Journal of Electronics, Electromedical Engineering, and Medical Informatics
Homepage: jeeemi.org; Vol. 7, No. 3, July 2025, pp: 740-762 e-ISSN: 2656-8632

RESEARCH ARTICLE OPEN ACCESS

Addressing Intrinsic Data Characteristics Issues of Imbalance Medical Data Using Nature Inspired Percolation Clustering

Kaikashan Siddavatam, Subhash Shinde

Department of Computer Engineering, Lokmanya Tilak College of Engineering, Navi Mumbai, India

Corresponding author: Kaikashan Siddavatam (e-mail: kaikashan.s@ltce.in), **Author(s) Email**: Dr. Subhash Shinde (e-mail: skshinde@ltce.in)

Abstract **Data on diseases are generally skewed towards either positive or negative cases, depending on their prevalence. The problem of imbalance can significantly impact the performance of classification models, resulting in biased predictions and reduced model accuracy for the underrepresented class. Other factors that affect the performance of classifiers include intrinsic data characteristics, such as noise, outliers, and within-class imbalance, which complicate the learning task. Contemporary imbalance handling techniques employ clustering with SMOTE (Synthetic Minority Oversampling Technique) to generate realistic synthetic data that preserves the underlying data distribution, generalizes unseen data and mitigates overfitting to noisy points. Centroid-based clustering methods (e.g., K-means) often produce synthetic samples that are too clustered or poorly spaced. At the same time, density-based methods (e.g., DBSCAN) may fail to generate sufficient meaningful synthetic samples in sparse regions. The work aims to develop nature-inspired clustering that, combined with SMOTE, generates synthetic samples that adhere to the underlying data distribution and maintain sparsity among the data points that enhance performance of classifier. We propose PC-SMOTE, which leverages Percolation Clustering (PC), a novel clustering algorithm inspired by percolation theory. The methodology of PC utilizes a connectivity-driven framework to effectively handle irregular cluster shapes, varying densities, and sparse minority instances. The experiment was designed using a hybrid approach to assess PC-SMOTE using synthetically generated data with variable spread and other parameters; second, the algorithm was evaluated on eight sets of real medical datasets. The results show that the PC-SMOTE method works excellently for the Breast cancer dataset, Parkinson's dataset, and Cervical cancer dataset, where AUC is in the range of 96% to 99%, which is high compared to the other two methods. This demonstrates the effectiveness of the PC-SMOTE algorithm in handling datasets with both low and high imbalance ratios and often demonstrates competitive or superior performance compared to K-means and DBSCAN combined with SMOTE in terms of AUC, F1-score, G-mean, and PR-AUC.**

Keywords: **Artificial Intelligence (AI), DBSCAN, Imbalance Dataset, K-Means, SMOTE, Percolation**

I. Introduction

The implementation of AI in the healthcare sector is progressively increasing focusing on health challenges, mostly in the improvement of disease diagnosis. The datasets used in the medical field to train machine learning models are often highly imbalanced. The nonlinear and imbalanced data distribution in medical datasets of diseases such as cancer, Parkinson's, hepatitis, and heart conditions presents significant challenges. Imbalance occurs when the majority class (e.g., healthy patients) vastly outnumbers the minority class (e.g., patients with a disease)[1], where minority samples often hold critical information. In medical applications like cancer diagnosis or rare species recognition, misclassifying minority instances can compromise model robustness and lead to severe real-world consequences [2][3]. Mislabeling minority instances as majority worsens the imbalance while mislabeling majority instances as minority reduces classification accuracy for the minority class. Traditional machine learning algorithms, trained on imbalanced datasets, often prioritize the majority class, resulting in poor performance and higher error rates in critical tasks like disease detection [4]. For instance, misclassifying malignant tumors as benign can be fatal and have serious repercussions [2].

Manuscript received December 8, 2024; Revised February 10, 2025; Accepted March 1, 2025; Available online June 5, 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.835

The other factors that affect the performance of classifiers are data characteristics such as noise(Lemma B), outliers[5][4], (Lemma A) small disjuncts(Lemma C) and imbalance within class that complicates the learning task, and its impacts have been investigated [6][7][8][9]. Within-class imbalance problems occur in some imbalanced datasets due to the unequal distribution of data points, which form sparse or dense subgroups. The data points that are located far from the dense majority data and do not conform to a notion of normal behavior are considered as outliers. Rare data points that are far from dense regions are also considered outliers. The samples of minority data points surrounded by majority samples are the noise samples. Class noise caused by mislabeled instances complicates classification, especially in imbalanced datasets. Approaches to address these issues are typically categorized as algorithm-level methods [9], ensemble methods [10][11] and data-level techniques [13] where the data-level techniques address the class imbalance by modifying the number of instances per class [14]. It uses under sampling to reduce majority class samples and oversampling to generate synthetic minority samples. [16][13] The Synthetic Minority Oversampling Technique (SMOTE) [15]is a widely used technique for class imbalance. It generates synthetic minority samples by random interpolation. However, it faces several challenges, including overgeneralization, sample overlapping, noise, and deviations in class boundaries [5]. These issues can degrade performance, especially in datasets with dispersed or irregularly distributed minority samples. SMOTE was originally designed for binary classification and can be extended to multi-class imbalanced datasets through various adaptation strategies. The most straightforward approach involves applying SMOTE iteratively to each minority class, as demonstrated by Static-SMOTE [38].The other more sophisticated decomposition-based strategies utilize binarization techniques, with One-vs-One (OVO) and One-vs-All (OVA) being the most prevalent [39]. Research consistently shows that "OVO and oversampling was the most robust approach overall" for multi-class problems, as it simplifies boundary areas and reduces overlapping compared to OVA, which can create extreme imbalance when contrasting small minorities against aggregated majorities. These multi-class imbalanced scenarios challenge standard SMOTE due to its focus solely on minority classes, ignoring inter-class overlap and optimal sampling rates. The methods proposed in the study such as MKC-SMOTE[40] for multiclass imbalance dataset and address by considering all classes and preserving class structure, achieving up to 13.86% performance gains over SMOTE. Thus, multi-class-specific algorithms are preferred for robust oversampling in complex imbalanced datasets. Various variations of SMOTE have been proposed to tackle these challenges, such as Borderline SMOTE [16], which focuses on oversampling near class boundaries; ADASYN [17], adaptive synthetic oversampling technique which generates more samples from harder-to learn instances; and Safe-level-SMOTE [18], which selects instances based on a weight reflecting their safety level. While these methods balance class distributions, they often neglect the minority sample distribution, potentially generating noisy or unsafe samples [20], amplifying overlap, and worsening within-class imbalance. A significant limitation of SMOTE and its variants is the fixed k-value for neighbor selection.

This static approach fails to adapt to varying dataset complexities. The dense areas require more neighbors for generalization, while the sparse or border area needs fewer to prevent noise and errors. In addition, sampling based on outliers also makes the data distributions distorted and prone to overfitting [21].
Recently, clustering-based methods are combined with oversampling methods to enhance SMOTE [22][23][24] by addressing imbalances within minority classes, overlapping, noise and between-class while emphasizing the importance of data points of minority class and generating synthetic samples within each sub-cluster to ensure a balanced distribution across sparse and dense subgroups. Clustering also helps in finding the total count of samples to be generated for minority class subgroups, thus improving the balance of the data [21]. Clustering allows the identification of homogeneous subgroups within the minority class, ensuring more targeted and effective synthetic data generation. Additionally, this approach provides a better estimation of the spatial distribution of minority class instances, enhancing classification performance. Clustering methods, such as density-based and hierarchical clustering, are well-suited for identifying clusters with irregular shapes and varying densities, which is often the case in medical datasets. However, the appropriate choices of clustering technique and algorithm selection depends on insight such as data distribution and kind of analysis to be performed, with regard to data size. As most medical datasets can be complex and diverse, they may have different types of features, varying densities, and non-linear relationships, so it is crucial to select the correct clustering algorithm to handle these aspects. Some algorithms, like K-means, consider the clusters to be spherical and evenly sized, which may not suit the irregular and varied shapes found in medical data. Others, density clusters, can find the clusters of arbitrary shapes but may struggle with outliers. By accurately identifying these clusters, it becomes possible to generate required quality synthetic data that mirrors the spatial structure of the minority class. This improves the model's ability to accurately identify the

Manuscript received December 8, 2024; Revised February 10, 2025; Accepted March 1, 2025; Available online June 5, 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.835

minority class accurately, thereby improving classification performance for rare and critical diseases. We have not found any research study which has specifically investigated how these cluster algorithms are affected by the size of a dataset and the distributions between datasets.

Clustering methods like DBSCAN [25]and k-means [21] are often ineffective for datasets with non-uniform density distributions or sparse data. To overcome these challenges, Percolation Clustering (PC), a novel clustering algorithm inspired by percolation theory, has been proposed. PC leverages a connectivity-driven framework to effectively handle irregular cluster shapes, varying densities, and sparse minority instances. In our approach, the dataset is first clustered using PC, and for each sub-cluster SMOTE is applied to balance the data . This percolation theory was first introduced by Broadbent and Hammersley in 1957 [33] which deals with the connectivity within porous materials, showing how components get connected and evolve inside a system by following interaction rules or specific thresholds. Such a principle creates a strong basis to tackle clustering issues, especially concerning healthcare data within a complex setting. The contributions of this research paper are as follows:

1. The presented work provides formal mathematical analysis to provide deeper insights into how SMOTE handles data imbalance while exposed to data intrinsic characteristics issues.
2. A novel clustering approach based on percolation theory that integrates with the SMOTE algorithm is proposed to effectively handle irregular cluster shapes, varying densities, and sparse minority instances.
3. A comparative evaluation is presented for the proposed method with the existing clustering algorithm(s) for different real medical datasets and Synthetic created datasets based on data characteristics, including class distribution, density, and complexity.

The rest of the paper is divided into four sections. In second Section, related work with a focus on clustering algorithms combined with oversampling techniques is presented. The third Section presents a mathematical analysis of SMOTE effectiveness in handling outliers and noise. The fourth Section offers in depth an explanation of the proposed method. The fifth and sixth Section discusses experimentation design and results in detail. Finally, the seventh and eighth Section gives the discussion and the conclusion of the paper.

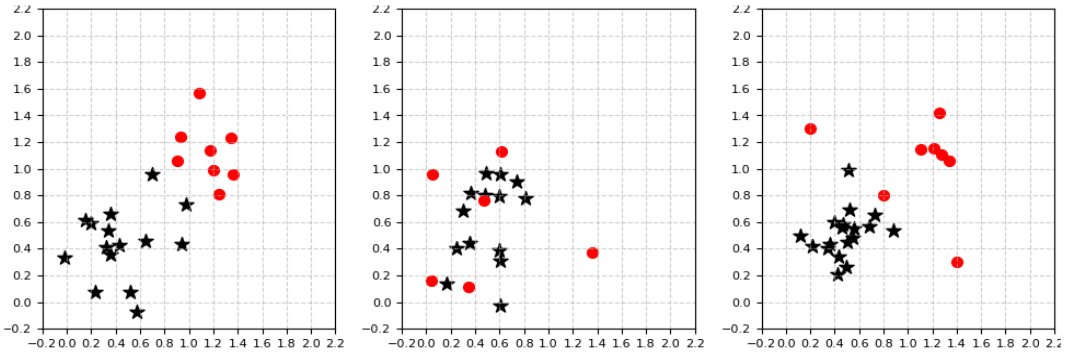
II. Related Work:

In much of the previous research work it can be seen that clustering techniques combined with data-level resampling techniques are used to solve the issues of imbalance related to both between-class and within-class allowing for more nuanced resampling that can preserve the structure of minority class distributions-means, DBSCAN, etc., are some of the existing clustering algorithms that have successfully improved the algorithm of SMOTE to a considerable extent and allowed the improved SMOTE algorithm with better processing with unbalanced datasets. We aim to summarize and analyze various clustering algorithms that are applied to imbalanced medical datasets with different data distributions.

