RBKWOT: Rough-Allowance Borderline K-Mode Clustering, Weighted Oversampling Technique for Class Imbalance Handling

Anima Pramanik and Sankar K. Pal, *Life Fellow, IEEE*

***Abstract*— The performance of machine learning (ML) algorithms is impacted due to the class imbalance issue. Although there are several techniques developed to take care of this issue, research on handling categorical or mixed imbalanced data has not been explored much. Moreover, the existing class imbalance handling techniques may not preserve the original characteristics of the data. To address these issues, a new technique, namely, the rough-allowance borderline K-mode clustering-based weighted oversampling technique (RBKWOT), is developed. It consists of four steps: 1) extraction of rough-allowance borderline minority samples; 2) generation of optimum K-mode clusters; 3) assignment of sampling weights to these clusters; and 4) generation of rules for resampling instances corresponding to each cluster based on its sampling weight. The RBKWOT is capable of reducing the imbalance ratio without incurring any loss of information. It also handles the uncertainty in decision-making for the samples belonging to the boundary region. The efficacy of RBKWOT over some state-of-the-art techniques is demonstrated using 21 categorical, continuous, and mixed data that are acquired from the UCI Machine Learning Repository. Experimental results reveal the superiority of RBKWOT.**

***Index Terms*— Borderline samples, class imbalance, K-mode clustering, oversampling, rough-allowance information.**

I. Introduction

COMPETING with the growing demand of the market, data are generated and stored in large amounts every day in every aspect of life. Nowadays, it is common practice to use machine learning (ML) algorithms to analyze a huge amount of data to make the system efficient. In the ML paradigm, classification of patterns is one of the primary tasks. However, if patterns are scarce or unusual, it becomes a typical imbalance problem. Data having a nonuniform distribution of samples across different classes are called imbalanced data. The imbalance issue severely affects the classification performances since the classification results are more inclined toward the majority class. This necessitates the development of a technique for identifying scarce patterns from data and handling the class imbalance issue.

Received 15 October 2024; revised 26 April 2025; accepted 20 September 2025. Date of publication 11 February 2026; date of current version 19 March 2026. This article was recommended by Associate Editor P. Chattopadhyay. *(Corresponding author: Anima Pramanik.)*

Anima Pramanik is with the Department of Information Management, Management Development Institute, Gurgaon 122007, India (e-mail: apramanik17@gmail.com; anima.pramanik@mdi.ac.in).

Sankar K. Pal is with the Center for Soft Computing Research, Indian Statistical Institute, Kolkata 700108, India (e-mail: sankar@isical.ac.in).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TSMC.2025.3627388.

Digital Object Identifier 10.1109/TSMC.2025.3627388

Class imbalance handling techniques are broadly classified into two categories, such as resampling [1] and ensemble learning [2]. Resampling techniques are easier to apply than ensemble techniques. In resampling techniques [1], the class distribution in the data is artificially balanced. Resampling techniques are broadly categorized into two classes, including under-sampling and oversampling. Under-sampling techniques help in removing major instances from the original data. Whereas, oversampling techniques generate synthetic minority samples. Though under-sampling techniques require less time complexity, they may cause useful information loss (IL), which can be handled using oversampling techniques with considerable time complexity. The computation process of oversampling techniques in a random manner is relatively simple, but it results in over-fitting for the minority class. Whereas, oversampling techniques followed in an informed manner are capable of generating samples to bring the balanced distribution of the classes with no over-fitting issue.

Synthetic minority oversampling technique (SMOTE) [3] is the first member of the oversampling family. Instead of doing simple replication, this technique generates samples by interpolating between an instance and its nearest neighbors corresponding to the minority class. Existing SMOTE-variants can be classified into two major categories: 1) hybridization of SMOTE using either sampling methods [1], noise filter methods [4], or feature selector methods [2] and 2) modification of SMOTE, including safe-level-SMOTE [2], borderline-SMOTE [5], weighted SMOTE [6], and SMOTEWEEN [7]. Although SMOTE variants perform well, SMOTE is restricted to continuous data [3]. For categorical data, initially, the data are converted from categorical to continuous form, and then SMOTE is applied over the continuous data for resampling. This increases the number of classes (in terms of continuous form) for each predictor attribute, thereby leading to a change in the original characteristics of the data. It creates ambiguity in the properties of the predictor attributes. These facts encourage us to develop a technique that enables us to suitably handle the oversampling for categorical, continuous, and mixed data.

The samples belonging to the boundary region of two decision classes are most sensitive in decision-making [8]. Characteristics of two samples corresponding to a decision class may vary at its boundary region. Moreover, two samples from two different classes may have similar characteristics in the boundary region. These create difficulty

in decision-making, thereby resulting in uncertainty in the system. Granular computing [9], which functions at the granular level for handling uncertainty, is effective for large data. Various techniques, such as fractional fuzzy grammar (FFG) [10], rough-fuzzy integration (RFI) [11], and neighborhood rough-fuzzy (NRF) model [12], are developed in handling such uncertainty arising from granularity. FFG deals with geometrical granularity in the image plane for its syntactic recognition. NRF deals with the formulation of class-dependent granules in a fuzzy environment based on the neighborhood rough set. Whereas, RFI is significant when the uncertainty arises from both overlapping and granular regions. Our study deals with the original data space, not at its granular level.

Based on the class imbalance information, data can be divided into three sets, such as safe, borderline, and noisy [2]. The borderline set of a decision class is also very sensitive in decision-making [5]. Samples belonging to either the borderline or its nearby are more responsible for misclassification than the ones far from the borderline. When the samples belonging to different decision classes hold the same predictor values, it causes uncertainty in decision-making. The SMOTE-variants-based resampling process is one of the reasons to make this happen. Rough set theory (RST) [8] is very efficient in handling the aforesaid uncertainties. In RST, the indiscernible reduct is obtained using the crisp knowledge of the information system pertaining to the decision classes. The concept of RST is used in this study to handle the uncertainty in decision-making for samples belonging to the boundary region. RS has two sets, including the lower approximation set (LAS) and the upper approximation set (UAS). LAS corresponding to a class represents the crisp information; whereas, its UAS represents the information of all possible samples that may belong to the same class. Samples that are present in UAS but not in LAS belong to the boundary region. If the borderline set is greater than the LAS, then the said uncertainty issue may not be solved using only the information of the LAS. A new set, namely, rough-allowance borderline set (RABS), is developed by introducing the concept of rough allowance within the borderline set.

During the oversampling process, the RABS is used for obtaining sampling weights to get the information about how many times resampling can be done for a particular sample. As samples present within one cluster hold the same property, instead of considering the entire data, the use of clusters can be effective in defining sampling weights. This increases the accuracy as well as speed. Due to the applicability to large-scale data and simplicity of implementation, K-mode clusters are formed over RABS. After clustering, the sampling weight (% oversampling) for each cluster is computed to obtain the number of resamples without altering the properties of predicted attributes/features belonging to the same cluster. All these constitute an oversampling technique, namely, rough-allowance borderline K-mode clustering-based weighted oversampling technique (RBKWOT). In RBKWOT, original classes for each predictor attribute are preserved throughout the resampling process. It can be used for categorical, continuous, and mixed imbalanced data.

The efficacy of RBKWOT for oversampling is demonstrated over 21 benchmark datasets collected from the UCI ML repository. Comparisons between SMOTE-variants and RBKWOT are done. Experimental results reveal that the RBKWOT-generated synthetic data is capable of producing better classification results when tested with various ML classifiers. The contributions of the study are as follows.

1) A new unsupervised oversampling technique, namely, RBKWOT, is developed for handling the imbalanced categorical, continuous, and mixed data using RABS extraction, K-mode clustering, sampling weight adaptation, and resampling.
2) A least doubtful region, called RABS, is introduced. It handles the uncertainty in decision-making for the samples which present in the boundary region.
3) A rule is defined for resampling of the instances corresponding to each cluster based on its weight-value.
4) Properties of predictor attributes are not changed throughout the resampling process for all kinds of data. Thus, it preserves the original characteristics of data.

The article proceeds as follows. Related works and preliminaries are presented in Sections II and III, respectively. The developed RBKWOT is illustrated in Section IV. Results and conclusions are presented in Sections V and VI, respectively.

II. Related Works

A. Class Imbalance Handling Techniques

A survey on various class imbalance techniques is presented in [13]. The simplest form of under-sampling is RUS [14], where samples belonging to the majority class are randomly removed until a desired class distribution is found. Cost-sensitive boosting is presented in [15] for handling the data having both imbalanced class information and unequal error values. A comparison between several bagging and boosting techniques for handling imbalanced and noisy binary-class data is presented in [16]. In [17], a rebalancing optimization is developed for training, followed by a label-distribution-aware margin loss-based reweighting for the text data. Unlike oversampling techniques, both under-sampling and cost-sensitive learning incur IL. Some key literature on oversampling is presented Section II-B.

