93	0.96	0.96	0.96	0.96	0.97	0.97	0.97	0.97	0.96	0.96	0.96	0.96
18	0.89	0.96	0.93	0.93	0.93	0.93	0.93	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
19	0.68	0.69	0.7	0.89	0.89	0.89	0.89	0.9	0.9	0.9	0.9	0.93	0.93	0.93	0.93
20	0.79	0.8	0.75	0.8	0.83	0.88	0.88	0.82	0.82	0.9	0.9	0.79	0.83	0.9	0.9
21	0.82	0.83	0.84	0.81	0.82	0.81	0.84	0.85	0.85	0.85	0.85	0.85	0.82	0.84	0.84
22	0.87	0.86	0.86	0.87	0.88	0.89	0.86	0.87	0.87	0.87	0.88	0.88	0.87	0.88	0.88

Sr. represents datasets defined in Table 1

the Haberman dataset, where its AUC values are significantly lower (0.53 for KNN and 0.43 for LR) compared to Oversee SMOTE (0.8 for KNN and 0.85 for LR) (Table 3).

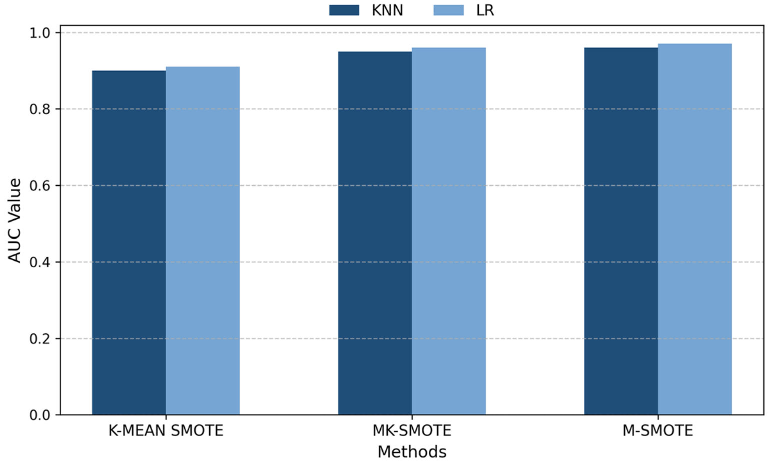
Moreover, MCC SMOTE demonstrates competitive performance, often closely matching or slightly surpassing Oversee SMOTE, particularly with KNN. For instance, in the Ecoli dataset, MCC SMOTE achieves an AUC of 1. For datasets with class imbalance ratios favoring the minority class (e.g., Heart and Pima datasets), Oversee SMOTE shows its robustness, achieving higher AUC values than both MCC SMOTE and K-means SMOTE. These results highlight the effectiveness of Oversee SMOTE in handling datasets with diverse imbalance ratios, maintaining high AUC values across classifiers and datasets.

Additionally, the figures visually corroborate these findings, with Oversee SMOTE consistently yielding the tallest bars in the AUC plots for both classifiers. This visual representation reinforces the tabular data, clearly showing that the proposed approaches significantly enhance classifier performance. The consistent improvement in AUC values across different datasets and imbalance ratios underscores the reliability of Oversee SMOTE and MCC SMOTE in addressing class imbalance challenges effectively. These findings provide strong evidence for recommending these methods in scenarios requiring high classification performance and robust handling of class imbalances.

5.4.1 Quality of newly generated data points

The quality of newly generated data points was analyzed to evaluate their placement and potential contribution to noise, as illustrated in Fig. 7. In the 'Heart' dataset, Fig. 7a shows that *K*-Means SMOTE often generates new instances that overlap with the majority class samples. This behavior can be attributed to the method's reliance on selecting majority class instances as nearest neighbors for randomly chosen minority class samples. Such overlap introduces noise, reducing the effectiveness of the resampling technique. In contrast, M-SMOTE, as shown in Fig. 7b, generates significantly fewer instances within the majority class, demonstrating a more refined approach. By ensuring that the model is trained with an equal representation of both positive and negative class instances, M-SMOTE minimizes bias in the generation process. This approach enables the creation of instances that are

Fig. 6 Comparison of K-Mean SMOTE, MK-SMOTE, and M-SMOTE using AUC



correctly classified into their respective classes, significantly reducing false positives and noisy samples.

Similar trends were observed in the 'Led' dataset, as depicted in Figs. 7c, d. K-Means SMOTE again exhibits a tendency to produce overlapping instances within the majority class, as shown in Fig. 7c, whereas M-SMOTE, in Fig. 7d, successfully avoids this issue by ensuring a more balanced and strategic distribution of synthetic samples. This improvement stems from M-SMOTE's design, which assigns probabilities to newly generated instances based on a balanced perspective of the positive and negative classes. As a result, synthetic samples are accurately placed, enhancing the dataset's quality and ensuring meaningful learning during model training.

These findings underscore the superiority of M-SMOTE over K-Means SMOTE in generating high-quality synthetic data points. The observed reductions in noise and false positives highlight M-SMOTE's robustness in addressing the challenges of class imbalance while preserving the integrity of the minority class distribution. This analysis, supported by Figs. 7a–d, demonstrates the tangible benefits of the proposed approach in maintaining dataset quality and enhancing classification outcomes.