K-means clustering, which is considered classical algorithms, has been used most for handling imbalanced data. The author proposed K-means SMOTE [2] by combining K-means with oversampling to handle the class imbalance. This method involves clustering the entire dataset, calculating the required number of samples based on the count and imbalance ratio of each sub-cluster, and subsequently applying SMOTE within clusters containing sparse minority data. Thus, it avoids intra class imbalance and achieves between-class balance. To address the imbalance problem in medical data, the author introduces KNSMOTE[3] to partition the dataset and filter out boundary and noisy samples prone to misclassification, retaining only ”safe samples” for synthesis. Next, an improved SMOTE generates synthetic samples through linear interpolation of these filtered” safe samples. Despite their utility, K-means-based approaches have some significant limitations. These include poor handling of noise and outliers, which leads to misclassifications and an inability to manage non-spherical or irregular cluster shapes. Another challenge is to identify the optimal number of clusters (k), which often requires prior knowledge or iterative testing. Moreover, K-means tend to overlook small or underrepresented groups, a critical issue in imbalanced datasets. Its inability to process complex or overlapping data distributions further limits its adaptability, especially for datasets with intricate or irregular structures.

To overcome the limitations of K-means, researchers have explored advanced clustering techniques. One such method, DBSCAN (Density-Based Spatial Clustering of Applications with Noise), is a widely used density-based clustering approach that divides data into different groups based on density eliminating the need to predefine the number of clusters [25]. DBSCAN excels at identifying clusters of arbitrary shapes and is highly resistant to noise, making it ideal for complex and varied datasets. Unlike K-means, DBSCAN identifies clusters by analyzing the density of data points and can recognize core clusters in areas of high density. while recognizing boundary regions with lower density, which may indicate overlap [24]. The study [25] uses DBSCAN clustering method to find

Manuscript received December 8, 2024; Revised February 10, 2025; Accepted March 1, 2025; Available online June 5, 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.835



(a) (b) (c)

Fig.1. Dataset with (a) high, (b) medium, (c) low variability in the density of data points

homogeneous patient groups with similar characteristics. Applied to coronary heart disease (CHD) data, it uncovers clusters representing mothers with shared traits and risk factors linked to CHD in infants. Reduced Noise-SMOTE (RNSMOTE) [27] as presented in a study, is another technique for addressing imbalanced datasets .Initially it applies SMOTE to oversample the minority class, followed by DBSCAN to detect and eliminate noise, ensuring that synthetic samples are generated from clean data. This improved dataset is then rebalanced with SMOTE before being input into a classifier. DB-SMOTE introduced by author [22], employs DBSCAN clustering and noise removal, performing oversampling along the smallest density connecting path to the central instance of the minority cluster. While DB-SMOTE avoids placing synthetic samples in most areas, it may still generate samples in non-representative regions due to class overlap. DBSCAN struggles with clusters of varying densities, as its density-based core point definition cannot adapt to differing density levels within the data. Density-based clustering methods often struggle with low-density clusters, which have regions with fewer data points for an area and form small clusters in imbalanced datasets.

Recently, another density-based method called Density Peak Clustering (DPC) has been developed to address these limitations, assuming cluster centers are at higher local densities and are significantly distanced from other cluster centers. The research paper [28] describes ADPCHFO, an adaptive oversampling method for imbalanced datasets, using Density Peak Clustering (DPC) and a heuristic filter. Density peak clustering (DPC) fails with datasets whose density distribution varies because the distance threshold dc needs to be determined manually, making it hard to determine correctly. It performs well only with convex data; however, it fails when the datasets are non-convex, noisy, or in overlapping regions(Lemma D) and, very often, results in overclassification. A key drawback of Density Peak is its bias toward selecting density peaks in dense regions, often neglecting sparse regions, which leads to two issues: sparse regions may be incorrectly assigned to distant clusters, and dense regions may generate multiple closely located peaks, causing most border points to have high densities and resulting in their misclassification as outliers. Additionally, DPC's computational cost increases with larger datasets, and despite enhancements like adaptive thresholds, it remains limited in effectively handling non-uniform or noisy data [29].

Previous studies suggest that such oversampling techniques based on clusters, including k-means or density-based, can identify clusters well and handle both intraclass and interclass imbalance simultaneously. Significant challenges arise when applying these techniques to datasets with varying data distributions, sparse clusters, or high variance. Fig. 1. represents the various datasets with variation in density of data points. Such characteristics are common in medical datasets, where critical minority-class instances are often scattered across large feature spaces or embedded in low-density regions. Low-density clusters, which often hold valuable information, and minimal clusters, common in highly imbalanced datasets, are challenging to identify effectively using existing methods. The density-based algorithms fail to handle data with high variance or a wide spatial distribution because they tend to focus on dense regions and may miss sparse but important areas. In addition, the performance of these algorithms largely relies on manual parameter tuning. The necessity to optimize parameters like k in k-means or the epsilon and minimum point thresholds in DBSCAN introduces complexities. It limits the scalability and robustness of these methods and makes it necessary to evaluate clustering algorithms based on the size of datasets and

Manuscript received December 8, 2024; Revised February 10, 2025; Accepted March 1, 2025; Available online June 5, 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.835

their distribution and inherent structure to determine which may be best suited.

To address these issues and to examine how dataset size and distribution affect clustering performance, there is a need for nature-inspired flow-based clustering methods, such as Percolation, that can capture a wider variety of samples by encompassing both dense and spread-out areas. This method can ensure that all the clusters identified are comprehensive and accommodating datasets with non-uniform density distributions. Unlike density-based methods, which rely heavily on predefined density thresholds, a percolation clustering method can dynamically adapt to data structures based on local connectivity. Furthermore, the dependency on parameter selection, a critical limitation in current density-based algorithms, must be reduced or automated to improve clustering reliability [25]. Percolation clustering methods can address these issues and work for both high-variance datasets and sparse datasets. Percolation methods do not assume uniform density within clusters. At a critical percolation threshold, an infinite cluster is formed, spanning the lattice or network [30]. They rely on the similarity of data points, allowing them to handle clusters with different densities effectively. Moreover, in some applications, rare and scattered points are of interest, particularly from a knowledge discovery point of view. Classical algorithms like K-means and DBSCAN [31] fail because of fixed-density assumptions and spherical shapes of clusters. K-means fails in cases of scattered minority instances because majority clusters absorb them, whereas DBSCAN sometimes classifies them as noise based on low density. Percolation-based methods, in contrast, focus on connectivity rather than density or shape, making them highly effective in identifying sparse, irregular clusters and isolating meaningful outliers.

This research aims to propose a novel algorithm that is applied to imbalanced medical datasets with different data distributions and compare them with the existing clustering method. The objective is to understand how proposed clustering enhances the quality of synthetic data generation and handle noise and outliers and thus performance of classification models is improved.

III. Formal mathematical analysis:

SMOTE [16] was developed to reduce the overfitting problem that arises with Random Oversampling (ROS). It produces synthetic samples(x_{new}) through interpolation between existing minority class data point(x_i) and one of its randomly selected neighbors($x_{neighbor}$). A synthetic sample is generated along the line segment connecting the original instance with one of its chosen neighbors. This is done by selecting a random point in the line segment using the formula Eq. (1) as follows [15]:

$$x_{new} = x_i + \lambda \times (x_{neighbor} - x_i) \text{ where } \lambda \in [0, 1]. \quad (1)$$

SMOTE, though, has shown good efficacy, but when the minority class is difficult to learn or when the generated samples introduce noise, it potentially leads to overfitting [33]. The adverse effect of complex minority classes on SMOTE can be formally investigated as follows:

Lemma A (Outlier): SMOTE exacerbates the impact of outliers by generating synthetic samples in low-density regions, increasing noise and degrading class separability.

Context and Assumptions:

Let $R \subseteq R^d$ represents the outlier region for the minority class, where the density of the minority class $p(x|y = 1)$ is significantly lower than the density of the majority class $p(x|y = 0)$, i.e. $(x|y = 1) \ll p(x|y = 0), \forall x \in R$.

Let $x_{outlier} \in R$ be a minority class outlier and $x_{neighbor} \in R^d$ a randomly selected minority class neighbor.

SMOTE generates synthetic samples such as Eq. (2)as given below [4] [5]:

$$X_{new} = x_{outlier} + \lambda \cdot (x_{neighbor} - x_{outlier}), \lambda \in U [0, 1] \quad (2)$$

Proof:

1. **Convex Combination in the Outlier Region:**
   The synthetic point x_{new} is a convex combination of $x_{outlier}$ and $x_{neighbor}$. Since $x_{outlier} \in R$ and SMOTE relies on $\lambda \in [0, 1]$, it follows that $x_{new} \in R$. Therefore, SMOTE generates additional synthetic samples in R, increasing the density of minority class points in this low-density region.
2. **Amplification of Class Ambiguity**
   R is characterized by $p(x|y = 1) \ll p(x|y = 0)$. The synthetic points x_{new} increase the minority class density in R, leading to
   $p(x|y = 1) \rightarrow p(x|y = 1) + \Delta p, \forall x \in R$, where Δp is the contribution of the synthetic points. This artificial increase in $p(x|y = 1)$ makes it comparable to $p(x|y = 0)$, causing the classifier to misinterpret R as a region of class overlap.
3. **Degradation of Decision Boundary** .
   The synthetic samples generated near $x_{outlier}$ are not representative of the true minority class distribution. Instead, they increase noise in R, leading to a blurred decision boundary between the minority and majority classes. Consequently, the classifier's ability to distinguish between classes deteriorates.

Thus, SMOTE exacerbates the problem caused by outliers by generating synthetic points in low-density, noisy regions .This increases class ambiguity and degrades the classifier's performance.

Manuscript received December 8, 2024; Revised February 10, 2025; Accepted March 1, 2025; Available online June 5, 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.835

Table 1. Comparison of Clustering Methods

points	percolation	k-means	DBSCAN
Flexibility in Cluster Shapes	It Can handle clusters of varying shapes and sizes by connecting data points based on proximity. It can adapt to natural cluster structures	K-means assumes that clusters are roughly spherical and evenly dense.	It is capable of identifying clusters of arbitrary shape and size
Adaptability to Varying Densities	It can deal with non-uniform density distributions .Percolation-based methods assume that the data is sparse and that the clusters are connected in a way that allows them to "percolate" through the data	It assumes the data is uniformly dense .It is less effective for non-uniform density distributions.	Well-suited for high variance datasets, particularly those with varying densities
Manually setting parameter values	Percolation methods do not require manual selection of cluster centers. They dynamically form clusters depending on the connectivity of data points,	It requires the manual initialization of cluster centers. It requires a pre-specified number of clusters.	It requires manual setting of eps (distance threshold) and minimum samples (minimum number of points in a neighborhood).

Lemma B (Noise): Let $x_i \in R^d$ be a minority class data point, and its noisy version be $x_{inoisy} = xi + \epsilon$, where $\epsilon \sim N(0, \sigma^2 I)$ is Gaussian noise. Let $x_{neighbor}$ be a true minority class neighbor of x_i. Then the synthetic sample generated by SMOTE as given in Eq (3) [4][15].

$$x_{new} = x_{inoisy} + \lambda \cdot (x_{neighbor} - x_{inoisy}), \lambda \sim U(0, 1) \quad (3)$$

lies outside the true minority class distribution with non-zero probability.

Proof: Substitute $x_{inoisy} = x_i + \epsilon$ into the SMOTE equation Eq (4) [5][15].:

$$x_{new} = (x_i + \epsilon) + \lambda \cdot (x_{neighbor} - (x_i + \epsilon)) \quad (4)$$

Simplify: Eq (5) [5].

$$x_{new} = x_i + \lambda (x_{neighbor} - x_i) + (1 - \lambda) \epsilon \quad (5)$$

The term $(1 - \lambda) \epsilon$ introduces noise independent of the minority class manifold. Since $\epsilon \sim N(0, \sigma^2 I)$, x_{new} deviates from the true minority class distribution. Thus, the probability of x_{new} lying outside the true minority class is non-zero.

Lemma C (Small Disjuncts): SMOTE overfits small disjunct regions of the minority class. Let $S_j \subset R^d$ represents a small disjunct region of the minority class such that $|S_j| \ll |D|$. Let $x_i, x_{neighbor} \in S_j$. The synthetic sample generated by SMOTE as given in Eq. (6) [5][15].

$$x_{new} = xi + \lambda \cdot (x_{neighbor} - x_i), \lambda \sim U(0, 1) \quad (6)$$

remains confined to Sj , leading to overfitting.