B. Oversampling Techniques

It is well known that oversampling followed by an informed way is better than random sampling. As an informed oversampling technique, KSMOTE is developed in [18], using K-means clustering for obtaining the desired class region. However, optimum clusters may not be detected. An advanced version, named KWMOTE [19], has been developed. Here, the weight of each selected cluster is defined to get the information of % oversampling. But this technique is noisy as the predictor-class information is changed after the resampling process. Aiming to avoid noise generation, a hierarchical clustering-based technique, named CURE-SMOTE, is developed in [20]. While noise generation is avoided, possible imbalances within the minority class are ignored. A loss

rescaling-based approach is developed in [21] to assign different weights to all clusters formed over text/image data. Several fitness functions for genetic programming are developed in [22] for handling binary-class imbalanced data. Binarization with boosting is done in [23] for handling multiclass imbalanced data. These are restricted to continuous data and incur noise in case of categorical and mixed data. To address these issues, a new oversampling technique, namely, RBKWOT, is developed by restricting the region of sampling within the set of classes corresponding to each predictor attribute, as described in Section IV.

III. Preliminaries

A. RST

Clustering within the universe (X) leads to incompleteness of knowledge in X, which is formulated using RST. Let $S = \langle X, A, C\rangle$ be an information table. Here, A and C represent the set of predictor attributes and decision classes, respectively. Let B be an indiscernible set ($B \subseteq A$). A set G for a decision class ($\in C$) can be represented by the information contained in its LAS and UAS. The LAS of G is denoted by the set, $\underline{B}G = \{g \in X : [g]_B \in G\}$. Whereas, $\overline{B}G = \{g \in X : [g]_B \cap G \neq \phi\}$ represents the UAS of G. Here, $[g]_B (= \{(x, g) \in X | gBx\})$ denotes the equivalence class of the sample, g, and gBx indicates that g is related to x with an equivalence relation, say I_B.

B. SMOTE

Using SMOTE, new minority samples are added by extrapolating the existing minority samples. Let $X : (q, m)$ be the dataset for a decision class of having q and m number of samples and predictor attributes ($\{a_{i\in m}\} \in A$), respectively. The X is defined as

$$X = \begin{bmatrix} x_1 \\ x_2 \\ . \\ x_q \end{bmatrix} = \begin{bmatrix} a_1^{x_1} & a_2^{x_1} & \dots & a_i^{x_1} & \dots & a_m^{x_1} \\ a_1^{x_2} & a_2^{x_2} & \dots & a_i^{x_2} & \dots & a_m^{x_2} \\ . & . & \dots & . & \dots & . \\ a_1^{x_q} & a_2^{x_q} & \dots & a_i^{x_q} & \dots & a_m^{x_q} \end{bmatrix} \quad (1)$$

where $a_i^{x_q}$ defines the value of ith predictor attribute for the qth sample (x_q) present in X. Let $y (= \{a_1^y, a_2^y, \dots, a_i^y, \dots, a_m^y\})$ be the generated sample corresponding to a sample x along the direction of its jth nearest neighbor (x_j). The a_i^y is defined as

$$a_i^y = a_i^x + \text{rand}\,(0, 1) \times \left(a_i^{x_j} - a_i^x\right). \quad (2)$$

C. K-Mode Clustering

The K-mode clustering technique [18] is designed to cluster the categorical data. It comprises various steps, including initialization, allocation of samples into clusters, and updating mode. Initially, K (number of clusters) is initialized. Then, samples are allocated to the nearest clusters based on their dissimilarity measures, and the modes of the clusters are updated. Thereafter, dissimilarity values for all the samples belonging to the current modes are recomputed. For any sample, if it is found that its nearest mode belongs to any different clusters rather than its current one, the sample is reassigned to the new cluster.

IV. Proposed RBKWOT

The developed RBKWOT has four phases: 1) RABS extraction; 2) cluster formation; 3) sampling weight adaptation; and 4) resampling, as explained in Sections IV-A–IV-D, respectively.

A. RABS Extraction

RABS consists of two sets, including the LAS and allowance borderline set (ABS). The LAS is obtained by the positive region of the equivalence relation (I_B). Whereas, ABS is obtained with the borderline samples having maximum correlation with the samples belonging to the LAS. As borderline samples are more sensitive in decision-making, oversampling using the rough-allowance borderline minority samples is more effective than considering all samples. As already defined, an information table is $S = \langle X, A, C\rangle$, where X represents the universe (dataset), $A = \{a_{i\in m}\}$ denotes the set of m number of predictor attributes, and $C = \{c_{j\in n}\}$ implies the set of n number of decision classes present in X. Predictor attributes are the features that are used for prediction/classification tasks. The extraction of LAS is explained in step 1.

1) Step 1 (LAS Extraction): The LAS (say, $\underline{S_{c_j}}$) for each decision class is obtained form X using its equivalence relation, I_B ($B \subseteq A$), defined as

$$\underline{S_{c_{j\in n}}} = \left\{[x \in X]_B \mid [x]_B \subseteq c_j\right\} \quad (3)$$

where $\underline{S_{c_j}}$ defines the LAS corresponding to jth decision class. For a decision class, the corresponding LAS represents the crisp region. With full certainty, this region is used for the oversampling task. But if only this region is considered for the resampling task, a huge amount of IL may occur in case of data having scarce LAS. Therefore, it is required to extract the allowance borderline samples (ABS) that have a high correlation with the LAS and are used for resampling, as explained in step 2.

2) Step 2 (ABS Extraction): ABS refers to the allowance borderline samples that belong to the least doubtful region corresponding to a decision class. In an information table (S), samples corresponding to a decision class are categorized into three sets, named safe, borderline, and noisy. Let $\{x_i\}_{i\in q_k}$ be the set of q_k number of samples corresponding to a decision class, say c_k. Let $r (n + 1 \leq r < q)$ be the number of nearest neighbors corresponding to each sample, x_i, where q indicates the total number of samples present in S. Here we explain the computation of r nearest neighbors of a sample, x_i, having categorical or mixed attributes. The dissimilarity scores between x_i and all other samples (data points) are first computed using (6). The score determines how close a sample is to the x_i. Based on the dissimilarity scores, r number of closest samples of x_i are identified as its r nearest neighbors. Let r' ($0 \leq r' < r$) be the number of samples from other decision classes which may belong to these r number of neighbors. The sample x_i is considered to be noisy if $r' = r$ [5]. Whereas, the sample is considered to be in a borderline set if $r/2 \leq r' < r$, or in a safe set if $0 \leq r' < r/2$. All these sets can be extracted for continuous data only. Borderline samples and the ones near the borderline are more apt to be

misclassified than the ones far from the borderline, and thus more important for classification. Therefore, there is a need to define a method for extracting borderline samples from all kinds of data (e.g., continuous, categorical, and mixed). Let $S_{\text{bo}}^{c_k}$ be the borderline set for a decision class, c_k, as defined

$$S_{\text{bo}}^{c_k} = \cup_{i=1,2,..,q}\, x_i \in c_k,\ if\ \frac{r}{2} \leq \text{NC}\,(x_i) < r \tag{4}$$

where $\text{NC}(x_i)$ represents the number of neighbors corresponding to x_i, as defined

$$\text{NC}\,(x_i) = \sum_{i,j \in q_k}^{j \neq i} 1,\ \text{if DS}\left(x_i, x_j \in c_k\right) \in \text{DS}\,(x_i, x_l) \tag{5}$$

where x_l represents the farthest sample belonging to the r-neighbors corresponding to x_i and $\text{DS}(x_i, x_j)$ indicates the dissimilarity score between x_i and x_j, as defined

$$\text{DS}\left(x_i, x_j\right) = \sum_{t=1}^{m} \left.\begin{array}{r} 1,\ \text{if } \left(a_t^{x_i} \neq a_t^{x_j}\right) = \text{category} \\ 0,\ \text{if } \left(a_t^{x_i} = a_t^{x_j}\right) = \text{category} \\ |a_t^{x_i} - a_t^{x_j}|, \text{otherwise.} \end{array}\right\} \tag{6}$$

After the detection of the borderline set ($S_{\text{bo}}^{c_k}$), an allowance threshold is used over this set for the extraction of ABS corresponding to each decision class (c_k). The target concept in ABS is similar to the rough set. The similarity relation (B^*) is defined between the LAS and samples of ABS corresponding to a decision class. Similarity relation is defined only if the following condition is satisfied.