5.4.2 Performance comparison of techniques

The performance of the oversampling techniques was evaluated by computing the True-Positive Rate (TPR) for a balanced dataset, as shown in Figs. 8 and 9. TPR is a critical metric for assessing the effectiveness of resampling methods, particularly in imbalanced datasets. In Fig. 8, which depicts TPR values generated using the K-Nearest Neighbors (KNN) classifier, M-SMOTE consistently outperforms K-Means SMOTE and MK-SMOTE across most datasets, including 'Breast Tissue,' 'Heart,' and 'Liver.' The improvement in TPR for M-SMOTE indicates its superior ability to generate synthetic samples that enhance the classifier's capacity to correctly identify positive instances.

Similarly, Fig. 9, which evaluates TPR using the Logistic Regression (LR) classifier, reveals a comparable trend. M-SMOTE maintains a higher TPR than both K-Means SMOTE and MK-SMOTE across various datasets. Notably, in datasets such as 'Diabetes' and 'Led,' the TPR achieved by M-SMOTE shows significant gains, indicating its robustness in addressing class imbalance and enhancing classification accuracy. These results highlight the adaptability of M-SMOTE to different classifiers and datasets, ensuring that synthetic instances contribute positively to the classification task.

The consistent performance of M-SMOTE across both classifiers underscores its effectiveness as an oversampling technique. By generating synthetic instances that align with the underlying data distribution, M-SMOTE ensures a balanced and meaningful augmentation of the dataset. This enables classifiers to achieve higher true positive rates, reducing the likelihood of misclassification and improving overall performance. The results in Figs. 8 and 9 validate the efficacy of M-SMOTE in handling imbalanced datasets and emphasize its practical applicability in real-world scenarios.

5.5 Results and discussion

This section provides a detailed discussion of the key findings, limitations, and strengths of the study. The findings highlight the primary outcomes and conclusions drawn from the research, focusing on the effectiveness and advantages of the proposed methodologies. The limitations are addressed by identifying challenges, constraints, or shortcomings encountered during the execution of the study, along with areas where further investigation or improvements may be necessary. Additionally, the strengths of the study are emphasized, focusing on the contributions and innovations