Proof:

Since xi and $x_{neighbor}$ both belong to S_j , the line segment as given by Eq (7) as follows [15].

$$L = xi + \lambda \cdot (x_{neighbor} - xi) \mid \lambda \in [0, 1], \quad (7)$$

is entirely contained in S_j . Therefore, the synthetic sample $x_{new} \in S_j$, reinforcing this isolated region without generalization.

Lemma D (Class Overlap): SMOTE amplifies class overlap by generating synthetic samples within ambiguous regions. Let $R \subset R^d$ be a region of class overlap such that: $p(x \mid y = 0) \approx p(x \mid y = 1), \forall x \in R$. For $x_i \in R$ (minority class) and $x_{neighbor} \in R$, the synthetic point generated by SMOTE in Eq. (8) [5][15].

$$x_{new} = x_i + \lambda \cdot (x_{neighbor} - x_i), \lambda \sim U(0, 1) \quad (8)$$

remains in R, increasing class ambiguity.

Proof:

The synthetic point x_{new} is a convex combination of $xi \in R$ and $x_{neighbor} \in R$. Since R is closed under convex combinations, $x_{new} \in R$. This increases the density of minority class points in R, amplifying the class overlap.

IV. The Proposed Method

This research considers the spatial distribution and density variability of datasets, including challenges by the outliers and heterogeneous data structures. In response to the limitations of conventional clustering algorithms this research has proposed a nature inspired flow-based clustering technique. This technique is based on percolation theory which handles a full range of data variability without just focusing on dense clusters. The proposed method, Percolation clustering with synthetic minority oversampling technique (PC-SMOTE), works in two stages: In the

Manuscript received December 8, 2024; Revised February 10, 2025; Accepted March 1, 2025; Available online June 5, 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.835

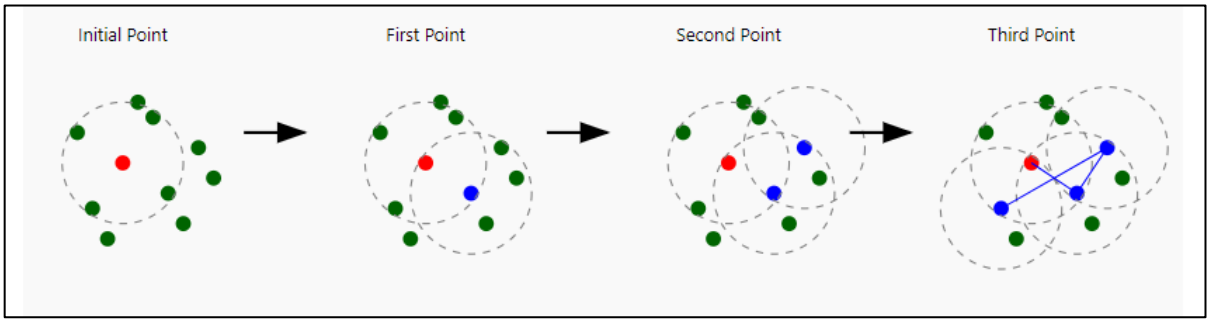


Fig. 2. Cluster formation using percolation analysis.

first stage it does generation of clusters using nature-inspired percolation-based theory that identifies natural clusters and filters outliers, and the second stage is then followed by oversampling using SMOTE applied to the refined the data points and lastly evaluating the method using several classifiers on the largest cluster. To provide a comprehensive understanding, we first present an overview of percolation theory in IV-A. This is followed by detailed steps for cluster identification using the percolation method in Sect. IV-B Finally, the complete PC-SMOTE algorithm is described in Sect. IV-C.

A. Percolation Theory Overview:

Percolation theory, introduced in 1957, describes the movement of a fluid through a network of interconnected channels (bonds) within a porous node represents data points, and edges signify relationships such as proximity or similarity. Percolation analysis, widely applied in physics, materials science, and network theory, identifies clusters as significant connected components that" percolate" through the system. Unlike traditional clustering methods, percolation does not assume uniform density within clusters. Instead, it forms clusters considering the proximity of data points, enabling effective management of clusters with varying densities. Percolation analysis identifies clusters within spatially distributed points using Euclidean distances. A distance threshold τ determines whether two points are" connected," meaning they belong to the same cluster. Fig. 2. illustrates the cluster formation technique, which begins with a randomly selected point and all neighboring points within τ are added to the cluster. This procedure is recursively repeated for newly added points until no further points meet the distance criterion, at which point the cluster is finalized. The term percolation threshold refers to the critical distance where connections between points form clusters [33][34].This technique forms clusters without any assumption on sizes, locations, or numbers, and it could, therefore, adapt to the natural structure of the data; this technique is also quite robust against noise, just disregarding irrelevant outliers up to later stages. By treating clusters as" connected components," percolation theory offers a dynamic framework for clustering datasets composed of randomly distributed points in feature space [35]. Table 1 compares the percolation method with the existing clustering methods such as k-means and DBSCAN based on parameters. This comparison shows that Percolation Clustering outperforms K-Means and DBSCAN by effectively identifying clusters of diverse shapes and adapting to non-uniform or sparse density distributions which is often the case in medical datasets.

B. Percolation based clustering technique:

To understand the formation of clusters with respect to percolation theory we have considered imbalanced medical datasets that are used to identify the clusters and outliers in a 2-dimensional feature space by interpreting the data points as nodes and their pairwise relationships (e.g., distances or similarities) as links Specifically, percolation-based clustering formation is explained in the following six key steps:

Step 1: Distance Matrix Construction:

Given a binary class dataset X = {x1, x2, . . ., x_n} where xi $\in$ R^2 , the dataset contains majority and minority class data points. For each pair of points (x_i, x_j), calculate the Euclidean distance as given by Eq (9) [34].

$$d(x_i , x_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2} \quad (9)$$

Store these distances in matrix D, where Eq(10)[34]

$$D [i,j] = d (x_i, x_j) \quad (10)$$

Step 2: Modeling the Dataset as a Graph.:

Each point $x_i = (x_{i1}, x_{i2})$ represents a node in the graph, with links established between nodes based on a distance matrix D [i,j]. Initially, the dataset is modeled as a graph which is fully connected, and the nodes

Manuscript received December 8, 2024; Revised February 10, 2025; Accepted March 1, 2025; Available online June 5, 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.835

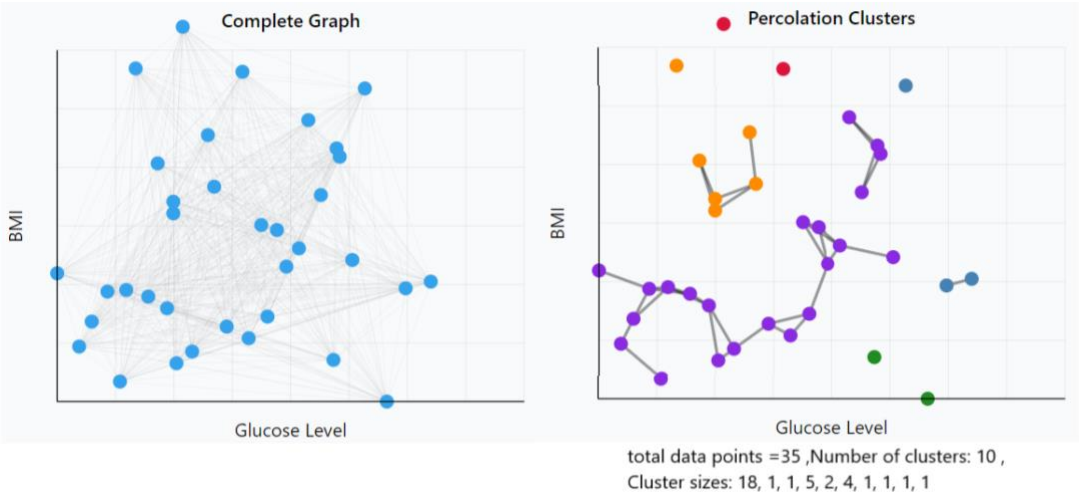


Fig. 3. Initial graph formation and clustering analysis

correspond to the elements of x as shown in Fig. 3. and the links are weighed according to the distance matrix D [i,j]. It acts as a grid where edges connect each point (vertex) to its neighboring points.

Step 3: Percolation Threshold:

The percolation threshold is a critical value that determines the connectivity of nodes. The percolation threshold critical distance value 'τ' at which clusters transition from being small and disconnected to forming significant, connected components. The pilot value of the percolation threshold is calculated using Connectivity Entropy, a statistical metric that quantifies the uncertainty or randomness in the connectivity patterns cluster or network. Connectivity entropy is fundamentally derived from the principles of Shannon entropy. The pilot value is further appropriated experimentally. Connectivity Entropy is given as follows in Eq (11) [34].

$$H(\tau) = -\sum_{k=1}^{k} p_k log(p_k) \quad (11)$$

where k is number of clusters formed at a given 'τ', p_k is given as ratio of $\frac{n_k}{N}$ points in clusters k, n_k is number of data points in clusters k and N is total number of data points. It then retains only distances d (x_i, x_j) ≤ τ, representing edges between nodes. The adjacency matrix A is also termed as the connection matrix and demonstrates the connection between x_i and x_j of a graph dataset (i,j). If two nodes are connected, the value of the adjacency matrix is set to 1 and considered to be set as 0 for no connection. This matrix A in Eq (12) [34].

$$f(x) = \begin{cases} -1, & if\ D[i,j] \leq \tau \\ x, & otherwise \end{cases} \quad (12)$$

represents an undirected graph as given in the next step.

Step 4: Graph construction based on Adjacency matrix A: Use the adjacency matrix A to construct a graph G = (V, E), where: V is the set of nodes corresponding to points in X. E is the set of edges derived from A. The process for constructing clusters is as follows:

1. Start with an unvisited node $x_i \in V$. Check all nodes x_j connected to xi in A (i.e., A [i, j] = 1). Add all connected nodes x_j to the current cluster.
2. Repeat this process iteratively for each newly added node until no further nodes can be included.
3. Mark the visited nodes and repeat the process for the remaining unvisited nodes in V

Step 5: Cluster Formation. :

We will perform a depth-first search (DFS) to find connected components. A connected component C(v) of the graph is defined as: C(v) = {u ∈ V | ∃ a path from v to u in G}. Clusters are formed by identifying all connected components in G:

1. Initialize the set of all points S = {v1, v2, . ., v_n} and an empty set of clusters C = { }.
2. While S ≠∅: Pick a node v from S and find C(v), the connected component containing v. Add C(v) to the set of clusters C. Remove all nodes in C(v) from S.

Step 6: Filter Clusters and Identify Outliers.

- The resulting set C contains all identified clusters.
- Nodes that do not belong to any cluster of sufficient size (based on a minimum size criterion) are classified as outlier.

The complete process explained in the above steps

Manuscript received December 8, 2024; Revised February 10, 2025; Accepted March 1, 2025; Available online June 5, 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.835

starts by modeling the imbalanced binary class dataset with varying density distributions as a fully connected graph, where the data point is represented as a node, and edges between the nodes are weighted by their pairwise distances. The distances are then stored in the distance matrix D_{ij} , and preprocessing may be necessary to ensure these distances are meaningful and appropriately scaled. To apply percolation, a threshold 'τ' is defined; edges with weights greater than 'τ' (distances exceeding the threshold) are removed, leaving only connections between nodes that are sufficiently close. The connections are represented by an Adjacency matrix A[i,j]. As the threshold 'τ' is progressively adjusted, clusters emerge as groups of connected nodes. The percolation here refers to how these edges propagate through the graph and form large, connected clusters, where each cluster represents a set of data points that are sufficiently close to each other. According to percolation theory, most nodes will remain within a Giant cluster for higher thresholds, while nodes outside the Giant cluster distribute into smaller disconnected clusters. The Giant cluster formed is then considered the primary structure of the data, while smaller disconnected components often represent outliers or distinct clusters. This approach dynamically identifies clusters and outliers based on the connectivity and distance relationships within the dataset. These typically lie in sparse regions and exhibit low local connectivity. The criteria rely on pairwise Euclidean distance points, which are considered outliers if they do not fall within the connectivity graph formed when distances are below τ. Fig. 3. represents the fully connected graph of data points and clusters formation.