Lemma 1 (L1): Symmetry, i.e., for each sample $x \in S_{\text{bo}}^{c_{k \in n}}$ and $y(\neq x) \in S$, $xB^*y\ \forall y \in \underline{S_{c_{k \in n}}}$, where the similarity relation (B^*) for $S_{\text{bo}}^{c_{k \in n}}$ is defined as

$$I_{B^*} = \cup_{k \in n} I_{B_{c_k}^*} = \cup_{k \in n}^{i \in q_k} x_i \in S_{\text{bo}}^{c_k},\ if\ x_i B^* y\,(\neq x_i) \tag{7}$$

where $y = \cup_{j \in q} y_j$ if $d((\cup_{j \in q} NE(x_i, y_j))/q < \tau) > (n+1)$. Here, d is the top $(n+1)$ nearest neighbors of x_i present within its LAS, and q is the number of samples present in S. τ indicates the allowance value, q_k represents the number of samples present in c_k, and $NE(x_i, y_j)$ implies the nearness value between x_i and y_j, as defined

$$\text{NE}\left(x_i, y_j\right) = \frac{\left.\begin{array}{r} 1, \text{if } \left(a_t^{x_i} = a_t^{y_j}\right) = \text{category} \\ \sum_{t=1}^{m} 0, \text{if } \left(a_t^{x_i} \neq a_t^{y_j}\right) = \text{category} \\ |a_t^{x_i} - a_t^{y_j}|, \text{otherwise} \end{array}\right\}}{m}. \tag{8}$$

Here, the allowance value τ is defined as

$$\tau = \frac{\sum_{i \in q_k, j \in q}^{k \in n, i \neq j} NE\left(x_i, y_j \in c_k\right)}{2 \times q}. \tag{9}$$

The ABS ($S_{\text{RAB}}^{c_k}$) for a decision class (say, c_k) is defined as

$$S_{\text{ABS}}^{c_k} = \bigcup_{i \in q_k} x_i \in c_k | x_i \notin \underline{S_{c_k}}\ \&\ x_i \text{ satisfies } L1. \tag{10}$$

For a decision class, the least doubtful region, i.e., RABS, is formed using the LAS and ABS, as explained in step 3.

3) Step 3 (RABS Extraction): RABS is obtained by integrating LAS with ABS. For a decision class, say c_k, the RABS is defined as: $S_{\text{RABS}}^{c_k} = S_{c_k} + S_{\text{ABS}}^{c_k}$.

The RABS is used for clustering to obtain the regions having samples with the highest co-similarity. Due to the simplicity and high speed, K-mode clustering is used. How the optimum K mode is selected is explained in Section IV-B.

B. Selection of Optimum K-Mode

For obtaining the optimum K-mode, we have defined a dissimilarity cost function following the proposition and updated the cluster information.

Proposition 1: Optimum K-mode is found when the minimum dissimilarity cost (error) is obtained with updated K-mode clusters' center in a finite number of iterations.

Proof: Let K be the number of clusters formed over RABS after υth iteration. Let $\{\text{CL}_\upsilon(K)\} = \{\text{cl}_1, \text{cl}_2, \ldots, \text{cl}_\beta, \ldots, \text{cl}_K\}$ be the set of centers of K-mode clusters. After every iteration, K increases by a unit value to check whether the information of centers for K-mode clusters needs to be updated or not. Let $\{\text{cl}_\beta(x_1), \text{cl}_\beta(x_2), \ldots, \text{cl}_\beta(x_{q_\beta})\}$ be the set of q_β number of samples associated with the center (cl_β) of the βth cluster. The sample, $\text{cl}_\beta(x_{q_\beta})$ is represented as: $\text{cl}_\beta(x_{q_\beta}) = \{a_{1,\beta}^{x_{q_\beta}}, a_{2,\beta}^{x_{q_\beta}}, \ldots, a_{m,\beta}^{x_{q_\beta}}\}$, where $a_{m,\beta}^{x_{q_\beta}}$ defines the value of a class that occurs maximum times for the mth predictor attribute corresponding to the βth cluster. Then, $\{\text{CL}_\upsilon(K)\}$ is defined as

$$\{\text{CL}_\upsilon\,(K)\} = \cup_{\beta \in K} \left\{a_{1,\beta}^x, a_{2,\beta}^x, \ldots, a_{m,\beta}^x\right\}. \tag{11}$$

For a predictor-attribute (say, jth predictor), the values of all samples corresponding to a cl_β is represented as: $a_{j,\beta}^{x_1}$, $a_{j,\beta}^{x_2}$, $\ldots$, $a_{j,\beta}^{x_{q_\beta}}$. Let n_1 be the number of samples that are newly added to these clusters at the next iteration. Based on these new samples, the centers of all the generated clusters may be changed. Let $\{\text{CL}_{\upsilon'}(K+1)\} = \{\text{cl}_1, \ldots, \text{cl}_\beta, \ldots, \text{cl}_K, \text{cl}_{K+1}\}$ be the updated clusters' centers, as defined

$$\left\{\text{CL}_{\upsilon'}\,(K+1)\right\} = \cup_{\beta \in K+1} \left\{a_{1,\beta}^x, a_{2,\beta}^x, \ldots, a_{m,\beta}^x\right\} \tag{12}$$

where $a_{j,\beta}^x$ represents the class that occurred maximum times for the jth predictor attribute corresponding to the cl_β. Let $\{a_j^1, a_j^2, \ldots, a_j^\varpi, \ldots, a_j^\xi\}$ be the set of ξ number of classes for the jth predictor attribute. $a_{j,\beta}^x$ is defined as

$$a_{j,\beta}^x = a_j^\varpi | o\left(a_{j,\beta}^\varpi\right) = \vee\left(o\left(a_{j,\beta}^1\right), o\left(a_{j,\beta}^2\right), \ldots, o\left(a_{j,\beta}^\xi\right)\right) \tag{13}$$

where $o(a_{j,\beta}^\varpi)$ represents the number of occurrences of ϖth class for the jth predictor attribute corresponding to the samples present under cl_β and $\vee$ defines the maximum operation. The $o(a_{j,\beta}^\varpi)$ is defined as: $o(a_{j,\beta}^\varpi) = \sum_{i \in q_\beta} 1 \mid a_{j,\beta}^{x_i} == a_{j,\beta}^\varpi$. After adding new samples to these clusters, their centers may be changed. Therefore, dissimilarity cost is computed for each cluster to check whether its center will be changed or not. The dissimilarity cost between intraclusters present in $\text{CL}_\upsilon(K)$

and $\mathrm{CL}_{\upsilon'}(K+1)$ are denoted as $Cst_{\upsilon}(K)$ and $Cst_{\upsilon'}(K+1)$, respectively. $Cst_{\upsilon}(K)$ is defined as

$$\mathrm{Cst}_{\upsilon}(K) = \frac{\sum_{\beta \in K} \left(\frac{1}{m \times (q_\beta)^2} \sum_{i,j \in q_\beta}^{i \neq j} \mathrm{DS}\left(x_i^\beta, x_j^\beta\right) \right)}{K} \tag{14}$$

where DS() represents the dissimilarity cost between two samples, as defined in (6). The clusters' center will be updated if rule 1 is satisfied.

Rule 1: K-mode $\leftarrow K$, if $\mathrm{Cst}_{\upsilon}(K) < \mathrm{Cst}_{\upsilon'}(K+1)$; Otherwise K-mode $\leftarrow (K+1)$.

If K-mode $\leftarrow (\mathrm{K}+1)$ then clusters' center will be updated; otherwise, clusters' centers will not be changed. In υth iteration, if $\mathrm{Cst}_{\upsilon}(K) < \mathrm{Cst}_{\upsilon'}(K+1)$, then the iteration stops and the optimum K-mode is obtained as K. The optimum K-mode can be found when the minimum dissimilarity cost is obtained with updated K-mode clusters' center. After the formation of K-mode clusters over RABS, sampling weights are adopted for these clusters, as explained in Section IV-C.

C. Sampling Weight Adaptation

In this step, initially, imbalanced ratio-based filtering is done over the optimum K-mode clusters to select some genuine clusters that can be used for the oversampling task. Let IR_β be the imbalance ratio for βth cluster, as defined

$$\mathrm{IR}_\beta = \left(q_\beta^{\mathrm{mi}} + 1\right) \Big/ \left(q_\beta^{\mathrm{ma}} + 1\right) \tag{15}$$

where q_β^{mi} and q_β^{ma} represent the number of samples present in βth cluster corresponding to minority and majority classes, respectively. βth cluster having $\mathrm{IR}_\beta < 1$ is selected for oversampling. That means a higher proportion of the minority samples is required for the cluster to be selected. After the selection of clusters, weights are assigned to these clusters to obtain the information about the percentage of oversampling (samples to be generated) corresponding to each selected cluster. This weight depends on how dense a single cluster is as compared with how dense all selected clusters are on average. Let $\{S(1), S(2), \ldots, S(\gamma), \ldots, S(\upsilon)\}$ be the set of υ clusters selected from the K number of clusters for the oversampling process based on the imbalance ratio. The computation of weight for a cluster, say $S(\gamma)$, is defined as follows. Let $\{S(\gamma)\} : \{x_1^\gamma, x_2^\gamma, \ldots, x_{q_k}^\gamma\}$ be the set of q_k number of samples present in γth cluster corresponding to a minority class, say c_k. The distance between rough-allowance borderline minority samples corresponding to the γth cluster is defined as

$$\mathrm{Dis}^k(S(\gamma)) = \sum_{i,j \in q_k}^{i \neq j} \left(\mathrm{DS}\left(x_i^\gamma, x_j^\gamma\right) \mid \left(x_i^\gamma, x_j^\gamma\right) \in S_{\mathrm{ABS}}^{c_k} \right). \tag{16}$$