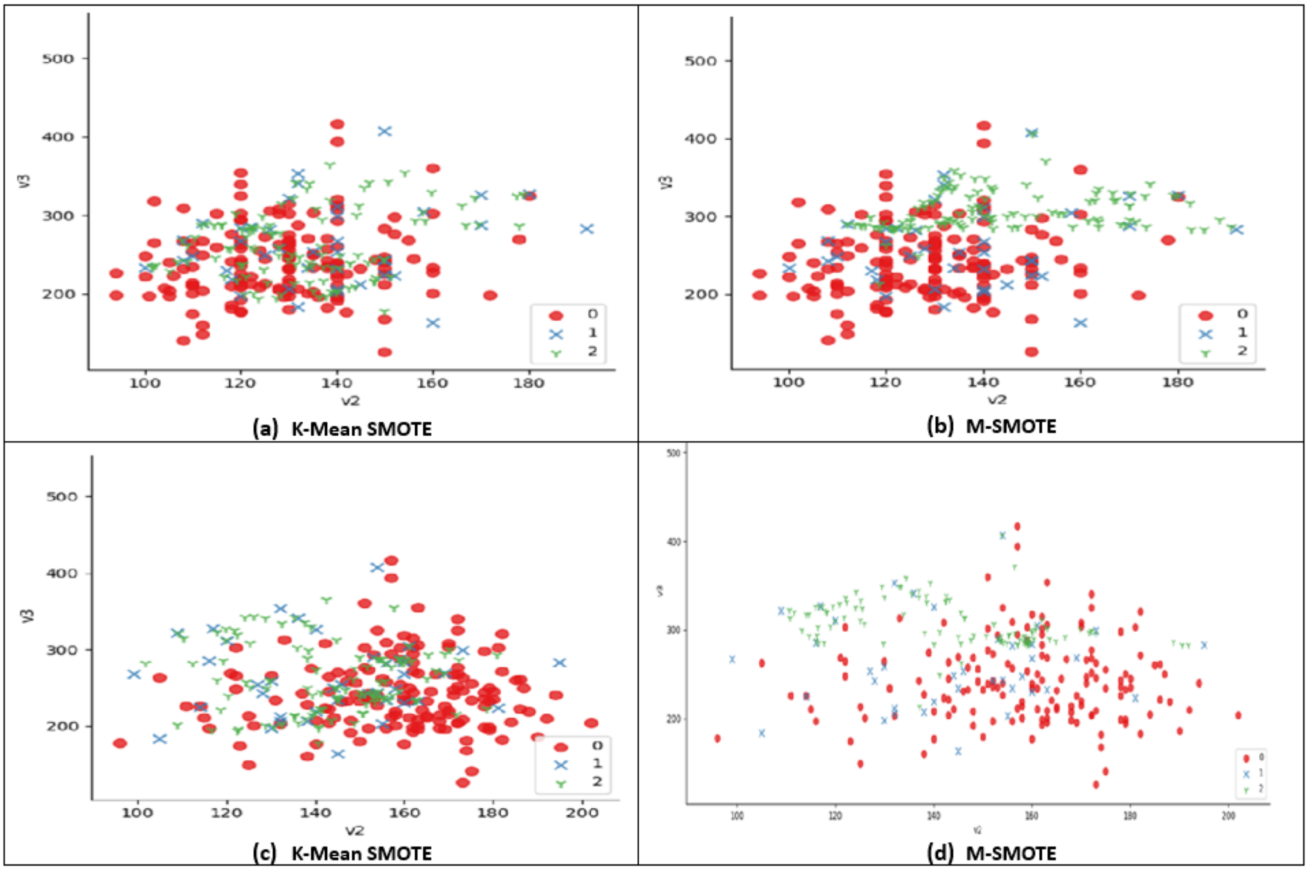


Fig. 7 Data set ‘Heart’ and ‘Led’ with synthetic instances for K-Mean SMOTE (**a**, **c**) and MK-SMOTE (**b**, **d**). ‘0’ represents majority class. ‘1’ represents minority class. ‘2’ are newly generated instances

made, as well as the aspects that reinforce the overall validity and impact of the research.

5.5.1 Findings

The study demonstrated that the proposed methods, M-SMOTE and MK-SMOTE, consistently outperformed the widely used K-Means SMOTE across diverse datasets and evaluation metrics. Specifically, M-SMOTE achieved notable improvements in G-Mean (up to 0.14), $F1$-score (up to 0.12), and accuracy (up to 0.15), highlighting its ability to effectively balance class distributions in imbalanced datasets. MK-SMOTE excelled on datasets with severe class imbalances, achieving average $F1$-scores, G-Means, and accuracies of 0.87, 0.86, and 0.89, respectively. Moreover, Oversee SMOTE achieved perfect or near-perfect AUC values (1.0) on challenging datasets such as Dermatology and Glass, demonstrating its reliability in assessing model performance.

Both M-SMOTE and MK-SMOTE showed strong adaptability across classifiers, with M-SMOTE particularly excelling with Logistic Regression (LR) due to its ability to generate high-quality synthetic samples while minimizing noise and false positives. However, the performance gains were dataset-dependent, with the methods excelling on highly imbalanced datasets while showing limited improvements on less imbalanced datasets like Page Blocks 1. This nuanced performance demonstrates the effectiveness of the proposed methods in addressing class imbalance challenges while maintaining robust performance across varying dataset complexities and metrics.

5.5.2 Strengths

The primary strength of the study lies in the novelty and effectiveness of the proposed approaches, as evidenced by their consistent improvements over K-Means SMOTE across multiple metrics, classifiers, and datasets. The use of diverse evaluation metrics, including accuracy, $F1$-score, G-Mean, and AUC, ensured a comprehensive assessment of performance. Furthermore, the methods demonstrated adaptability