The Giant Cluster, after filtering, undergoes oversampling through SMOTE technique. Due to SMOTE random nature of selecting a data point for interpolation may generate outliers or noisy points which may potentially introduce additional noise into the dataset. SMOTE basically does not consider within-class imbalance issues. In regions of the feature space where minority class samples are densely clustered, more synthetic samples are likely to be generated. Conversely, sparse regions often remain underrepresented, leading to uneven sample distribution and potentially reducing model effectiveness. The challenges need to be overcome, and for that, it becomes very important to identify a meaningful subset of data points that can effectively contribute toward the generation of synthetic samples. In this regard, the Percolation Cluster-based. Oversampling method addresses the aforementioned concerns by identifying relevant subsets of data points to improve synthetic sample generation. Algorithm 1 of the proposed method is given below.

C. Percolation cluster-SMOTE algorithm (PC-SMOTE):

The proposed approach focuses on improving the oversampling algorithm, which is SMOTE, by integrating a percolation-based clustering mechanism to guide the oversampling process. This proposed method generates synthetic data samples in safe zones, and it covers both within class and between

Algorithm 1: PC-SMOTE

Require: Dataset $X \in \mathbb{R}^{nxd}$, Labels $y \in \{0$ (majority), 1 (minority)$\}$
Ensure: Augmented dataset X_aug

1. Compute the pairwise Euclidean distance matrix D
$$d(x_i , x_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2}$$
2. Percolation Graph Construction:
Create adjacency matrix A using threshold τ:
$$f(x) = \begin{cases} -1, & \text{if } D[i,j] \leq \tau \\ x, & \text{otherwise} \end{cases}$$
3. Cluster Detection Using DFS over A, find connected components:
C={C1,C2,…,Cm},where |Ci|≥θ
4. To Identify the cluster with the maximum number of total data points containing both classes:
$C = \arg\max_{ci \in C} \{| Ci || \exists$ xj, xk∈Ci,yj=0,yk=1$\}$
5. SMOTE within Giant Cluster:
Let M = $\{x_i \in C^* \mid y_i = 1\} \subseteq$ minority samples within C*
for each $x_i \in M$ do
Find k-nearest neighbors $N_k(x_i) \subseteq M$
for each x_neighbor $\in N_k(x_i)$ do
Generate synthetic sample:
$x_{new} = x_i + \lambda \cdot (x_{neighbor} - x_i)$,
where $\lambda \sim U(0, 1)$
Append x_new to synthetic set X_synt
end for
end for
6. Augment dataset:
$X_{aug} = X \cup X_{synt}$
return X_aug

class imbalances through strategy. The PC-SMOTE method achieves three important objectives: it improves the distribution of synthetic samples by capturing diverse regions, including both dense and sparse areas, resulting in a more representative synthetic dataset; it enhances cluster-based representation by adapting the oversampling process to datasets with sparse or less dense distributions, preserving the data structural integrity mitigates noise

Manuscript received December 8, 2024; Revised February 10, 2025; Accepted March 1, 2025; Available online June 5, 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.835

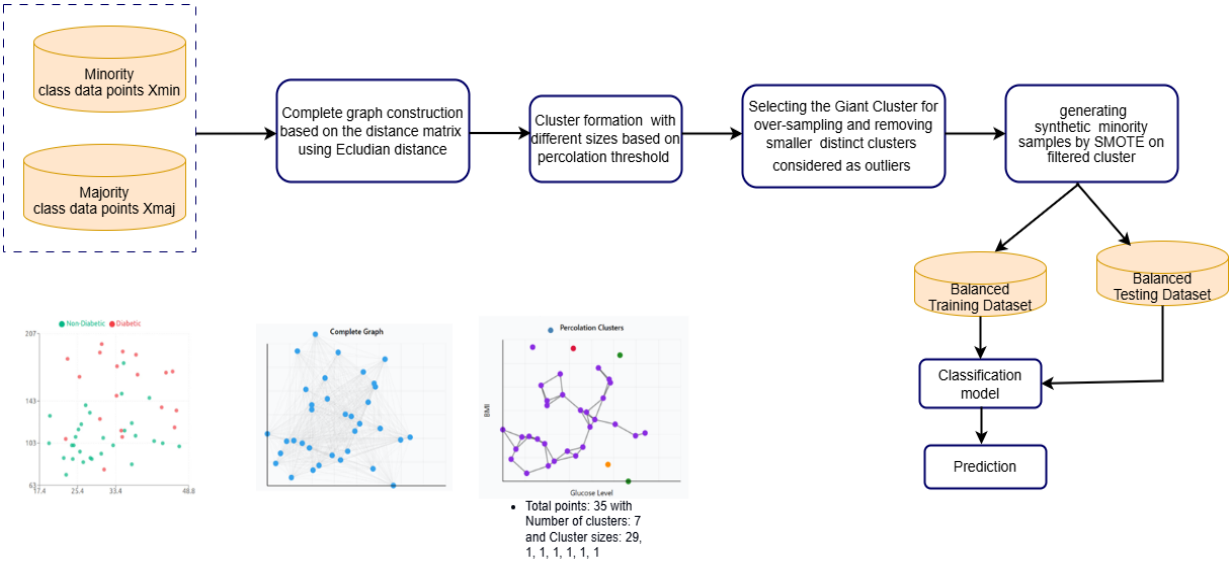


Fig. 4. Workflow of proposed PC-SMOTE method

amplification by avoiding the generation of synthetic samples in noisy or unsafe areas, thereby reducing the risk of introducing erroneous patterns. Fig. 4. represents the proposed PC-SMOTE workflow. The steps to generate high-quality synthetic minority.

1. The binary class imbalance dataset X = {x_1, x_2, . . . , x_n } which contains majority and minority class data points, is given to the proposed percolation-based clustering method to partition the imbalance training set into clusters of different sizes and densities. A graph-like structure is formed where nodes (data points) are connected by edges based on similarity, creating clusters of varying sizes.
2. Data points not connected to any clusters considered as outliers and clusters that are too small are identified and removed from the dataset.
3. The distribution of minority and majority data points is taken from the selected giant cluster. This helps to identify and find the required count of synthetic samples to be generated in the giant cluster.
4. The SMOTE method (explained in section III)is applied to the selected giant cluster to generate the required number of minority data points.

Through the above steps, the proposed algorithm PC-SMOTE method generates high-quality synthetic minority class samples, effectively balancing the dataset while addressing class imbalances and removing outliers. The percolation process connects instances based on a distance threshold. If a point is reachable under this threshold, it is assigned to a cluster. SMOTE selects only the minority class instances within the giant selected cluster as seeds for generating synthetic data. A synthetic sample is generated by linear interpolation between the seed and one of its neighbors. Within the giant cluster, SMOTE selects well-supported minority samples in dense areas to generate synthetic data, reduce noise, and preserve intrinsic data structure. After that, a given classifier is trained effectively on the improved class imbalance training set to achieve better classification performance.

V. Experiment Design

A. Datasets

1. Real medical Dataset

To evaluate the proposed method PC-SMOTE, we have taken eight different medical binary labeled datasets of different imbalance ratios. UCI repository is referred to as clinical datasets. In binary labeled datasets, minority class are assigned a label of 1, while the majority class is labeled as 0. These datasets are chosen to encompass varying levels of imbalance ratios. Characteristics of all the datasets involved in the experiment are shown in Table 2.The table has columns such as count of total minority class samples, and the count of total majority class samples for a dataset, number of features, the total samples in the dataset and imbalance ratio. Imbalance ratio (IR) is considered as the ratio of the count of positive data points compared with negative data points[37], which is calculated by Eq (13) [15].

$$\text{IR} = \frac{\text{Total count of positive instances}}{\text{Total count of negative instances}} \quad (13)$$

2. Synthetic Dataset

To examine the proposed method for different nonuniform data distributions and address both inter

Manuscript received December 8, 2024; Revised February 10, 2025; Accepted March 1, 2025; Available online June 5, 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.835

Table 2. Imbalanced Medical Datasets Characteristics

Sr. No	Dataset	Features	Samples	Minority Samples	Majority Samples	Imbalance Ratio (IR)
D1	Heart	13	270	120	150	1.25
D2	Haberman's Survival	3	306	81	225	2.78
D3	Parkinson's Disease Dataset	23	197	48	147	3.06
D4	Pima Diabetes	8	768	268	500	1.87
D5	Heart Failure Clinical Records	12	299	96	203	2.11
D6	Cervical Cancer	35	858	55	803	14.60
D7	Breast Cancer Wisconsin (F15)	9	699	241	458	1.90
D8	Indian Liver Patient Dataset (F14)	10	583	167	416	2.49

class imbalance and intra class imbalance, including outliers' issues, we have created three 2-dimensional datasets. All these synthetic generated imbalanced datasets, which are represented as DS1, DS2, and DS3 having an imbalance ratio in the range of 2.0 to 3.5, and the variability is considered to be high, medium, and low, which shows the spreads in the data density. These datasets have data characteristics such as outliers, different densities, and data distributions. Table 3 gives a summary of the synthetic dataset. Fig. 5. gives the visualization of these created synthetic datasets with different densities. For these synthetic datasets, the percolation clustering algorithm is applied, clusters are formed, and further oversampling technique is applied to balance the cluster minority samples. Then, the balanced dataset is given to the classifiers, and performance is evaluated on the test datasets, and metrics such as F1score, AUC,G-Mean and PRAUC are calculated to determine the top-performing classifiers.

3. Experimental Design

To evaluate the efficacy of our approach, comparative analysis is conducted for the proposed clustering algorithm with existing cluster algorithms used in combination with SMOTE oversampling techniques, such as K-means+ SMOTE[21] and DBSMOTE[22],which are popular clustering methods. or this, we used 3 different classifiers and 4 different evaluations metrics.

Table 3. Synthetic Dataset Characteristics

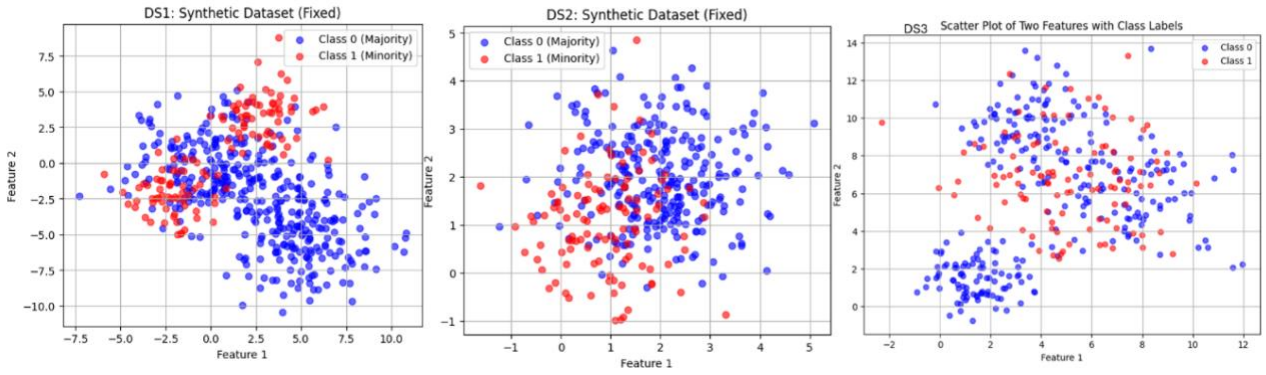
Dataset	Variability in Density/Spread	Data points	Imbalance Ratio (IR)
DS1	High	350	3.5
DS2	Medium	378	2.4
DS3	Low	375	2.6

The experiment carried out uses the default hyperparameter further; in the case of SVM, probability=True and random state=42 is set, and for Random Forest, random state=42 is used to ensure reproducibility. We conducted the experiment on both real medical datasets and synthetically created datasets. The default configuration of the KNN classifier was used without tuning. Specifically, we used 8 real medical datasets and 3 synthetic datasets. In this experiment, 70% of the original data is used as training data points, while the remaining 30% of the samples are used to test the classifiers. Along with that, a 10-fold cross-validation is used to produce an unbiased result. The experiments taken were executed on a 2GHz CPU with 16GB RAM. Table 4 provides the setting values of experimental parameters. For SMOTE, the need for nearest neighbors (NN) of minority class to be found for each minority data point to generate synthetic samples can be set to 3 or 5. For the k-means method, the parameter k, which denotes the cluster number, is set to either 3 or 5. Similarly, for clustering with DBSCAN, both parameters such as ϵ represents the maximum radius that defines a point's neighborhood, while *MinPts* denotes the minimum number of neighboring points required to be tuned to obtain the expected clustering result. For specific

Table 4. The parameters of comparison methods

Technique Used	Parameter	Value
K-means	Number of clusters (k)	3, 5
DBSCAN		2, 3, 4, 5
	Eps or (ε)	1.5, 2.0
SMOTE	Nearest neighbor (r)	3, 5
Percolation	Percolation threshold (τ)	1.5, 2.5

Manuscript received December 8, 2024; Revised February 10, 2025; Accepted March 1, 2025; Available online June 5, 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.835



DS1 DS2 DS3

Fig. 5. Synthetic dataset with low(DS3),medium(DS2) and high(DS1) density.

datasets, such as New- Thyroid1, Cervical Cancer, and Liver Patient datasets, appropriate adjustments are made with specific k values to achieve optimal performance.