After computing the $\mathrm{Dis}^k(S(\gamma))$ for γth cluster, it is normalized using the average distance $(\mathrm{Avg}^k(S(\gamma)))$, as defined

$$\mathrm{Avg}^k(S(\gamma)) = \frac{\mathrm{Dis}^k(S(\gamma))}{(q_k - 1) + (q_k - 2) + \cdots + 1}. \tag{17}$$

After computing the average distance for γth cluster, its density $(\mathrm{Den}^k(S(\gamma)))$ is computed, as defined

$$\mathrm{Den}^k(S(\gamma)) = q_\gamma \Big/ \left(\mathrm{Avg}^k(S(\gamma))\right)^m. \tag{18}$$

After computing the density of γth cluster, its sparsity weight $(\mathrm{SW}^k(S(\gamma)))$ is measured, as defined

$$\mathrm{SW}^k(S(\gamma)) = 1/\mathrm{Den}^k(S(\gamma)). \tag{19}$$

Finally, sampling weight $(W(S(\gamma)))$ is defined as

$$W(S(\gamma)) = \frac{\mathrm{SW}^k(S(\gamma))}{\sum_{i \in K} \mathrm{SW}^k(S(i))}. \tag{20}$$

The aggregated value of all sampling weights corresponding to all selected clusters is one. For a cluster (say γth), percentage (%) of oversampling is measured using the function $\|W(S(\gamma)) \times 1/(\mathrm{IR}_\gamma)\|$. Here, % oversampling is not user-defined and can be varied according to the nature of the imbalance dataset. After obtaining the value of % oversampling, resampling is done to generate synthetic minority samples, as explained in Section IV-D.

D. Resampling

Resampling is done to generate the minority samples. Here, $S(\gamma) = \{x_i^\gamma\}_{i \in q_k}$ and $x_i^\gamma = \{a_{j,\gamma}^{x_i}\}_{j \in m}$. Let $y_i^\gamma = \{a_{j,\gamma}^{y_i}\}_{j \in m}$ be the synthetic sample generated corresponding to x_i^γ. In the resampling process, initially, $q_{rs}(n+1 < q_{rs} < q_k)$ number of samples (neighbors) are randomly selected from $S(\gamma)$ in a way such that x_i^γ does not belong to these selected samples. Then, distances between x_i^γ and all selected random samples are computed. The normalized weight $(Nr_j(x_i^\gamma))$ for the sample, x_i^γ corresponding to the jth predictor attribute is defined as

$$\mathrm{Nr}_j\left(x_i^\gamma\right) = \mathrm{rand}(0.5, 1) \times \frac{\sum_{l \in q_{\mathrm{rs}}}^{i \neq l} \mathrm{Sim}\left(a_{j,\gamma}^{x_i}, a_{j,\gamma}^{x_l}\right)}{q_{\mathrm{rs}}} \tag{21}$$

where rand(0.5, 1) defines the rand function to select number between 0.5 and 1 and $\mathrm{Sim}(a_{j,\gamma}^{x_i}, a_{j,\gamma}^{x_l})$ represents the similarity between the values of two samples, and x_i^γ and x_l^γ corresponding to the jth predictor attribute. $\mathrm{Sim}(a_{j,\gamma}^{x_i}, a_{j,\gamma}^{x_l})$ is defined as

$$\mathrm{Sim}\left(a_{j,\gamma}^{x_i}, a_{j,\gamma}^{x_l}\right) = \left.\begin{array}{r} 1, \text{ if } \left(a_{j,\gamma}^{x_i} = a_{j,\gamma}^{x_l}\right) = \text{category} \\ 0, \text{ if } \left(a_{j,\gamma}^{x_i} \neq a_{j,\gamma}^{x_l}\right) = \text{category} \\ |a_{j,\gamma}^{x_i} - a_{j,\gamma}^{x_l}|, \text{otherwise.} \end{array}\right\} \tag{22}$$

After obtaining the normalized weight of x_i^γ for jth predictor attribute, a rule is defined to generate a new value corresponding to the same attribute for x_i^γ. Let $a_{j,\gamma}^{y_i}$ be the generated value corresponding to $a_{j,\gamma}^{x_i}$. Let a_j^ϖ (ϖth class under jth predictor attribute) be the category (ca)/value (co) of $a_{j,\gamma}^{x_i}$. As already defined, ξ number of classes present under jth predictor attribute. $a_{j,\gamma}^{y_i}$ is defined as

$$a_{j,\gamma}^{y_i} = \left.\begin{array}{c} a_j^\varpi, \text{ if } \mathrm{Nr}_j\left(x_i^\gamma\right) \geq 0.5 \;\&\; a_{j,\gamma}^{x_i} = \mathrm{ca} \\ a_j^{q \in \xi}, \text{ elif } o\left(a_j^{q \neq \varpi}\right) \left(\in \left\{x_i^{\gamma,\mathrm{ne}}\right\}\right) = \mathrm{Max} \;\&\; a_{j,\gamma}^{x_i} = \mathrm{ca} \\ a_j^\varpi, \text{ elif } \mathrm{Nr}_j\left(x_i^\gamma\right) = 0 \;\&\; a_{j,\gamma}^{x_i} = \mathrm{co} \\ a_j^\varpi + \mathrm{Nr}_j\left(x_i^\gamma\right), \text{ elif } \mathrm{Nr}_j\left(x_i^\gamma\right) \neq 0 \;\&\; a_{j,\gamma}^{x_i} = \mathrm{co} \end{array}\right\} \tag{23}$$

where $\{x_i^{\gamma,\text{ne}}\}$ defines the set of neighbor samples of x_i belonging to the γth cluster, $o(a_j^q)$ represents the number of occurrence of the a_j^q class in $\{x_i^{\gamma,ne}\}$, and Max defines the maximum number. Here, $\{x_i^{\gamma,\text{ne}}\}$ is defined based on the dissimilarity score between two samples, as computed in (6). $\{x_i^{\gamma,\text{ne}}\}$ consists of $(n+1)$ number of neighbor samples that have the lowest dissimilarity scores with the x_i^{γ}. The values of all predictor attributes for y_i^{γ} are generated based on (23). Using the resampling process, the total number of generated samples (gen(c_k)) for ν number of selected clusters corresponding to a minority class, c_k, is defined as

$$\text{gen}(c_k) = \sum_{i \in \nu} \left(q_i \times W(S(i)) + \left(I(c_k) - S_{\text{ABS}}^{c_k}\right)\right) \quad (24)$$

where q_i represents the number of rough-allowance borderline samples for the ith cluster, $S(i)$ corresponds to the minority class (c_k), and $I(c_k)$ is the total number of input samples for the same minority class.

The pseudocode of RBKWOT is presented in Section S.I in the Supplementary Material. To explain the key phases of the developed RBKWOT, some figures are presented in Section S.II-A in the Supplementary Material. Experimental results are presented in Section V.

V. Experiments

In RBKWOT, two phases, such as RABS extraction and rule generation are mainly contribute to making the resampling task efficient for all types of data. Our study has four broad subobjectives in order to show: 1) the usefulness of RABS in restricting the region of generated samples; 2) the effectiveness of the developed rules for resampling in preserving the original characteristics of data, 3) the effectiveness of RBKWOT in generating synthetic data; and 4) to demonstrate the superiority of RBKWOT over some recent state-of-the-art techniques. The details of datasets, experimental setup, and experimental results are discussed in Sections V-A–V-C, respectively.

A. Datasets

The performance of the developed RBKWOT has been demonstrated extensively over twenty one benchmark datasets ("Balance scale (BS)," "Ballons," "Breast cancer (BC)," "Connec," "Monk," "Nursery," "Thiroid," "Tic-Toc (Tic)," "Tumar," "Voting Records (VR)," "Car evaluation (CE)," "Ecoli," "Glass," "Image," "Adult," "Abalone," "Annealing," "Seeds," "Default credit card (DCC)," "Mushroom," and "Online retail (ORe)") collected from the UCI Machine Learning Repository. These are either continuous, categorical, or mixed data with varying levels of sample sizes and number of decision classes. A summary of each of the data with: 1) types; 2) number of predictors; 3) sample size; and 4) number of decision classes is presented in Table I.