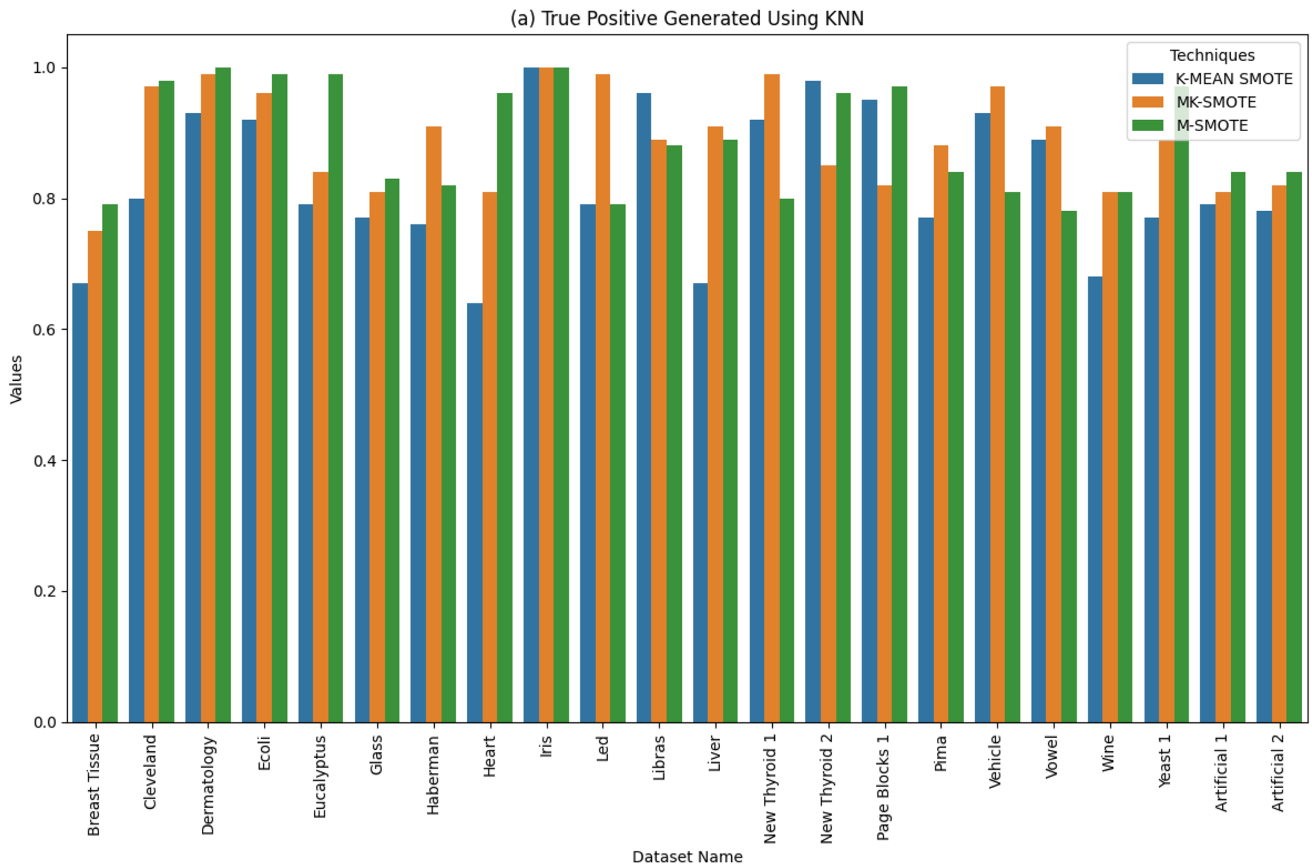


Fig. 8 Comparison of True-Positive Rate (KNN) for M-SMOTE, MK-SMOTE, and K-Mean SMOTE

across a wide range of datasets with varying imbalance ratios, showcasing their versatility.

Visualization of synthetic sample placement and noise reduction enhanced the interpretability of the results, particularly highlighting the effectiveness of M-SMOTE in minimizing noise and balancing class distributions. The robustness of the findings was ensured by averaging results over multiple iterations, further strengthening the validity of the conclusions.

5.5.3 Weaknesses

Despite its strengths, the study has several limitations that should be considered when applying MK-SMOTE and M-SMOTE to real-world problems. The performance of these methods was found to be dataset-dependent, with limited improvements on certain datasets like Iris and Page Blocks 1, suggesting that their effectiveness may vary based on dataset characteristics. Moreover, the methods may struggle with high-dimensional data, where the curse of dimensionality complicates the identification of meaningful synthetic instances.

Additionally, the iterative nature of M-SMOTE results in longer computation times, which could become impractical for large-scale datasets. The methods also showed variability in performance across different classifiers, indicating classifier dependence, and are sensitive to the choice of threshold values, which require fine-tuning for optimal results. While noise reduction was analyzed, handling extremely noisy datasets remains underexplored, and the performance on highly imbalanced or overlapping classes might be further impacted.

Finally, scalability issues arise when dealing with large datasets, particularly in terms of memory consumption and processing time. These limitations highlight the need for further research, especially in handling complex, high-dimensional, or noisy datasets.

6 Conclusion

Addressing class imbalance through oversampling remains a significant challenge in machine learning. Generating duplicate positive instances can lead to overfitting, which