4. Evaluation of datasets

The classifiers used to assess performance such as KNN and RF were selected for their robustness to handle imbalanced datasets effectively, [37]. SVM with a radial kernel was included because it effectively classifies non-linearly separable classes. These performances of classifiers are measured on the test data by using F1score, G-mean, AUC and PR AUC; the classification accuracy parameter is not considered to be a proper metric for imbalanced data, as it tends to favor majority predictions while overlooking errors in minority class predictions, making it insensitive to imbalances in the data. The confusion matrix used for binary datasets are shown in Table 5 consists of the following parameters. True positive(TP) is the count of real positive samples that are correctly predicted as positive. False Negative (FN) is the count of real positive samples that are incorrectly predicted as negative False Positive (FP) is the count of real negative samples that are incorrectly predicted as positive. True Negative (TN) is the count of real negative samples that are correctly predicted as negative. These specified parameters of the confusion matrix help to find other metrics such as F1,Precision and Recall. F1-score is the weighted average of Precision and Recall, calculated using their harmonic means where the higher value indicates that the model can effectively classify the positive samples, which indicates high accuracy for the minority class. Recall can also be referred to as Sensitivity, which is the percentage of the actual positive data samples that the model correctly picks up. Precision, on the other hand, is the percentage of accurate positive predictions among all the instances that the model has marked as positive. The model's ability to predict both negative and positive samples is evaluated by G-mean, or geometric mean. The high value of the G-mean implies that a classifier is suitable for both binary label classes without showing bias toward either class. Area Under the Receiver Operating Characteristic Curve (AUC-ROC): The recall vs. false positive rate plot is termed the receiver operating curve(ROC) for different thresholds. The area under the ROC is termed as the AUC. If the value of AUC is closer to 1, the model's performance is better, and the model shows a stronger ability to distinguish between positive and negative classes.PR-AUC represents the area under the curve of Precision versus Recall.

Table 5. Confusion matrix

	Predictive Positive	Predictive Negative
Real Positive	True positive	False negative
Real Negative	False positive	True negative

VI. Experimentation Results

To compare and examine our proposed method PC-SMOTE and the other clustering methods such as k-means+SMOTE[22] and DBSMOTE[23] on 8 imbalanced medical datasets with different metrics are shown in Table 6 .The mentioned classifiers are used for training and testing as they are widely recognized and commonly employed in medical detection applications.

A. Synthetic data results analysis

To Compare and examine our proposed method PC-SMOTE and the other clustering methods such as k-means SMOTE[22] and DBSMOTE[23] we have

Manuscript received December 8, 2024; Revised February 10, 2025; Accepted March 1, 2025; Available online June 5, 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.835

created 3 synthetic datasets. The reason for creating such test datasets is to have control to generate datasets that are not dense or concentrated and have non-uniform data distribution. The proposed method can form clusters of varying densities and then can be evaluated for the adaptability of clustering methods to sparse and dense regions. The dataset DS3 as the variability in data distribution is high the performance of the proposed method is better for all evaluation metrics as shown in Table 7.From the evaluation metrics reading taken for different clustering algorithms for the dataset which is taken to be sparse and varying distribution the density-based clustering algorithm even though tends to make irregular shape clusters have less improvement for these datasets. Fig. 6,7 and 8 shows the performance for three different classifiers .

B. Real dataset result analysis

The evaluation results for the 8 medical imbalance datasets are shown in Table 6. The performance of the three selected classifiers for the k-means SMOTE, DBSMOTE and PC-SMOTE methods is described using metrics AUC,F1-Score , G-mean and AUPRC . The highlighted numbers in the table indicate that those datasets achieved the best performance. By comparing it with other clustering-based methods it was found that PCSMOTE has the best performance in most of the medical datasets with regard to at least one of three performance metrics. Specifically, the PC-SMOTE method works excellent for Breast cancer dataset, Parkinson dataset and Cervical cancer dataset where AUC is in the range is 96% to 99%, which is high compared to the other two methods. This demonstrates the effectiveness of the PC-SMOTE algorithm in handling datasets with both low and high imbalance ratios. Fig. 9 and Fig.10, shows the performance metrices of various methods for three different classifiers.

C. Reliability and Time Complexity

It is also crucial to evaluate the reliability of the results. To assess this, we conducted a cross-validation folds t-test, which is an excellent way to demonstrate PC-SMOTE's stability and robustness. We ran a 5-fold cross-validation and obtained performance metrics, Including F1-Scores, for various sample datasets: Heart, Parkinson's Disease, and Pima Diabetes. The T-test was conducted for the classifiers Random Forest (RF), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN). In the case of the Heart dataset for F1-score test score were, RF had 2.45(T-stat), 0.0368(p-value), SVM had 34.76(T-stat), 6.67e-11(p-value) while KNN 4.36(T-stat), 0.012(p-value). Similarly, for the Parkinson's Disease Dataset, RF T-stat of 2.64, a p-value of 0.050, SVM had a T-stat of 4.40, a p-value of 0.010, while KNN had a T-stat of

Table 6. Performance comparison of different methods on medical dataset

Data sets	Methods	RF				SVM				KNN			
		AUC	F1-score	G-mean	PR-AUC	AUC	F1-score	G-mean	PR-AUC	AUC	F1-score	G-mean	PR-AUC
D1	K-means SMOTE	0.87	0.66	0.72	0.85	0.89	0.76	0.80	0.88	0.88	0.70	0.74	0.85
	DBSMOTE	0.93	0.88	0.87	0.94	0.93	0.86	0.86	0.93	0.91	0.89	0.89	0.91
	PC-SMOTE	0.91	0.83	0.81	0.90	0.75	0.68	0.69	0.74	0.75	0.70	0.69	0.71
	SMOTE	0.90	0.80	0.80	0.88	0.73	0.65	0.68	0.73	0.69	0.60	0.62	0.67
	Imbalance	0.90	0.79	0.80	0.89	0.72	0.57	0.62	0.72	0.69	0.60	0.63	0.68
D2	K-means SMOTE	0.66	0.35	0.49	0.38	0.71	0.43	0.56	0.45	0.66	0.45	0.56	0.54
	DBSMOTE	0.88	0.76	0.78	0.84	0.76	0.65	0.67	0.70	0.78	0.69	0.71	0.73
	PC-SMOTE	0.83	0.77	0.75	0.84	0.80	0.70	0.69	0.78	0.79	0.77	0.74	0.80

Manuscript received December 8, 2024; Revised February 10, 2025; Accepted March 1, 2025; Available online June 5, 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.835

	SMOTE	0.61	0.38	0.54	0.36	0.71	0.39	0.53	0.50	0.63	0.40	0.56	0.40
	Imbalance	0.66	0.29	0.44	0.42	0.71	0.00	0.00	0.51	0.64	0.31	0.43	0.45
D3	K-means SMOTE	0.90	0.97	0.84	0.97	0.98	0.97	0.84	0.99	0.83	0.94	0.65	0.96
	DBSMOTE	0.98	0.93	0.91	0.97	0.97	0.88	0.88	0.97	0.97	0.87	0.87	0.98
	PC-SMOTE	**0.97**	0.90	0.88	**0.97**	0.94	0.88	0.87	0.95	0.95	0.91	0.91	0.97
	SMOTE	0.91	0.88	0.83	0.91	0.68	0.82	0.60	0.84	0.87	0.86	0.77	0.95
	imbalance	0.88	0.79	0.83	0.85	0.72	0.81	0.48	0.87	0.85	0.89	0.68	0.91
D4	K-means SMOTE	0.80	0.64	0.71	0.66	0.80	0.63	0.71	0.70	0.76	0.64	0.71	0.60
	DBSMOTE	0.89	0.81	0.80	0.89	0.86	0.81	0.80	0.85	0.83	0.79	0.76	0.83
	PC-SMOTE	**0.90**	**0.83**	**0.83**	**0.90**	0.81	0.68	0.69	0.82	0.80	0.74	0.72	0.80
	SMOTE	0.82	0.65	0.73	0.68	0.76	0.62	0.62	0.71	0.72	0.58	0.66	0.61
	imbalance	0.82	0.64	0.71	0.70	0.72	0.57	0.65	0.71	0.73	0.55	0.64	0.63
D5	K-means SMOTE	0.86	0.63	0.69	0.82	0.80	0.58	0.65	0.77	0.74	0.55	0.62	0.76
	DBSMOTE	0.95	0.88	0.86	0.96	0.90	0.86	0.84	0.88	0.82	0.79	0.74	0.84
	PC-SMOTE	**0.98**	**0.92**	**0.93**	**0.97**	**0.85**	0.82	0.84	**0.85**	**0.84**	0.76	**0.79**	0.79
	SMOTE	0.90	0.76	0.82	0.81	0.44	0.21	0.33	0.31	0.43	0.32	0.44	0.32
	imbalance	0.89	0.73	0.79	0.80	0.49	0.00	0.00	0.35	0.44	0.22	0.35	0.32
D6	K-means SMOTE	0.97	0.62	0.70	0.71	0.94	0.53	0.63	0.51	0.92	0.42	0.54	0.58
	DBSMOTE	0.99	0.97	0.97	0.99	0.98	0.96	0.96	0.98	0.97	0.96	0.96	0.97
	PC-SMOTE	**0.99**	**0.98**	**0.98**	**0.99**	**0.89**	0.77	0.79	0.86	0.96	0.92	0.91	0.96
D7	K-means SMOTE	0.98	0.94	0.96	0.95	0.99	0.97	0.97	0.98	0.87	0.70	0.77	0.80
	DBSMOTE	0.01	0.98	0.98	0.30	0.01	0.98	0.98	0.33	0.01	0.97	0.97	0.28
	PC-SMOTE	**0.98**	0.97	0.98	0.98	0.44	0.62	0.63	0.48	0.69	0.60	0.60	0.63

Manuscript received December 8, 2024; Revised February 10, 2025; Accepted March 1, 2025; Available online June 5, 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.835

	SMOTE	0.90	0.84	0.86	0.89	0.41	0.36	0.59	0.59	0.67	0.60	0.60	0.56
	imbalance	0.93	0.83	0.85	0.93	0.68	0.35	0.47	0.56	0.71	0.61	0.28	0.36
D8	K-means SMOTE	0.80	0.38	0.53	0.56	0.71	0.00	0.00	0.46	0.62	0.43	0.60	0.39
	DBSMOTE	0.78	0.72	0.70	0.78	0.74	0.74	0.67	0.67	0.74	0.70	0.65	0.75
	PC-SMOTE	**0.86**	**0.79**	**0.75**	**0.86**	0.72	0.71	0.61	0.69	0.75	0.75	0.71	0.74
	SMOTE	0.73	0.45	0.59	0.48	0.67	0.53	0.59	0.48	0.67	0.49	0.62	0.47
	imbalance	0.73	0.33	0.47	0.49	0.69	0.02	0.04	0.44	0.66	0.36	0.51	0.45

3.71, a p-value of 0.020. For the PIMA Diabetes Dataset, RF had a T-stat of 4.33, a p-value of 0.0124, SVM had a T-stat of 3.22 and a p-value of 0.0322, while KNN had a T-stat of 2.30 and a p-value of 0.0470. Furthermore, a t-test was conducted on the G-means, and the results were analyzed for the classifiers RF, SVM, and KNN. In the case of the Heart dataset, RF had a T-stat of -1.0000, a p-value of 0.3434, SVM had a T-stat of 2.6090 and a p-value of 0.0283, while KNN had a T-stat of 3.17 and a p-value of 0.030. Similarly, for the Parkinson's Disease Dataset, RF had a T-stat of -4.17 and a p-value of 0.014, SVM had a T-stat of -4.964 and a p-value of 0.007, while KNN had a T-stat of -4.95 and a p-value of 0.007. For the PIMA Diabetes Dataset, RF yielded a T-stat of 0.480 and a p-value of 0.6429, SVM yielded a T-stat of -4.339 and a p-value of 0.0019, and KNN yielded a T-stat of -2.819 and a p-value of 0.0201. The T-test across classifiers and datasets revealed that PC-SMOTE achieves statistically significant performance differences in multiple cases. In particular, it performs significantly better on the Heart dataset with SVM and KNN. It shows notable performance deviations on the Parkinson and PIMA datasets, where, in some cases, Baseline outperforms PC-SMOTE. This highlights the need for selecting a method based on dataset characteristics.