TABLE I
Summary of the Benchmark Datasets Used

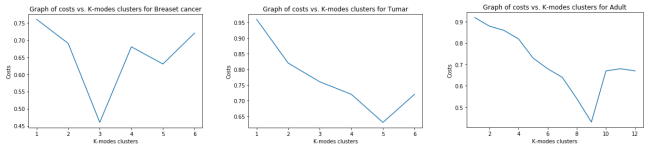
Datasets	Types	Size	Predictors	Decisions
Balance scale	Categorical	625	5	3
Ballons	Categorical	76	6	2
Breast cancer	Categorical	286	9	2
Monk	Categorical	554	5	2
Connec	Categorical	518	10	3
Thyroid	Categorical	334	9	3
Nursery	Categorical	12960	8	5
Tic-Toc	Categorical	958	9	2
Voting records	Categorical	435	16	2
Car evaluation	Categorical	1728	6	4
Tumar	Categorical	339	17	2
Ecoli	Continuous	336	9	8
Glass	Continuous	214	11	6
Image	Continuous	210	21	7
Seeds	Continuous	210	9	3
Abalone	Mixed	4177	8	3
Adult	Mixed	48842	14	2
Acute	Mixed	120	6	2
Default credit card	Continuous	30002	23	2
Mushroom	Mixed	61070	18	2
Online retail	Mixed	508771	7	8

B. Experimental Setup and Parameter Analysis

In the experimental setup, the developed RBKWOT algorithm is coded with Python 3.6 (Anaconda 2) and implemented in Windows 7 with an Intel i3 processor clocked at 3.30 GHz, and 8 GB of memory. The libraries used are "numpy," "pandas," "scipy," "math," and "random." The performance metrics used in the study are accuracy, recall, $F1$-score, G-mean, speed, AUROC, and Matthews correlation coefficient (MCC) [21]. Here, speed refers to the required runtime to execute an algorithm. AUROC defines the area under the receiver operating characteristic curve. MCC measures the correlation between predicted and actual values in a classification model. In RBKWOT, parameters are selected for the three phases, such as: 1) extraction of rough-allowance borderline minority samples; 2) formation of K-mode clusters; and 3) sampling weight (% of oversampling) adaptation, as explained in Section V-B.1–V-B.2, respectively.

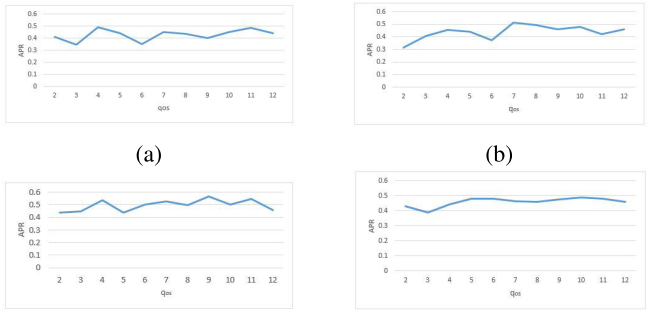
1) Parameter Selection in RABS Extraction: The RABS set is the combination of LAS and ABS. Here, ABS contains samples that belong to the borderline set and follow an allowance-threshold criterion. For the generation of borderline samples corresponding to any decision class, $r(n+1 \leq r < q)$ number of nearest neighbors are considered. As much as the nearest neighbor region is reduced, the probability of finding the boundary samples increases. Moreover, the nearest neighbor region must consider at least n number of samples, where n is the number of decision classes present in the data. To extract the most relevant borderline samples from the input data, $r(= n + 1)$ number of nearest neighbors are considered. Two parameters, namely, d and τ (allowance value), are used over the borderline set for extracting the ABS. For a sample, d is the top $(n+1)$ nearest neighbors selected from its LAS. Whereas, τ is based on the information of nearness value ($NE()$) and total number of samples (q) present in the original data. $NE()$ is computed using the class information of predictor attributes. Therefore, the computation of τ is purely based on the inherent information of data.

2) Parameter Selection in K-Mode Clustering: The parameter K is used to obtain the optimum number of clusters in K-mode clustering based on the dissimilarity cost. Graphs of dissimilarity costs versus K-mode clusters (which are formed over the RABS of the minority class) for categorical, continuous, and mixed data are presented in Fig. 1. Here,



(a) (b) (c)

Fig. 1. Dissimilarity costs versus K-mode clusters. (a) BC. (b) Tumar. (c) Adult.



(a) (b) (c) (d)

Fig. 2. APR versus q_{os} for "Breast cancer" data. (a) SVM. (b) KNN. (c) NB. (d) RF.

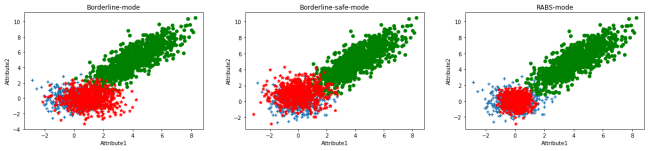
Fig. 1(a)–(c) corresponds to "Breast cancer," "Tumar," and "Adult" data, respectively. From Fig. 1(a), it is evident that, if three clusters are formed over the RABS of the minority class for "Breast cancer" data, then the dissimilarity cost is minimum. Therefore, for "Breast cancer" data, $K = 3$ is considered as the optimum number of clusters formed over the minor class samples. In the same way, $K = 5$ and $K = 9$ clusters are formed over the minority class samples for "Tumar" and "Adult" data, respectively.

3) Parameter Selection for Sampling Weight Adaptation: The % oversampling is based on q_{os}, which represents the number of samples to be generated. Optimum % oversampling is found at the criteria of maximum accuracy. For different values of q_{os}, different values of % oversampling are found for each selected cluster. For a cluster (say, βth cluster), q_{os} is varied from 2 to $1/(\text{IR}_\beta)$. Various ML-algorithms, such as SVM, KNN, RF, and NB, are used for obtaining the accuracy in terms of average positive rate (APR) for different values of q_{os}. The results for "Breast cancer" data are shown in Fig. 2. From Fig. 2(a), it is evident that for $q_{\text{os}} = 13$, SVM provides the best accuracy. Hence, for SVM classification over "Breast cancer" data, $q_{\text{os}} = 13$ is selected for obtaining the optimum % oversampling. From Fig. 2(b) to (d), it is found that for the same data, in case of KNN, NB, and RF classifiers, q_{os} is selected as 11, 13, and 13, respectively. The selection of % oversampling is based on the information of the data and the type of ML classifier used.

C. Experimental Results

Experiments along with comparisons are conducted, in line with the aforesaid four sub-objectives, as discussed in Sections V-C.1–V-C.4.

1) Effectiveness of RABS in Restricting the Region of Generated Samples: The effectiveness of RABS is assessed by analyzing the distribution of generated samples. To see the superiority of RABS, two other sets, namely, borderline and borderline+safe, are used in this experiment. The rule defined in (23) is used over borderline, borderline+safe, and RABS for minority sample generation. The results over "Ballons" data are shown in Fig. 3. Here, Fig. 3(a)–(c) represents the distribution of generated samples corresponding to borderline, borderline+safe, and RABS, respectively. In these figures, minor, major, and generated samples are represented by blue-colored "+," green-colored dot, and red-colored "*," respectively. If only the safe set of the minority class is used for sample generation, then there is a high possibility of generating duplicate samples. Whereas, the region of the borderline set is large, so synthetic samples can be generated without duplicity. Therefore, either the borderline or borderline + safe set is used for sample generation. As the borderline set of the minority class and the noisy set of the majority class overlap with each other, for some instances belonging to this overlapped region, the values of all predictor attributes may be the same. Therefore, the minority samples that are generated using the defined resampling rule [see (23)] may belong to the noisy set of the majority class. It is also found in Fig. 3(a) and (b), a small portion of the generated samples are present in the noisy set of the majority class. The developed RABS contains the minority samples that are far from the noisy set of the majority class. The minority samples that are generated from the RABS using the defined resampling rule [see (23)] may belong to the borderline set of the minority class [see Fig. 3(c)]. Hence, the generated samples are noise-free and can be used for the resampling task.



(a) (b) (c)

Fig. 3. RBKWOT results for different sets of Ballons data. (a) Borderline. (b) Borderline and safe. (c) RABS.

2) Effectiveness of the Developed Resampling Rule: In the RBKWOT, the developed resampling rule is used to preserve the original characteristics of the data. It is well known that the working principle of SMOTE-variants for oversampling is restricted only to continuous data. For the application of SMOTE-variants corresponding to categorical and mixed data, the same is initially converted into continuous form, and then oversampling is done for the generation of synthetic samples. In such cases, the number of values (in continuous form) for each predictor attribute increases. These newly generated continuous values of the synthetic samples for each predictor attribute cannot be retransformed back to their categorical form. That means the original characteristics of the categorical or mixed data get lost in the process of SMOTE-variants for oversampling. y_i^γ is the generated sample for x_i^γ. The IL in

the generated sample is defined as

$$\mathrm{IL}\left(y_i^{\gamma}\right) = \sum_{j \in m} \left(\mathrm{rand}\,(0,1) \times \mathrm{loss}\left(a_{j,\gamma}^{y_i}\right)\right) \tag{25}$$

where rand$(0, 1)$ represents the random value generated between 0 and 1. If $a_{j,\gamma}^{y_i} \in \{a_j^1, a_j^2, \ldots, a_j^{\xi}\}$, loss$(a_{j,\gamma}^{y_i}) = 0$; otherwise, loss$(a_{j,\gamma}^{y_i}) = 1$. Lowering the IL results in better information preservation and hence oversampling. In case of continuous, binary, ordinal, discrete, categorical, and mixed data, after applying SMOTE and its variants, the value of each predictor attribute corresponding to the synthetic sample is changed and may not belong to the set of classes present in each predictor attribute of the original data.