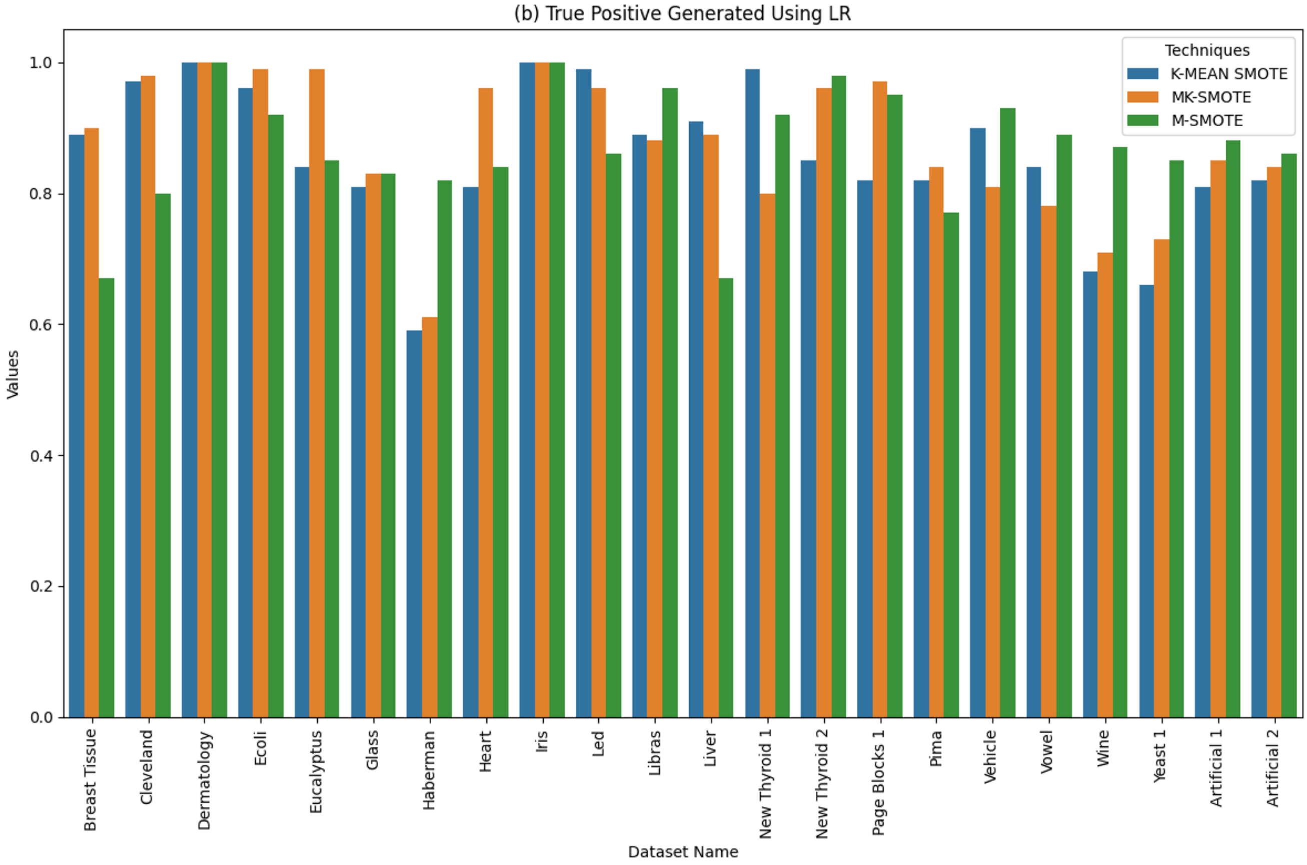


Fig. 9 Comparison of True-Positive Rate (LR) for M-SMOTE, MK-SMOTE, and *K*-Mean SMOTE

obscures crucial information and ultimately degrades classifier performance. Moreover, oversampling techniques may inadvertently introduce noise from the majority class into the synthetic instances. To effectively improve classification algorithms on imbalanced datasets, it is essential to ensure that synthetic data are free from noisy samples, avoids excessive duplication, and minimizes the influence of the majority class.

In this study, two novel oversampling techniques, MK-SMOTE and M-SMOTE, were proposed to address these challenges. MK-SMOTE achieved an accuracy of 90% by clustering only the minority class and creating a safe zone for generating new instances, reducing the introduction of noisy samples. M-SMOTE further improved performance by iteratively calculating the probability of newly generated instances, ensuring the inclusion of high-quality synthetic samples and achieving an overall accuracy of 91%. Additionally, M-SMOTE minimized duplication by carefully distributing the new instances within the minority class.

The comparative analysis demonstrated that both MK-SMOTE and M-SMOTE outperformed the widely used *K*-Means SMOTE technique. These results underscore the effectiveness of the proposed methods in enhancing classifier performance on imbalanced datasets while mitigating the limitations of traditional oversampling approaches.

7 Future work

While the proposed approaches showed significant improvements over the baseline method and achieved high-quality results, several opportunities for further research remain in this domain. One key observation during the execution of Oversee SMOTE was its sensitivity to runtime, particularly due to the iterative probability calculation step. Future work could focus on optimizing the algorithm at the structural level to reduce computational overhead and enhance efficiency.

Additionally, identifying the most suitable range for threshold values of the minority class is an area worth exploring. Determining an optimal threshold range could enable the generation of more representative and effective synthetic data, further improving the performance of oversampling techniques. These directions present promising opportunities to refine and advance the methodologies introduced in this study (Table 4).