Furthermore, time complexity is a critical factor in assessing its applicability in the real world. The PC-SMOTE involves two main computational components: a pairwise distance matrix and graph traversal. The Distance matrix involves computing distances between all data point pairs, leading to a time complexity of $O(n^2)$. The graph traversal employs standard algorithms such as Breaths First Search (BFS) or Depth First Search DFS have the time complexity as O(n+m), and 'n' is given as the number of nodes, and of edges. Compared with the baseline method,

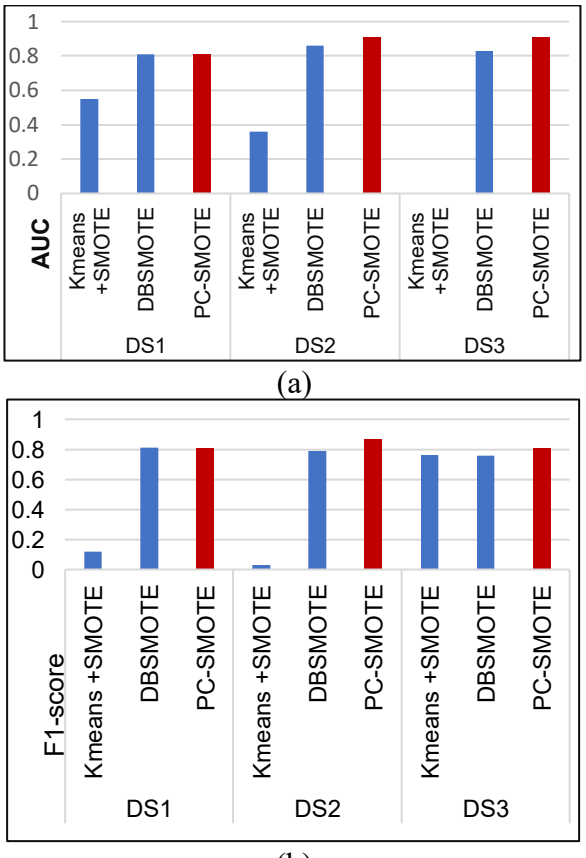


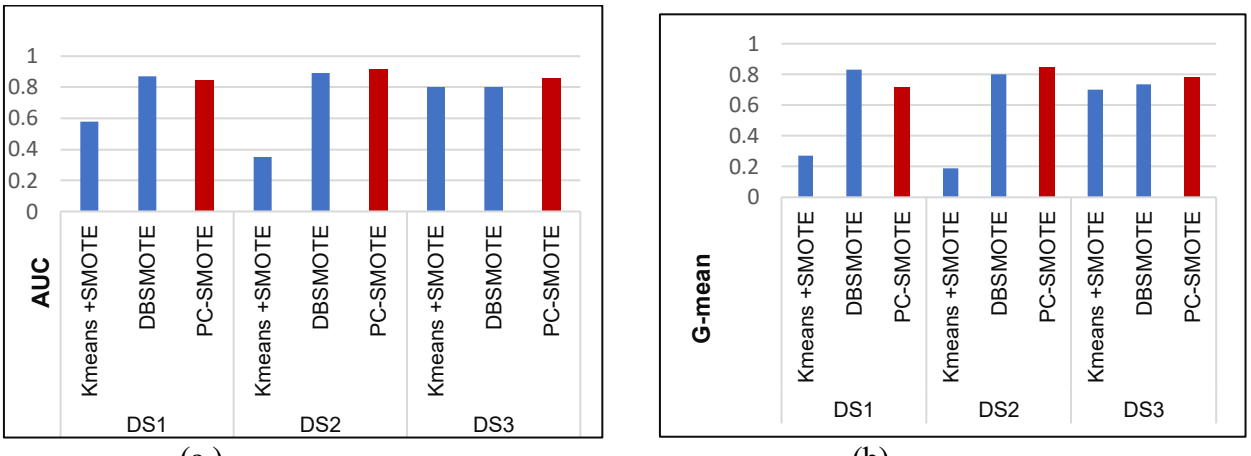
Fig. 6. The result for performance comparison of oversampling methods on Synthetic datasets (a) F1-score and (b)G-mean with RF classifier

DBSCAN with naive search yields $O(n^2)$, while K-means SMOTE, which is a combination of K-means and SMOTE with 'k' clusters and 't' iterations, comes as $O(nkt) + O(n^2)$. Empirically, we observed that the runtime of PC-SMOTE was comparable to that of DB-

Manuscript received December 8, 2024; Revised February 10, 2025; Accepted March 1, 2025; Available online June 5, 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.835

Table 7. Performance comparison of different methods on Synthetic dataset

Data sets	Methods	RF				SVM				KNN			
		AUC	F1-score	G-mean	PR-AUC	AUC	F1-score	G-mean	PR-AUC	AUC	F1-score	G-mean	PR-AUC
DS1	Kmeans +SMOTE	0.57	0.14	0.27	0.58	0.57	0.01	0.01	0.56	0.55	0.12	0.25	0.54
	DBSMOTE	0.87	0.83	0.83	0.86	0.81	0.79	0.79	0.79	0.81	0.81	0.81	0.79
	PC-SMOTE	**0.84**	0.70	0.72	0.82	0.80	0.81	0.80	0.73	0.81	0.81	0.80	0.73
DS2	Kmeans+SMOTE	0.35	0.08	0.19	0.41	0.37	0.03	0.12	0.42	0.36	0.03	0.12	0.40
	DBSMOTE	0.89	0.80	0.80	0.89	0.87	0.80	0.79	0.86	0.86	0.79	0.79	0.88
	PC-SMOTE	**0.92**	0.85	0.85	0.86	**0.90**	0.78	0.78	0.85	**0.91**	0.87	0.87	0.86
DS3	Kmeans+SMOTE	0.80	0.73	0.70	0.76	0.79	0.75	0.71	0.77	0.82	0.76	0.755	0.80
	DBSMOTE	0.80	0.73	0.73	0.77	0.79	0.75	0.75	0.77	0.83	0.76	0.76	0.81
	PC-SMOTE	0.86	0.79	0.78	0.86	0.83	0.78	0.73	0.78	**0.91**	0.81	0.80	**0.89**



(a) (b)

Fig 7: The results for performance comparison of oversampling methods on Synthetic datasets (a) AUC with the RF classifier, (b) G-mean with the SVM classifier.

Manuscript received December 8, 2024; Revised February 10, 2025; Accepted March 1, 2025; Available online June 5, 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.835

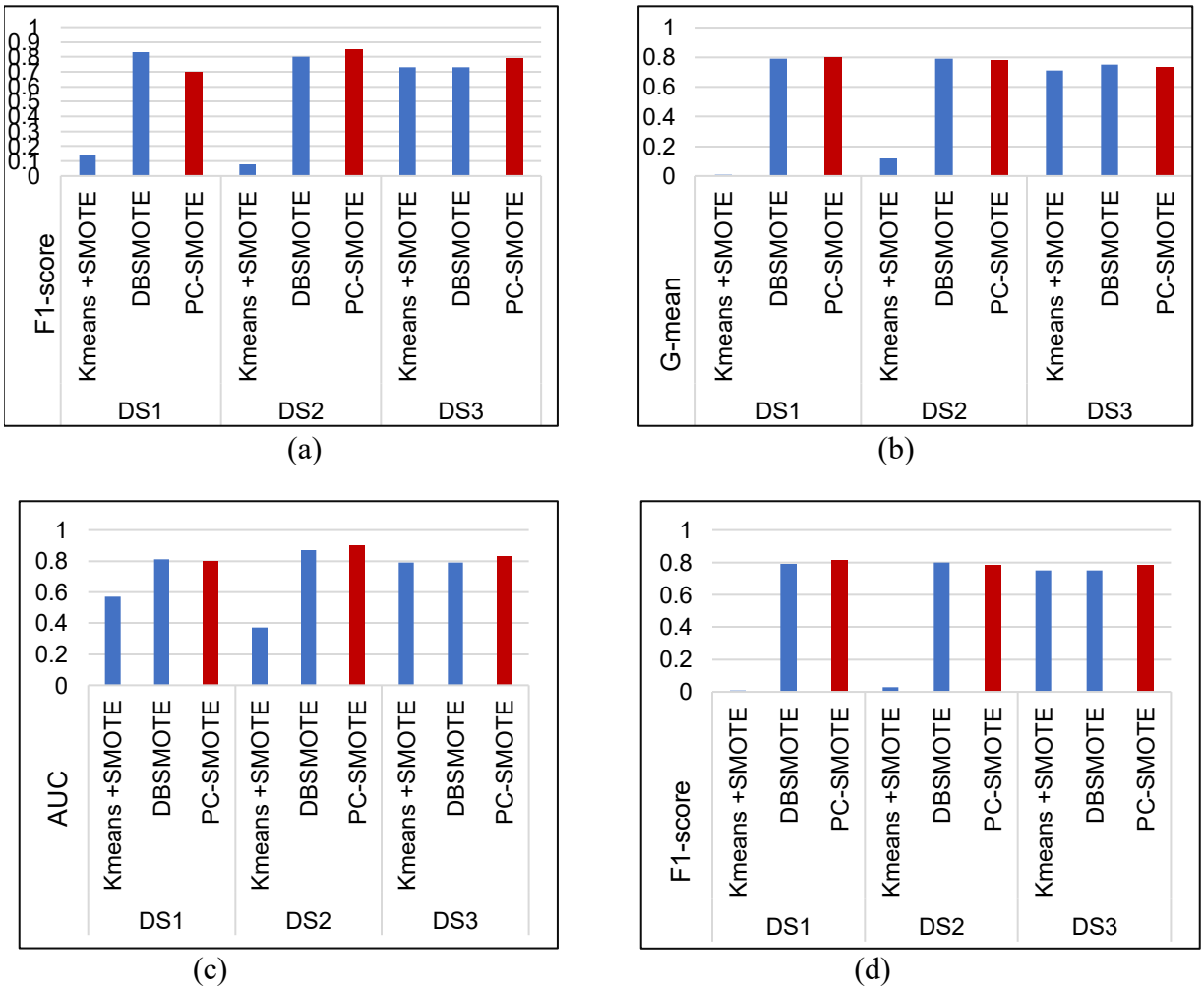


Fig. 8: The results for performance comparison of oversampling methods on Synthetic datasets (a) Accuracy and (b) G-mean with the SVM classifier, (c) Accuracy and (d) G-mean with the KNN classifier.