This issue does not exist in the RBKWOT. Here, as per the resampling rule defined in (23), for each newly generated synthetic sample, the value of each predictor attribute always belongs to the set of its original classes. Therefore, in RBKWOT, by using the defined resampling rule [see (23)], the IL for the newly generated sample = $\sum_{j \in m}(\mathrm{rand}(0, 1) \times (\mathrm{loss}_j = 0)) = 0$. This is the main advantage of using the resampling rule in RBKWOT to retain all the original information properly; hence, no IL is incurred. The RBKWOT enables preservation of the original characteristics of categorical, continuous, and mixed data during the resampling process. Therefore, RBKWOT can be applied to categorical, continuous, and mixed data for handling the class imbalance problem.

3) Significance of RBKWOT in Handling Class Imbalance Problem: The significance of any class imbalance handling technique over any data is proved using the performance of ML classifiers. The original characteristics of the categorical or mixed data get lost in the process of SMOTE-variants. RBKWOT works on the rough-allowance borderline samples. As borderline samples are more sensitive in decision-making, the concept of rough set is used to extract more samples closest to the borderline samples to get a more certain region for resampling tasks. During the resampling process, the number of classes for any predictor attribute is not changed. Hence, the characteristics of the data in terms of actual classes for all predictor attributes are preserved. This makes RBKWOT superior to SMOTE-variants.

For the quantitative assessment of RBKWOT, three performance metrics, such as Recall, $F1$-score, and G-mean, are used. Before and after applying RBKWOT over the data, these performance metrics are obtained to check the significance of RBKWOT for handling the class imbalance issue that occurs in continuous, categorical, and mixed data. ML-classifiers, such as SVM, KNN, NB, and RF, are used. The results of RBKWOT for one categorical data (BS), one continuous data (Ecoli), and one mixed data (Adult) are presented in Table II. In this table, for imbalanced data, "Rel," "$F1$," and "Gmn" represent the recall, $F1$-score, and G-mean, respectively. Whereas, for balanced data, "$Rel^{/}$," "$F1^{/}$," and "$Gmn^{/}$" define the recall, $F1$-score, and G-mean, respectively. From this table, it is seen that all the ML-classifiers provide better results on balanced categorical, continuous, and mixed data generated by the RBKWOT. Results for other categorical, continuous, and mixed data are presented in Section S.II-B in the Supplementary Material.

TABLE II

RESULTS FOR BOTH IMBALANCED AND RBKWOT-BASED BALANCED CATEGORICAL, CONTINUOUS, AND MIXED DATA

Data	Types	Classifiers	Rel	$Rel^{/}$	F1	$F1^{/}$	Gmn	$Gmn^{/}$
BS	Categorical	SVM	0.87	0.90	0.88	0.90	0.88	0.91
BS	Categorical	KNN	0.57	0.81	0.57	0.81	0.57	0.82
BS	Categorical	NB	0.66	0.71	0.63	0.72	0.63	0.72
BS	Categorical	RF	0.36	0.37	0.36	0.36	0.36	0.36
Ecoli	Continuous	SVM	0.61	0.92	0.62	0.93	0.62	0.93
Ecoli	Continuous	KNN	0.54	0.89	0.55	0.89	0.55	0.89
Ecoli	Continuous	NB	0.36	0.0.68	0.37	0.68	0.37	0.68
Ecoli	Continuous	RF	0.28	0.54	0.28	0.55	0.28	0.55
Adult	Mixed	SVM	0.56	0.79	0.57	0.8	0.57	0.90
Adult	Mixed	KNN	0.41	0.83	0.42	0.84	0.42	0.83
Adult	Mixed	NB	0.36	0.59	0.36	0.6	0.36	0.6
Adult	Mixed	RF	0.13	0.37	0.14	0.37	0.13	0.37

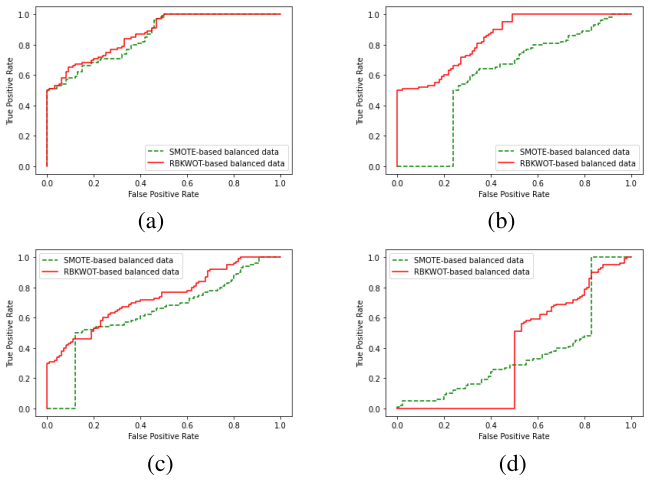


Fig. 4. AUROC for "Ballons" data. (a) SVM. (b) KNN. (c) NB. (d) RF.

For qualitative assessment of RBKWOT over SMOTE, the performance metric, AUROC, is used. It represents the probability that the model, if given a randomly chosen positive and negative example, will rank the positive higher than the negative. After the application of SMOTE and RBKWOT over the "Ballons" data, AUROC is computed for four ML-classifiers, including SVM, KNN, NB, and RF. The experimental results are presented in Fig. 4, where it is evident that, after applying SVM, KNN, NB, and RF over SMOTE-based balanced data, AUROC scores are 0.85, 0.59, 0.63, and 0.36, respectively. For RBKWOT-based balanced data, AUROC scores are 0.87, 0.84, 0.73, and 0.38 for SVM, KNN, NB, and RF, receptively. For all the classifiers, AUROC scores are higher in RBKWOT than the SMOTE. This proves the efficacy of RBKWOT over SMOTE.

4) Comparative Study: To assess the performance of RBKWOT for handling the class imbalance issue, four types of comparative studies based on: 1) visual performance; 2) classification performance; 3) CPU runtime; and 4) statistical analysis are done, as discussed in this section.

a) Visualization performance: The effectiveness of RBKWOT is assessed by analyzing the data visualization of various oversampling techniques. The difference between characteristics of RBKWOT and SMOTE variants is illustrated

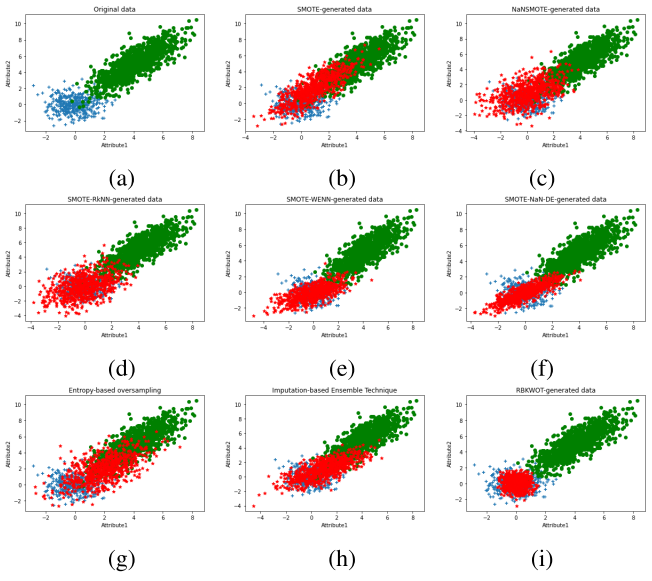


Fig. 5. Visualization of data generation. (a) Original. (b) SMOTE. (c) NaNSMOTE. (d) SMOTE-RkNN. (e) SMOTE-WENN. (f) SMOTE–NaN-DE. (g) EOT. (h) IET. (i) RBKWOT.

in Fig. 5. Here, Fig. 5(a) represents the distribution of an imbalanced data, "Ballons," with respect to two attributes. Here, blue-colored "+" represents samples of the minority class, whereas green-colored dot defines samples belonging to the majority class. The result of SMOTE [3] is presented in Fig. 5(b), where red-colored "*" represents the generated samples. From this figure, it is evident that a major portion of the generated minority samples persists near the center, noisy set, and borderline set of the majority class. The reason is: in the case of SMOTE, synthetic minority samples are generated randomly; therefore, instances may be generated in overlapping regions (of two/more classes), which are far from the safe set regions of other classes. This can lead to the generation of samples that may not hold the original characteristics of minority samples. It may negatively impact classification performance. The result of NaNSMOTE [24] is presented in Fig. 5(c). From this figure, it is evident that a small portion of the generated minority samples belong to the noisy set of the majority class. The reason is that in the case of NaNSMOTE, samples that belong to the safe set of the minority class have more neighbors for generating synthesized minority samples.