Table 4 Hyperparameter tuning process for MK-SMOTE and M-SMOTE

Parameter	MK-SMOTE	M-SMOTE
Clustering algorithm	K-means clustering (applied only to minority class)	Not used (Naïve Bayes classifier-based)
Number of clusters (k)	Hyperparameter choice. **Tuning Process**: The number of clusters is determined using cross-validation. Different values of k (e.g., 2, 3, 5, 10) are tested, and the optimal value is chosen based on classification performance. **Used value**: The optimal k value is reported as 3 based on cross-validation results	Not applicable (uses Naïve Bayes instead of clustering)
Exponential factor (de)	Controls the influence of distance in density calculation. **Tuning Process**: This factor is tuned empirically by testing different values (e.g., 0.1, 0.2, 0.3) and selecting the one that maximizes performance (e.g., classification accuracy or $F1$-score). **Used value**: Set to 0.2 for the experiments	Not used
Density factor (df)	Calculated based on the number of minority instances and average distance. **Tuning Process**: Adjusted indirectly as part of the overall algorithm tuning process; the resulting density factor affects cluster sparsity and oversampling	Not used
Sparsity factor	Reflects the density of a cluster; sparser clusters are oversampled more. **Tuning Process**: Evaluated based on cluster analysis during cross-validation. **Used Value**: Set based on the density factor, with higher sparsity leading to more synthetic samples	Not applicable (M-SMOTE uses Naïve Bayes probability thresholds)
Sampling weight	Derived from sparsity factor, determines how many synthetic samples to generate for each cluster. **Tuning Process**: Sampling weight is adjusted by changing the sparsity factor. No direct hyperparameter tuning was performed for this factor	Based on probability of synthetic instance belonging to minority class
Oversampling method	SMOTE (using nearest neighbors to create synthetic instances). **Tuning Process**: The number of nearest neighbors (k) is adjusted based on the dataset characteristics and cross-validation. **Used value**: $k = 5$ neighbors used in the experiments	SMOTE (with Naïve Bayes validation for synthetic instances)
SMOTE nearest neighbors (knn)	The number of nearest neighbors used in SMOTE. **Tuning Process**: The number of neighbors (k) is optimized using cross-validation to determine the optimal number for generating synthetic samples. **Used value**: $k = 5$ neighbors based on cross-validation performance	$k = 5$ neighbors (same as MK-SMOTE)
Imbalance ratio threshold (irt)	Determines when to apply oversampling based on class imbalance. **Tuning Process**: This threshold is set based on empirical testing and dataset imbalance. For this experiment, a threshold of 0.1 was used to trigger oversampling	Not explicitly used in the algorithm
Threshold range for minority class	Not applicable	**Tuning process**: The thresholds $T_low)$ and $T_high)$ were tuned based on Naïve Bayes probability output. The thresholds were varied and optimized by testing the $F1$-score for different combinations. **Used values**: $T_low) = 0.3$, $T_high) = 0.7$, chosen based on the validation set
Number of iterations (N)	Not applicable	**Tuning process**: The number of iterations was set empirically. After testing different iteration numbers (e.g., 3, 5, 10), $N = 5$ was chosen as it showed the best performance. **Used value**: $N = 5$ iterations
Naïve Bayes classifier	Not used	**Tuning process**: Naïve Bayes was chosen for its probabilistic approach to assigning synthetic samples to the minority class. The performance was validated with various hyperparameters, but no explicit tuning for Naïve Bayes was done

Acknowledgements This project was conceived by Asifa Kanwal and Nayyer Masood. Nayyer Masood and Maryam Abdul Ghafoor supervised the research. Asifa Kanwal designed and conducted the experiments. Data analysis was performed by Asifa Kanwal and Samreen Ayaz, with conceptual advice from Nayyer Masood and Maryam Abdul Ghafoor. Asifa Kanwal wrote the manuscript, incorporating feedback from Maryam Abdul Ghafoor, Ghulam Mustafa, and Samreen Ayaz. All authors contributed to the discussion of the results and commented on the manuscript.

Author Contributions A.K conduct experiment and write paper, N.M, M.A.G, A.Y supervise the research Study, G.M reveiew the paper.

Funding The authors declare that no funding was received for the research, authorship, or publication of this manuscript.