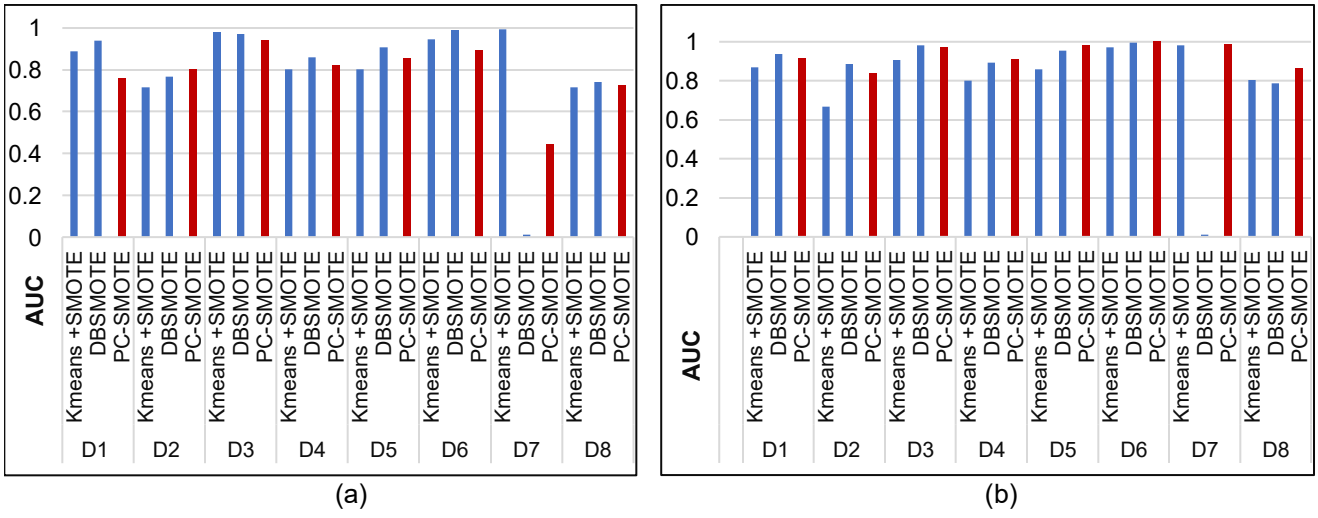


Fig. 9. Classification results on eight real-world datasets using different oversampling methods(a)AUC with SVM classifier(b) AUC with RF Classifier

Manuscript received December 8, 2024; Revised February 10, 2025; Accepted March 1, 2025; Available online June 5, 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.835

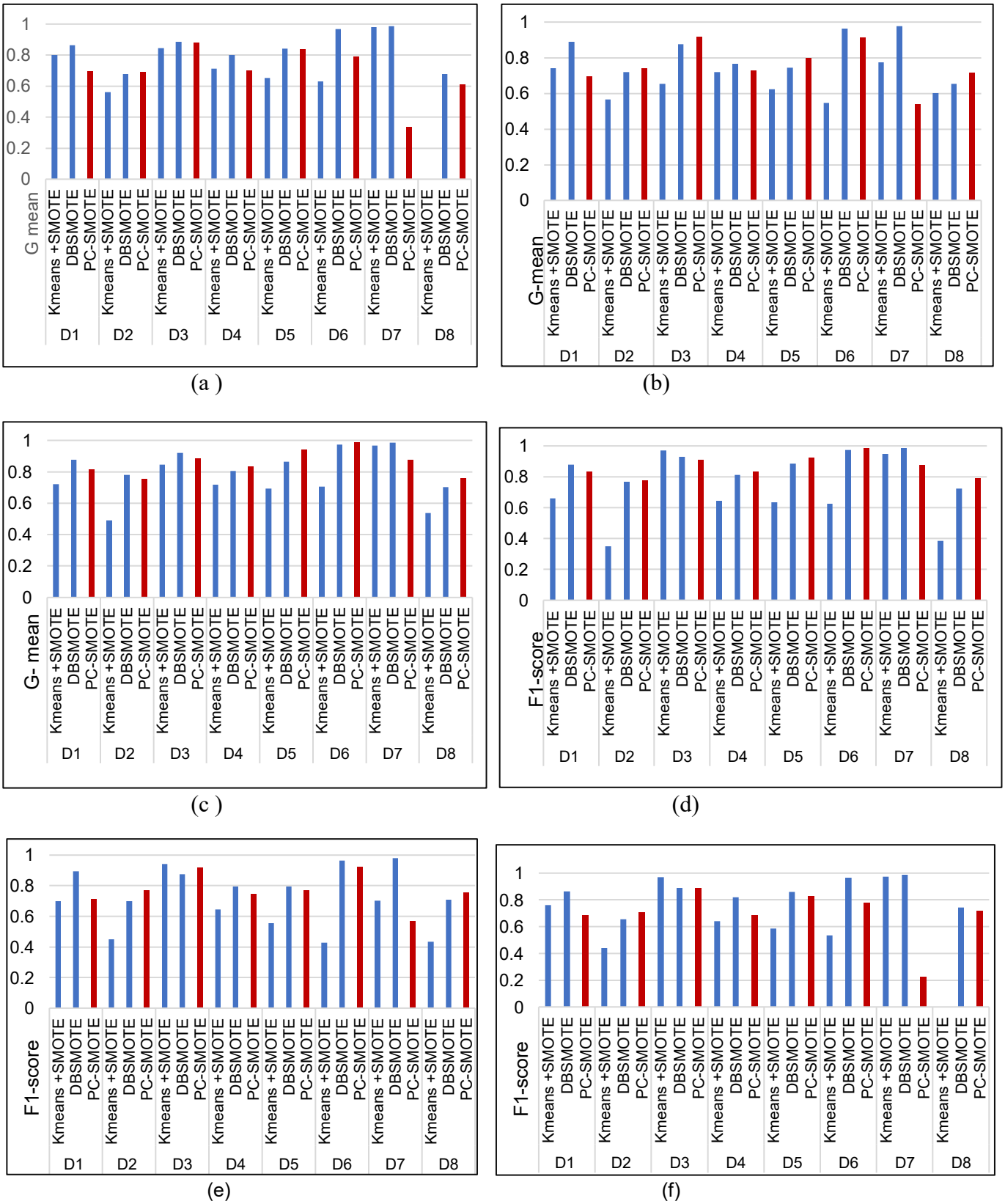


Fig. 10: Classification results on eight real-world datasets using different oversampling methods:(a) (f) SVM classifier;(b)(e) KNN classifier;(c) (d) RF classifier.

Manuscript received December 8, 2024; Revised February 10, 2025; Accepted March 1, 2025; Available online June 5, 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.835

SMOTE and significantly more efficient than K-means SMOTE, particularly in high-dimensional or large datasets.

VII. Discussion.

We performed t-test on 5-fold cross-validation to evalute reibaility and meaningfullness of the results. Performance was measured with F1-scores and G-means to give a complete understanding of the effectiveness of classification when faced with class imbalance. The paired t-test results affirm the consistency and statistical solidity of the PC-SMOTE algorithm over various datasets and classifiers. Significant p-values are observed in most cases, particularly with SVM and KNN, indicating consistent reliability and statistical robustness of the PC-SMOTE method across multiple datasets and classifiers and also indicating consistent performance improvement in F1 scores. While a few cases showed no significant difference or favored the baseline in terms of G-means, the overall findings support the stability of PC-SMOTE in handling imbalanced data. These results affirm that the observed improvements are not due to random variation but reflect a consistent enhancement in classification performance.

To compare performance, we look at Table 6, where a comparative evaluation of PC-SMOTE to other imbalance handling methods is given For the classifiers used in the study(RF,SVM and KNN) the proposed method PC-SMOTE shows a increase or boost in the value of F1-score to more than 119% for SVM and 47% for KNN. This establishes the PC-SMOTE as a superior approach for optimizing precision and recall in classification tasks. G-Mean sees gains, exceeding 122% with SVM and 46% with KNN, proving PC-SMOTE's ability to balance sensitivity and specificity under extreme imbalance. PR-AUC improvements surpass 43% (KNN), which is critical in domains where accurate positive class prediction is vital. Even in AUC metrics, PC-SMOTE outperforms others by up to 35%(RF), demonstrating that it maintains global model performance while addressing class bias. Further, the performance of our proposed PC-SMOTE method can be comprehensively compared with previous oversampling studies, particularly DB-SMOTE and K-means SMOTE, which have addressed similar challenges in imbalanced learning through clustering-based approaches. DB-SMOTE [22] contributes to optimize minority class oversampling through density-based clustering, achieving high F1-scores ranging from 0.812 to 0.962 across multiple datasets while effectively reducing noise generation in dense regions. DB-SMOTE's experimental results across varying imbalance ratios provide valuable comparison points. On the Pima dataset (largest minority class incidence rate ~25%), DB-SMOTE achieved an F-value of 0.877 and AUC of 0.888 using the k-NN classifier. The DBSCAN-based method [23] can effectively determine the best cluster boundaries for oversampling, enhancing classification performance, particularly in datasets with well-defined dense minority regions. However, DB-SMOTE [22] showed limitations in handling fragmented distributions, as evidenced by poor performance (AUC: 0.01) on certain datasets where density assumptions failed. K-means SMOTE algorithm[21] for minority class augmentation is used to achieve optimal classification accuracy by filtering out noisy and irrelevant regions through traditional clustering approaches. The method achieved average AUPRC improvements of 0.035 and demonstrated the ability to reduce false positives by 55% compared to traditional SMOTE, with some datasets achieving over 90% false positive reduction. The SMOTE algorithm, proposed by Chawla et al. (2002)[15], has consistently performed well in several applications, showing a 5-15% gain in F1-score and 8-18% in G-Mean. Borderline-SMOTE [16], shows a siginificant improvement with a raise of 10-20% in F1-score and 12-22% in G-Mean on UCI benchmark datasets. In contrast to these previous methods, our PC-SMOTE approach utilizes percolation theory for structural connectivity analysis, achieving superior F1-score improvements up to 119% and G-Mean improvements exceeding 122% across diverse datasets. PC-SMOTE collected connectivity patterns from minority class distributions and extracted structural coherence through percolation-based clustering, enabling the identification of connected components regardless of shape or density assumptions. The method performed has given excellent performance for all eight datasets (D1-D8),achieving good results with AUC values of up to 0.99 and F1-scores of 0.98, significantly outperforming density-based (DB-SMOTE) and traditional clustering methods (K-means SMOTE). The porposed method used with RF classifier gives the results for AUC with value of 0.90 and F1-score value of 0.83,which represents a significant improvement.

The limitation of PC-SMOTE, due to its inherent nature, may fail to isolate meaningful regions for oversampling in datasets where the class imbalance is not structurally separable, such as those with overlapping minority and majority instances. Furthermore, a critical point for consideration is that while PC-SMOTE helps in datasets with fragmented or irregular class distributions (where DB-SMOTE may underperform), it may slightly degrade performance in datasets where the original minority class is already well-structured (as observed, both D1 and D4 have an Imbalance ratio below 2), leading to marginally reduced gains when additional transformation is unnecessary. Additionally, it is worth noting that the intention of PC-SMOTE design is not to excel in every single case but to be a more stable and uniform improvement across

Manuscript received December 8, 2024; Revised February 10, 2025; Accepted March 1, 2025; Available online June 5, 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.835

various imbalance cases, particularly noisy, disconnected, or non-globular data areas, in which DB-SMOTE's use of local density estimation could malfunction. The combination of the percolation theory with the SMOTE algorithm in PC-SMOTE efficiently mitigates class imbalance in sparse and irregular data, which is a common phenomenon in medical data sets. Utilizing the structural knowledge of percolation-based clustering makes the integrity of synthetic sample generation better, allowing minority class regions to be modeled more accurately.

The better performance of PC-SMOTE has direct clinical implications for medical diagnosis and treatment planning. The better sensitivity and specificity balance imply that clinical decision support systems adopting PC-SMOTE may decrease both missed diagnoses (false negatives) and unnecessary interventions (false positives), which results in improved patient outcomes and more effective healthcare resource utilization PC-SMOTE's ability to handle sparse and fragmented minority class distributions makes it well-suited for healthcare data. It can improve predictive modeling for rare diseases, drug side effects, and other low-frequency, high-impact medical events.