From Fig. 5(d), it is evident that, in SMOTE-RkNN [25], a small portion of the generated minority samples belong to the safe set and boundary of the borderline set of the majority class. The reason is that in the case of SMOTE-RkNN, k-nearest neighbors are considered for both safe and borderline minority samples. In SMOTE-WENN [7], the weight-edited nearest neighbors of the safe set and borderline set minority samples are considered for data generation. But the information on local imbalance and spatial sparsity is not considered here. Therefore, in SMOTE-WENN [refer to Fig. 5(e)], a small portion of generated minority samples belong to the safe set and boundary of the borderline set of the majority class. The oversampling technique, SMOTE-NaN-DE [4], can improve the SMOTE-based method by changing the position of error (noisy) samples using differential evaluation between natural neighbors corresponding to the center of borderline samples. Instead of the center of the safe set, the center of the borderline set is used; therefore, some noisy samples may not change their position at a safe distance. Therefore, in SMOTE-NaN-DE [refer to Fig. 5(f)], a small portion of the generated minority samples belong to the noisy set of the majority class. A non-SMOTE technique, namely, the entropy-based oversampling technique (EOT) [26], uses entropy-based imbalance degrees to measure the class imbalance instead of using the traditional imbalance ratio. In EOT, new samples are generated based on the characteristics of the majority class. Therefore, in EOT [refer to Fig. 5(g)], minor samples may be generated within the safe set and the borderline set of the majority class. Another non-SMOTE technique, namely, the imputation-based ensemble technique (IET) [27], uses training of a learning algorithm followed by the missing value imputation-based resampling. As missing value imputation is done for the resampling, it may generate noisy samples as shown in Fig. 5(h).

From the aforesaid results, it is evident that for both SMOTE- and non-SMOTE-variants, minority samples are generated within the boundary of the majority class. The result of the developed RBKWOT is presented in Fig. 5(i). From this figure, it is evident that all synthetic samples are generated within the boundary of the minority class. Unlike SMOTE, in the developed RBKWOT, synthetic samples are generated based on the minimum average distance between class values for all predictor attributes corresponding to the minority class. Unlike NaNSMOTE, in the developed RBKWOT, a new set, called RABS, is developed. The minority samples belonging to the RABS are used to generate instances without loss of originality. Unlike SMOTE-RkNN, in RBKWOT, local neighbor information for samples belonging to the safe set and RABS is considered. Unlike SMOTE-WENN, in RBKWOT, the local imbalance and spatial sparsity information corresponding to the minority class of RABS is considered for sampling weight adaptation. Unlike SMOTE-NaN-DE, in RBKWOT, both the centers of the safe set and RABS are used for the resampling.

For both SMOTE and non-SMOTE variants, corresponding to each predictor attribute, the number of classes is increased after the resampling. It creates ambiguity in decision-making. Therefore, these samples are called noisy. Whereas, in RBKWOT, the information of all classes for all predictor attributes corresponding to the minority class is considered for synthetic data generation. This holds the original characteristics of all predictor attributes present in the entire data. Thus uncertainty issue arising in predictor attributes during the data sampling is handled in RBKWOT.

b) Classification performance: An extensive comparative study is done over the twenty one benchmark data to show the efficacy of RBKWOT over some SMOTE variants (SMOTE (SM) [3], NaNSMOTE (NaNS) [24], K-means SMOTE (KS) [18], SMOTE-WENN (WENN) [7], and SMOTE-RkNN (RkNN) [25]), two non-SMOTE variants

TABLE III
RESULTS IN TERMS OF PERFORMANCE METRICS

Data	Metrics	SM	NanS	KS	EOT	WENN	RkNN	IET	LDAM	RBKWOT
BS	Recall	0.23	0.21	×	0.63	0.27	×	×	×	**0.90**
Ecoli	Recall	0.64	0.69	0.67	0.89	0.88	0.93	0.82	0.93	**0.94**
Adult	Recall	0.41	0.49	×	0.52	0.68	×	×	×	**0.77**
BS	F1-score	0.28	0.24	×	0.69	0.29	×	×	×	**0.90**
Ecoli	F1-score	0.69	073	0.77	0.91	0.92	0.94	0.89	0.93	**0.95**
Adult	F1-score	0.48	0.54	×	0.59	0.72	×	×	×	**0.80**
BS	G-mean	0.29	0.25	×	0.68	0.27	×	×	×	**0.91**
Ecoli	G-mean	0.69	073	0.77	0.91	0.92	0.93	0.89	0.93	**0.98**
Adult	G-mean	0.47	0.54	×	0.57	0.71	×	×	×	**0.90**
BS	MCC	0.4	0.52	×	0.21	0.46	×	×	×	**0.77**
Ecoli	MCC	0.45	0.51	0.59	0.58	0.65	0.77	0.69	0.71	**0.78**
Adult	MCC	0.02	0.09	×	0.08	0.14	×	×	×	**0.79**

(EOT [26] and IET [27]), and one deep learning-based technique (label-distribution-aware margin loss-based imbalance handling (LDAM) [17]). An SVM classifier is used. The comparative results between RBKWOT and the aforesaid state of the art for three datasets (BS, Ecoli, and Adult) are shown in Table III based on Recall, $F1$-score, G-mean, and MCC. Here, × represents "Not applicable." For each combination of data and classifier, the best results are marked by bold font. From these tables, it is found that for all data (categorical, continuous, and mixed), RBKWOT achieves higher Recall, $F1$-score, G-mean, and MCC as compared with the state of the art. It implies that in RBKWOT, the developed RABS represents boundary samples more precisely. The generated synthetic samples using RBKWOT are more accurate as compared with the state of the art. It proves the superiority of RBKWOT over some state of the art. Classification performance for other datasets is presented in Section S.II-C in the Supplementary Material.

c) CPU runtime: The computational complexity (CC) for the RBKWOT is $O(\text{RABS}) + O(\text{KM}) + O(\text{WA}) + O(\text{RSM})$, where $O(\text{RABS})$, $O(\text{KM})$, $O(\text{WA})$, and $O(\text{RSM})$ define the CCs for RABS extraction, K-mode clustering, sampling weight adaptation, and resampling, respectively. The $O(\text{RABS}) = O(\psi) + \psi(O(((q-1)^2+2) + (\psi \times q^2 - q) + 1))$, where $O(\psi)$ defines the CC for obtaining the LAS for $\psi(=(n-1))$ number of minor decision classes present in the data. $\psi(O((q-1)^2+2))$ and $\psi(O(\psi \times q^2 - q))$ represent CCs for obtaining borderline minority samples and ABS, respectively. The $O(\text{KM}) = \upsilon\psi(q_s(O(m(q_s-1)+K)) + m\psi(O(q_\zeta+1)) + O(Kq_k^2))$, where $q_s(O(m \times q_s - m + K))$ represents the CC for obtaining K clusters over $q_s(\in q_k)$ number of samples selected under a decision class (say, kth class). The CC for updating K-mode is $(m \times \psi(O(q_\zeta+1)))$, where $q_\zeta(\in q_\psi)$ is the number of samples selected in ζth cluster belonging to the kth decision class. Finally, $O(\text{Kq}_k^2)$ is the CC for obtaining the optimal value of K. The $O(\text{WA}) = \psi(O(1+\upsilon\times(q_\gamma^2+4)))$, where $\psi(O(1))$ represents the CC for obtaining υ number of filtered clusters from K-mode clusters corresponding to ψ number of decision classes. $\psi(\upsilon\times(q_\gamma^2+4))$ is the CC for assigning sampling weights (% oversampling) to all the filtered clusters corresponding to the ψ number of decision classes. Finally, $O(\text{RSM}) = \psi(O(\upsilon \times \text{pw}_\gamma \times q_\gamma))$ is the CC for generating synthetic samples. Here, pw_γ represents the % oversampling for the $\gamma(\in \upsilon)$th filtered cluster for a minor decision class.

TABLE IV
STATISTICAL SIGNIFICANCE TEST RESULTS

Algorithm	SMOTE	NaNS	KS	EOT	WENN	RkNN	IET	LDAM
RBKWOT	0.005	0.012	0.008	0.036	0.041	0.022	0.038	0.0016

For the BS dataset, the CPU runtimes for SMOTE, NaNS, KS, EOT, WENN, RkNN, IET, LDAM, and RBKWOT are 3, 4.5, 7, 10, 12, 9, 11, 12, and 7 s, respectively. The speed of RBKWOT is less than SMOTE variants, except for SMOTE and NaNS. Various tasks, such as RABS extraction, K-mode clustering, and sampling weight adaptation, are not performed in any member of the SMOTE family. But using RBKWOT, the generated synthetic samples are more accurate compared with the SMOTE variants.