Data availability The data supporting this study are publicly available at GitHub at Link https://github.com/ghulammustafacomsat/dataimbalance.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Mustafa, G., Usman, M., Yu, L., Afzal, M.T., Sulaiman, M., Shahid, A.: Multi-label classification of research articles using word2vec and identification of similarity threshold. Sci. Rep. **11**(1), 21900 (2021). https://doi.org/10.1038/s41598-021-01460-7
2. Mustafa, G., Rauf, A., Ahmed, B., Afzal, M.T., Akhunzada, A., Alharthi, S.Z.: Comprehensive evaluation of publication and citation metrics for quantifying scholarly influence. IEEE Access **11**, 65759–65774 (2023)
3. Usman, M., Mustafa, G., Afzal, M.T.: Ranking of author assessment parameters using logistic regression. Scientometrics **126**(1), 335–353 (2021)
4. Mustafa, G., Usman, M., Afzal, M.T., Shahid, A., Koubaa, A.: A comprehensive evaluation of metadata-based features to classify research paper's topics. IEEE Access **9**, 133500–133509 (2021). https://doi.org/10.1109/ACCESS.2021.3115148
5. Igual, L., Seguí, S.: Introduction to Data Science: A Python Approach to Concepts, Techniques and Applications, vol. 2, pp. 1–218. Springer, Berlin (2024). https://doi.org/10.1007/978-3-319-50017-1
6. Shah, S.M.A.H., Ullah, A., Iqbal, J., Bourouis, S., Ullah, S.S., Hussain, S., Khan, M.Q., Shah, Y.A., Mustafa, G.: Classifying and localizing abnormalities in brain MRI using channel attention based semi-Bayesian ensemble voting mechanism and convolutional auto-encoder. IEEE Access **11**, 75528–75545 (2023)
7. Basha, S.J., Madala, S.R., Vivek, K., Kumar, E.S., Ammannamma, T.: A review on imbalanced data classification techniques. In: 2022 International Conference on Advanced Computing Technologies and Applications (ICACTA), vol. 2, pp. 1–6. IEEE (2022). https://doi.org/10.1109/ICACTA54488.2022.9753392
8. Ghosh, K., Bellinger, C., Corizzo, R., Branco, P., Krawczyk, B., Japkowicz, N.: The class imbalance problem in deep learning. Mach. Learn. **113**(7), 4845–4901 (2024)
9. Khan, A.A., Chaudhari, O., Chandra, R.: A review of ensemble learning and data augmentation models for class imbalanced problems: combination, implementation and evaluation. Expert Syst. Appl. **244**, 122778 (2024)
10. Mustafa, G., Rauf, A., Al-Shamayleh, A.S., Sulaiman, M., Alrawagfeh, W., Afzal, M.T., Akhunzada, A.: Optimizing document classification: unleashing the power of genetic algorithms. IEEE Access **11**, 83136–83149 (2023)
11. Elreedy, D., Atiya, A.F.: A comprehensive analysis of synthetic minority oversampling technique (smote) for handling class imbalance. Inf. Sci. **505**(7), 32–64 (2019). https://doi.org/10.1016/j.ins.2019.07.070
12. Rachel Schutt, C.O.: Doing Data Science: Straight Talk from the Frontline, vol. 3, p. 406. O'Reilly Media, Inc. (2013). https://doi.org/10.1016/C2009-0-61819-5
13. Goswami, S., Singh, A.K.: A literature survey on various aspect of class imbalance problem in data mining. Multimed. Tools Appl. **2024**, 1–26 (2024)
14. Zhou, Q., Sun, B.: Adaptive k-means clustering based undersampling methods to solve the class imbalance problem. Data Inf. Manag. **8**(3), 100064 (2024)
15. Mustafa, G., Rauf, A., Al-Shamayleh, A.S., Afzal, M.T., Waqas, A., Akhunzada, A.: Defining quantitative rules for identifying influential researchers: insights from mathematics domain. Heliyon **10**(9), e30318 (2024)
16. Singh, A., Ranjan, R.K., Tiwari, A.: Credit card fraud detection under extreme imbalanced data: a comparative study of data-level algorithms. J. Exp. Theor. Artif. Intell. **34**(4), 571–598 (2022). https://doi.org/10.1080/0952813X.2021.1907795
17. Ahmed, B., Wang, L., Mustafa, G., Afzal, M.T., Akhunzada, A.: Evaluating the effectiveness of author-count based metrics in measuring scientific contributions. IEEE Access **11**, 101710–101726 (2023)
18. Giorgio, A., Cola, G., Wang, L.: Systematic review of class imbalance problems in manufacturing. J. Manuf. Syst. **71**, 620–644 (2023)
19. Ahmed, B., Wang, L., Al-Shamayleh, A.S., Afzal, M.T., Mustafa, G., Alrawagfeh, W., Akhunzada, A.: Machine learning approach for effective ranking of researcher assessment parameters. IEEE Access **11**, 133294–133312 (2023)
20. Douzas, G., Bacao, F., Last, F.: Improving imbalanced learning through a heuristic oversampling method based on k-means and smote. Inf. Sci. **465**(1), 1–20 (2018). https://doi.org/10.1016/j.ins.2018.06.056
21. Kaur, P., Gosain, A.: Comparing the behavior of oversampling and undersampling approach of class imbalance learning by combining class imbalance problem with noise. In: ICT Based Innovations: Proceedings of CSI 2015, vol. 653, pp. 23–30 (2018). https://doi.org/10.1007/978-981-10-6602-3_3
22. Lim, P., Goh, C.K., Tan, K.C.: Evolutionary cluster-based synthetic oversampling ensemble (eco-ensemble) for imbalance learning. IEEE Trans. Cybern. **47**(9), 2850–2861 (2016). https://doi.org/10.1109/TCYB.2016.2579658
23. Mustafa, G., Rauf, A., Al-Shamayleh, A.S., Ahmed, B., Alrawagfeh, W., Afzal, M.T., Akhunzada, A.: Exploring the significance of publication-age-based parameters for evaluating researcher impact. IEEE Access **11**, 86597–86610 (2023)
24. Mustafa, G., Rauf, A., Afzal, M.T.: Enhancing author assessment: an advanced modified recursive elimination technique (mret) for ranking key parameters and conducting statistical analysis of top-ranked parameter. Int. J. Data Sci. Anal. **2024**, 1–23 (2024)
25. Zhang, J., Li, Y., Zhang, B., Wang, X., Gong, H.: A new oversampling approach based differential evolution on the safe set for highly imbalanced datasets. Expert Syst. Appl. **234**(11), 121039 (2023). https://doi.org/10.1016/j.eswa.2023.121039
26. Mustafa, G., Rauf, A., Afzal, M.T.: Gk index: bridging gf and k indices for comprehensive author evaluation. Knowl. Inf. Syst. **2024**, 1–36 (2024)
27. Hu, J., He, X., Yu, D.J., Yang, X.B., Yang, J.Y., Shen, H.B.: A new supervised over-sampling algorithm with application to protein-nucleotide binding residue prediction. PLoS ONE **9**(9), 107676 (2014). https://doi.org/10.1371/journal.pone.0107676