VIII. Conclusion

The work aims to develop nature-inspired clustering that, combined with SMOTE, generates synthetic samples that adhere to the underlying data distribution and maintain sparsity among the data points. Incorporating percolation theory as a clustering approach has the potential to dynamically address the issues and yield high-quality clustering in complex medical datasets. PC-SMOTE method works excellently for the Breast cancer dataset, Parkinson's dataset, and Cervical cancer dataset, where AUC is in the range of 96% to 99%, which is high compared to the other two methods. This demonstrates the effectiveness of the PC-SMOTE algorithm in handling datasets with both low and high imbalance ratios and often demonstrate competitive or superior performance compared to K-means and DBSCAN combined with SMOTE in terms of AUC, F1-score, G-mean, and PR-AUC. The percolation threshold is a critical value that determines the connectivity of nodes. The effectiveness of Percolation clustering depends on the optimization threshold value. Future work would be to information. As hyperparameter tuning can further improve classification results, we recommend future efforts to apply techniques to tune the hyperparameter and evaluate its impact on classification, considering the proposed technique. In datasets where the class improves percolation clustering techniques; we suggest heuristic-based threshold value calculations that not only have better clustering but will include domain imbalance is not structurally separable, such as overlapping minority and majority instances, percolation clustering might fail to isolate meaningful regions for oversampling. We suggest future work should explore noise sample filtering mechanisms and hybrid clustering strategies to improve robustness and applicability. Further, we recommend that PC-SMOTE be tested across different kinds of medical data for enhanced diagnostic accuracy and patient stratification.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

RESEARCH FUNDING

This research received no external funding.

IX. References

[1] Salmi, M., Atif, D., Oliva, D., Abraham, A., & Ventura, S. (2024). Handling imbalanced medical datasets: review of a decade of research. *Artificial Intelligence Review*, *57*(10), 273. https://doi.org/10.1007/s10462-024-10884-2

[2] Khushi, M., Shaukat, K., Alam, T. M., Hameed, I. A., Uddin, S., Luo, S., Yang, X., & Reyes, M. C. (2021). A Comparative Performance Analysis of Data Resampling Methods on Imbalance Medical Data. *IEEE Access*, *9*, 109960–109975. https://doi.org/10.1109/ACCESS.2021.3102399

[3] Xu, Z., Shen, D., Nie, T., Kou, Y., Yin, N., & Han, X. (2021). A cluster-based oversampling algorithm combining SMOTE and k-means for imbalanced medical data. *Information Sciences*, *572*, 574–589. https://doi.org/10.1016/j.ins.2021.02.056

[4] Ali, Aida, Siti Mariyam Shamsuddin, and Anca L. Ralescu. "Classification with class imbalance problem." *Int. J. Advance Soft Compu. Appl* 5.3 (2013): 176-204.

[5] Dudjak, M., & Martinović, G. (2021). An empirical study of data intrinsic characteristics that make learning from imbalanced data difficult. *Expert Systems with Applications*, *182*, 115297. https://doi.org/10.1016/j.eswa.2021.115297

[6] Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study1. *Intelligent Data Analysis*, *6*(5), 429–449. https://doi.org/10.3233/IDA-2002-6504

[7] García, V., Mollineda, R. A., & Sánchez, J. S. (2008). On the k-NN performance in a challenging scenario of imbalance and overlapping. *Pattern Analysis and Applications*, *11*(3–4), 269–280. https://doi.org/10.1007/s10044-007-0087-5

Manuscript received December 8, 2024; Revised February 10, 2025; Accepted March 1, 2025; Available online June 5, 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.835

[8] López, V., Fernández, A., García, S., Palade, V., & Herrera, F. (2013). An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, *250*, 113–141. https://doi.org/10.1016/j.ins.2013.07.007

[9] Cao, C., & Wang, Z. (2018). IMCStacking: Cost-sensitive stacking learning with feature inverse mapping for imbalanced problems. *Knowledge-Based Systems*, *150*, 27–37. https://doi.org/10.1016/j.knosys.2018.02.031

[10] Roy, S., Roy, U., Sinha, D., & Pal, R. K. (2023). Imbalanced ensemble learning in determining Parkinson's disease using Keystroke dynamics. *Expert Systems with Applications*, *217*, 119522. https://doi.org/10.1016/j.eswa.2023.119522

[11] Khuat, T. T., & Le, M. H. (2020). Evaluation of Sampling-Based Ensembles of Classifiers on Imbalanced Data for Software Defect Prediction Problems. *SN Computer Science*, *1*(2), 108. https://doi.org/10.1007/s42979-020-0119-4

[12] Thabtah, F., Hammoud, S., Kamalov, F., & Gonsalves, A. (2020). Data imbalance in classification: Experimental evaluation. *Information Sciences*, *513*, 429–441. https://doi.org/10.1016/j.ins.2019.11.004

[13] Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, *73*, 220–239. https://doi.org/10.1016/j.eswa.2016.12.035

[14] Haibo He, Yang Bai, Garcia, E. A., & Shutao Li. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 1322–1328. https://doi.org/10.1109/IJCNN.2008.4633969

[15] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/jair.953

[16] Han, H., Wang, W.-Y., & Mao, B.-H. (2005). *Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning* (pp. 878–887). https://doi.org/10.1007/11538059_91

[17] Haibo He, Yang Bai, Garcia, E. A., & Shutao Li. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 1322–1328. https://doi.org/10.1109/IJCNN.2008.4633969

[18] Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2009). *Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem* (pp. 475–482). https://doi.org/10.1007/978-3-642-01307-2_43

[19] Dou, J., Gao, Z., Wei, G., Song, Y., & Li, M. (2023). Switching synthesizing-incorporated and cluster-based synthetic oversampling for imbalanced binary classification. *Engineering Applications of Artificial Intelligence*, *123*, 106193. https://doi.org/10.1016/j.engappai.2023.106193

[20] Chen, W., Guo, W., & Mao, W. (2024). An adaptive over-sampling method for imbalanced data based on simultaneous clustering and filtering noisy. *Applied Intelligence*, *54*(22), 11430–11449. https://doi.org/10.1007/s10489-024-05754-x

[21] Douzas, G., Bacao, F., & Last, F. (2018). Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. *Information Sciences*, *465*, 1–20. https://doi.org/10.1016/j.ins.2018.06.056

[22] Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2012). DBSMOTE: Density-Based Synthetic Minority Over-sampling TEchnique. *Applied Intelligence*, *36*(3), 664–684. https://doi.org/10.1007/s10489-011-0287-y

[23] Yang, Y., Akbarzadeh Khorshidi, H., & Aickelin, U. (2023). A Diversity-Based Synthetic Oversampling Using Clustering for Handling Extreme Imbalance. *SN Computer Science*, *4*(6), 848. https://doi.org/10.1007/s42979-023-02249-3

[24] Zhang, M., Ma, Y., Li, J., & Zhang, J. (2023). A density connection weight-based clustering approach for dataset with density-sparse region. *Expert Systems with Applications*, *230*, 120633. https://doi.org/10.1016/j.eswa.2023.120633

[25] Mahesh Kumar, K., & Rama Mohan Reddy, A. (2016). A fast DBSCAN clustering algorithm by accelerating neighbor searching using Groups method. *Pattern Recognition*, *58*, 39–48. https://doi.org/10.1016/j.patcog.2016.03.008

[26] Kaur, I., & Ahmad, T. (2024). A cluster-based ensemble approach for congenital heart disease prediction. *Computer Methods and Programs in Biomedicine*, *243*, 107922. https://doi.org/10.1016/j.cmpb.2023.107922

[27] Arafa, A., El-Fishawy, N., Badawy, M., & Radad, M. (2022). RN-SMOTE: Reduced Noise SMOTE based on DBSCAN for enhancing imbalanced

Manuscript received December 8, 2024; Revised February 10, 2025; Accepted March 1, 2025; Available online June 5, 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.835

data classification. *Journal of King Saud University - Computer and Information Sciences*, *34*(8), 5059–5074. https://doi.org/10.1016/j.jksuci.2022.06.005

[28] Tao, X., Li, Q., Guo, W., Ren, C., He, Q., Liu, R., & Zou, J. (2020). Adaptive weighted over-sampling for imbalanced datasets based on density peaks clustering with heuristic filtering. *Information Sciences*, *519*, 43–73. https://doi.org/10.1016/j.ins.2020.01.032

[29] Tong, W., Wang, Y., & Liu, D. (2023). An Adaptive Clustering Algorithm Based on Local-Density Peaks for Imbalanced Data Without Parameters. *IEEE Transactions on Knowledge and Data Engineering*, *35*(4), 3419–3432. https://doi.org/10.1109/TKDE.2021.3138962

[30] Huth, G., Lesne, A., Munoz, F., & Pitard, E. (2014). Correlated percolation models of structured habitat in ecology. *Physica A: Statistical Mechanics and Its Applications*, *416*, 290–308. https://doi.org/10.1016/j.physa.2014.08.006

[31] Ijaz, M., Alfian, G., Syafrudin, M., & Rhee, J. (2018). Hybrid Prediction Model for Type 2 Diabetes and Hypertension Using DBSCAN-Based Outlier Detection, Synthetic Minority Over Sampling Technique (SMOTE), and Random Forest. *Applied Sciences*, *8*(8), 1325. https://doi.org/10.3390/app8081325 .

[32] Hong, S., An, S., & Jeon, J.-J. (2024). *Improving SMOTE via Fusing Conditional VAE for Data-adaptive Noise Filtering*.

[33] Torquato, S. (2002). *Random Heterogeneous Materials* (Vol. 16). Springer New York. https://doi.org/10.1007/978-1-4757-6355-3

[34] Maddison, M. S., & Schmidt, S. C. (2020). Percolation Analysis – Archaeological Applications at Widely Different Spatial Scales. *Journal of Computer Applications in Archaeology*, *3*(1), 269–287. https://doi.org/10.5334/jcaa.54

[35] Amil, P., Almeira, N., & Masoller, C. (2019). Outlier Mining Methods Based on Graph Structure Analysis. *Frontiers in Physics*, *7*. https://doi.org/10.3389/fphy.2019.00194

[36] .Ma, L., & Fan, S. (2017). CURE-SMOTE algorithm and hybrid algorithm for feature selection and parameter optimization based on random forests. *BMC Bioinformatics*, *18*(1), 169. https://doi.org/10.1186/s12859-017-1578-z

[37] Stefanowski, J. (2013). *Overlapping, Rare Examples and Class Decomposition in Learning Classifiers from Imbalanced Data* (pp. 277–306). https://doi.org/10.1007/978-3-642-28699-5_11

[38] Fernández-Navarro, F., Hervás-Martínez, C., & Antonio Gutiérrez, P. (2011). A dynamic over-sampling procedure based on sensitivity for multi-class problems. *Pattern Recognition*, *44*(8), 1821–1833. https://doi.org/10.1016/j.patcog.2011.02.019

[39] Wang, J., & Awang, N. (2025). A Novel Synthetic Minority Oversampling Technique for Multiclass Imbalance Problems. *IEEE Access*, *13*, 6054–6066. https://doi.org/10.1109/ACCESS.2025.3526673

[40] Wang, J., & Awang, N. (2024). MKC-SMOTE: A Novel Synthetic Oversampling Method for Multi-Class Imbalanced Data Classification. *IEEE Access*, *12*, 196929–196938. https://doi.org/10.1109/ACCESS.2024.3521120

Biography of Authors

Kaikashan I Siddavatam is currently pursuing her Doctor of Philosophy (Ph.D.)at Lokmanya Tilak College of Engineering ,Navi Mumbai ,Maharashtra, where she is actively engaged in advanced research in the field of Computer Science and Engineering. She obtained her Master of Engineering (M.E.) degree in Electronics and Telecommunication engineering .She has developed a solid foundation in core subjects like Data Science, Machine learning, Deep Learning, which has equipped her with essential problem -solving skills. Her active participation in these organizations reflects his passion for knowledge sharing and academic excellence. Email Kaikashan.s@ltce.in

Dr. Subhash K. Shinde, an accomplished Computer Engineer, is the Principal and Professor in Computer Engineering at Lokmanya Tilak College of Engineering, Navi Mumbai. He has completed his M.E. in Information Technology (1999) and Ph.D. in Computer Science and Engineering (2012). He has more than 24 years' experience in the field of academics, administration and research. He has published about 55 papers in International Journals and Conferences and has copyright to his credit. He has also authored five books through reputable publishers. Under his supervision, 6 research scholars (PhD) and 30 PG (ME) Students have successfully completed their degrees from the University of Mumbai. He is working as a Chairman Board of Studies in Computer Engineering under the Faculty of Technology, University of Mumbai, and a Member of the Academic Council, BUTR, RRC, of the University of Mumbai from Nov 2023. Email: skshinde@rediffmail.com

Manuscript received December 8, 2024; Revised February 10, 2025; Accepted March 1, 2025; Available online June 5, 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i3.835