The CC of an oversampling technique depends on the number of samples, predictor attributes, and decision classes. The CC will increase with the increase of any of these factors. In RBKWOT, for each decision class, the size of the RABS is less than its entire sample size. Unlike traditional SMOTE-variants, we only consider RABS (instead of entire samples) for the sampling operation corresponding to each decision class. K-mode clusters are formed over the RABS and are used in defining sampling weights (% of oversampling). In RBKWOT, not all K clusters (formed over the RABS) are considered for the sampling operation. The clusters with an imbalance ratio less than one are considered for the sampling operation, thereby resulting in increased classification accuracy and speed. These two approximations are considered to reduce the CC of RBKWOT for its application, even in the case of large-scale data.

d) Statistical analysis: A nonparametric statistical test is conducted to ensure the statistical significance of RBKWOT over some state of the art (SMOTE, NaNS, KS, EOT, WENN, RkNN, IET, and LDAM). The Wilcoxon signed-rank test is conducted with $\alpha = 0.05$. The null hypothesis H_0 is set as "RBKWOT is statistically significant than state of the art." If the p-value < 0.05, then H_0 will fail to reject; otherwise, H_0 will be rejected. Based on the accuracy, the Wilcoxon signed-rank test is done for each pairwise technique. Results are shown in Table IV. It is evident from this table that RBKWOT is statistically significant than all the aforesaid state of the art (as $p < 0.05$). Unlike RBKWOT, for the aforesaid state-of-the-art oversampling techniques, original characteristics of data are lost, which directly affects the performance of ML-classifiers.

VI. CONCLUSION

A new oversampling technique, namely, RBKWOT, is developed to handle the class imbalance issue arising in categorical, continuous, and mixed data. It is an advanced version of the SMOTE family. RBKWOT comprises four steps, including RABS extraction, K-mode clustering, sampling weight adaptation, and resampling. RABS is a newly designed set that represents the least doubtful region corresponding to rough boundary and borderline samples. The defined cost function can automatically determine the optimum K-value

in K-mode clustering over RABS. The proposed resampling rule ensures preservation of the original characteristics of the data. Unlike SMOTE-variants, RBKWOT can handle the uncertainty in decision-making for samples that are present in the overlapping regions, and reduce the imbalance ratio without IL. The proposed resampling rule in RBKWOT is equally applicable to categorical, continuous, and mixed data. Note that the functioning of RBKWOT is restricted only to the samples in the set RABS, instead of all the samples corresponding to a minority class. This leads to a significant improvement in the performance in terms of both accuracy and speed. Experimental results reveal that RBKWOT is superior to some SMOTE-variants, non-SMOTE-variants, and deep learning techniques when tested over 21 benchmark data (continuous, categorical, and mixed) acquired from the UCI machine learning repository.

ACKNOWLEDGMENT

Prof. Sankar K. Pal acknowledges the "National Science Chair" Award of the SERB-DST, Government of India.

REFERENCES

[1] Y. Xie, M. Qiu, H. Zhang, L. Peng, and Z. Chen, "Gaussian distribution based oversampling for imbalanced data classification," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 2, pp. 667–679, Feb. 2022.

[2] J. Sun, H. Li, H. Fujita, B. Fu, and W. Ai, "Class-imbalanced dynamic financial distress prediction based on AdaBoost-SVM ensemble combined with SMOTE and time weighting," *Inf. Fusion*, vol. 54, pp. 128–144, Feb. 2020.

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[4] J. Li et al., "SMOTE-NaN-DE: Addressing the noisy and borderline examples problem in imbalanced classification by natural neighbors and differential evolution," *Knowledge-Based Syst.*, vol. 223, Jul. 2021, Art. no. 107056.

[5] H. Han, W. Y. Wang, and B. H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," in *Proc. Int. Conf. Intell. Comput.*, Aug. 2005, pp. 878–887.

[6] M. R. Prusty, T. Jayanthi, and K. Velusamy, "Weighted-SMOTE: A modification to SMOTE for event classification in sodium cooled fast reactors," *Prog. Nucl. Energy*, vol. 100, pp. 355–364, Sep. 2017.

[7] H. Guan, Y. Zhang, M. Xian, H. D. Cheng, and X. Tang, "SMOTE-WENN: Solving class imbalance and small sample problems by oversampling and distance scaling," *Int. J. Speech Technol.*, vol. 51, no. 3, pp. 1394–1409, Mar. 2021.

[8] A. Pramanik, S. Sarkar, J. Maiti, and P. Mitra, "RT-GSOM: Rough tolerance growing self-organizing map," *Inf. Sci.*, vol. 566, pp. 19–37, Aug. 2021.

[9] S. K. Pal, B. Uma Shankar, and P. Mitra, "Granular computing, rough entropy and object extraction," *Pattern Recognit. Lett.*, vol. 26, no. 16, pp. 2509–2517, Dec. 2005.

[10] A. Pathak and S. Pal, "Fuzzy grammars in syntactic recognition of skeletal maturity from X-rays," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-16, no. 5, pp. 657–667, Sep. 1986.

[11] S. K. Pal, "Soft data mining, computational theory of perceptions, and rough-fuzzy approach," *Inf. Sci.*, vol. 163, nos. 1–3, pp. 5–12, Jun. 2004.

[12] S. K. Pal, S. K. Meher, and S. Dutta, "Class-dependent rough-fuzzy granular space, dispersion index and classification," *Pattern Recognit.*, vol. 45, no. 7, pp. 2690–2707, Jul. 2012.

[13] N. A. Azhar, M. S. Mohd Pozi, A. Mohamed Din, and A. Jatowt, "An investigation of SMOTE based methods for imbalanced datasets with data complexity analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 7, pp. 6651–6672, Jul. 2022.

[14] A. Elhassan, M. Aljourf, A. F. Mohanna, and M. Shoukri, "Classification of imbalance data using tomek link (T-link) combined with random under-sampling (RUS) as a data reduction method," *Global J. Technol. Optim.*, vol. 1, no. S1, pp. 1–11, 2016.

[15] Y. Sun, M. S. Kamel, A. K. C. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern Recognit.*, vol. 40, no. 12, pp. 3358–3378, Dec. 2007.

[16] T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Comparing boosting and bagging techniques with noisy and imbalanced data," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 41, no. 3, pp. 552–568, May 2011.

[17] K. Cao, C. Wei, A. Gaidon, N. Aréchiga, and T. Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1–12.

[18] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE," *Inf. Sci.*, vol. 465, pp. 1–20, Oct. 2018.

[19] Z. Xu, D. Shen, T. Nie, Y. Kou, N. Yin, and X. Han, "A cluster-based oversampling algorithm combining SMOTE and k-means for imbalanced medical data," *Inf. Sci.*, vol. 572, pp. 574–589, Sep. 2021.

[20] L. Ma and S. Fan, "CURE-SMOTE algorithm and hybrid algorithm for feature selection and parameter optimization based on random forests," *BMC Bioinf.*, vol. 18, no. 1, pp. 1–18, Dec. 2017.

[21] Y. Guo, L. Nie, Z. Cheng, Q. Tian, and M. Zhang, "Loss re-scaling VQA: Revisiting the language prior problem from a class-imbalance view," *IEEE Trans. Image Process.*, vol. 31, pp. 227–238, 2022.

[22] U. Bhowan, M. Johnston, and M. Zhang, "Developing new fitness functions in genetic programming for classification with unbalanced data," *IEEE Trans. Syst., Man, Cybern., B (Cybernetics)*, vol. 42, no. 2, pp. 406–421, Apr. 2012.

[23] A. Sen, M. M. Islam, K. Murase, and X. Yao, "Binarization with boosting and oversampling for multiclass classification," *IEEE Trans. Cybern.*, vol. 46, no. 5, pp. 1078–1091, May 2016.

[24] J. Li, Q. Zhu, Q. Wu, and Z. Fan, "A novel oversampling technique for class-imbalanced learning based on SMOTE and natural neighbors," *Inf. Sci.*, vol. 565, pp. 438–455, Jul. 2021.

[25] A. Zhang, H. Yu, Z. Huan, X. Yang, S. Zheng, and S. Gao, "SMOTE-RkNN: A hybrid re-sampling method based on SMOTE and reverse k-nearest neighbors," *Inf. Sci.*, vol. 595, pp. 70–88, May 2022.

[26] L. Li, H. He, and J. Li, "Entropy-based sampling approaches for multi-class imbalanced problems," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 11, pp. 2159–2170, Nov. 2020.

[27] R. Razavi-Far, M. Farajzadeh-Zanajni, B. Wang, M. Saif, and S. Chakrabarti, "Imputation-based ensemble techniques for class imbalance learning," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 5, pp. 1988–2001, May 2021.