28. Ahmed, B., Wang, L., Hussain, W., Mustafa, G., Afzal, M.T.: Investigating scholarly indices and their contribution to recognition patterns among awarded and non-awarded researchers. Int. J. Data Sci. Anal. **2025**, 1–18 (2025)
29. Lee, J., Kim, N.R., Lee, J.H.: An over-sampling technique with rejection for imbalanced class learning. In: Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication, pp. 1–6 (2015). https://doi.org/10.1145/2701126.2701181
30. Mustafa, G., Rauf, A., Tanvir Afzal, M.: Mret: modified recursive elimination technique for ranking author assessment parameters. PLoS ONE **19**(6), 0303105 (2024)
31. Torres, F.R., Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F.: Smote-d a deterministic version of smote, vol. 9703, pp. 1–11 (2016). https://doi.org/10.1007/978-3-319-39393-3_18
32. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**(1), 321–357 (2002). https://doi.org/10.1613/jair.953
33. Kang, Q., Chen, X., Li, S., Zhou, M.: A noise-filtered under-sampling scheme for imbalanced classification. IEEE Trans. Cybern. **47**(12), 4263–4274 (2016). https://doi.org/10.1109/TCYB.2016.2606104
34. Siriseriwan, W., Sinapiromsaran, K.: Adaptive neighbor synthetic minority oversampling technique under 1NN outcast handling. Songklanakarin J. Sci. Technol. **39**(5), 1–10 (2017). https://doi.org/10.1145/2701126.2701181
35. Sharma, S., Bellinger, C., Krawczyk, B., Zaiane, O., Japkowicz, N.: Synthetic oversampling with the majority class: A new perspective on handling extreme imbalance, vol. 47, pp. 447–456 (2018). https://doi.org/10.1109/ICDM.2018.00060
36. Liu, Y., Liu, Y., Bruce, X.B., Zhong, S., Hu, Z.: Noise-robust oversampling for imbalanced data classification. Pattern Recognit. **133**(5), 109008 (2023). https://doi.org/10.1016/j.patcog.2022.109008
37. Wongvorachan, T., He, S., Bulut, O.: A comparison of undersampling, oversampling, and smote methods for dealing with imbalanced classification in educational data mining. Information **14**(1), 54 (2023). https://doi.org/10.3390/info14010054
38. Feng, F., Li, K.C., Yang, E., Zhou, Q., Han, L., Hussain, A., Cai, M.: A novel oversampling and feature selection hybrid algorithm for imbalanced data classification. Multimed. Tools Appl. **82**(3), 3231–3267 (2023). https://doi.org/10.1007/s11042-022-13240-0
39. Adnan, S.M., Ahmad, W., Mahmood, I., Mustafa, G., Dattana, V.: Enhancing text mining efficiency using an effective topic modeling approach. Tech. J. **29**(01), 39–46 (2024)
40. Lin, Z., Feng, M., Santos, C.N.D., Yu, M., Xiang, B., Zhou, B., Bengio, Y.: A structured self-attentive sentence embedding. arXiv preprint (2017). arXiv:1703.03130
41. Rayhan, F., Ahmed, S., Mahbub, A., Jani, R., Shatabda, S., Farid, D. M.: Cusboost: cluster-based under-sampling with boosting for imbalanced classification. In: 2017 2nd International Conference on Computational Systems and Information Technology for sustainable solution (csitss), pp. 1–5. IEEE (2017)
42. Sikora, R., Raina, S.: Controlled under-sampling with majority voting ensemble learning for class imbalance problem. In: Intelligent Computing: Proceedings of the 2018 Computing Conference, vol. 2, pp. 33–39. Springer International Publishing (2019)
43. Bellinger, D.C.:Lead contamination in Flint—an abject failure to protect public health. New Engl. J. Med. **374**(12), 1101–1103 (2016)
44. Han, J., Kesner, P., Metna-Laurent, M., Duan, T., Xu, L., Georges, F., Koehl M., Abrous, D.N., Mendizabal-Zubiaga, J., Grandes, P., Liu, Q., Bai, G., Wang, W., Xiong, L., Ren, W., Marsicano, G., Zhang, X.: Acute cannabinoids impair working memory through astroglial CB1 receptor modulation of hippocampal LTD. Cell **148**(5), 1039–1050 (2012)
45. Mullick, S.S., Datta, S., Das, S.: Adaptive learning-based k-nearest neighbor classifiers with resilience to class imbalance. IEEE Trans. Neural Netw. Learn. Syst. **29**(11), 5713–5725 (2018)
46. O'Neil, C., Schutt, R.: Doing data science: straighttalk from the frontline. O'Reilly Media, Inc. (2013)
47. Thabtah, F., Hammoud, S., Kamalov, F., Gonsalves, A.:Data imbalance in classification: Experimental evaluation. Inf. Sci. **513**, 429–441 (2020)
48. Yun, S.H., Sim, E.H., Goh, R.Y., Park, J.I., Han, J.Y.: Platelet activation: the mechanisms and potential biomarkers. BioMed Res. Int. **2016**(1), 9060143 (2016)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.