Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.057538

ARTICLE

RE-SMOTE: A Novel Imbalanced Sampling Method Based on SMOTE with Radius Estimation

Dazhi E[1], Jiale Liu[2], Ming Zhang[1,*], Huiyuan Jiang[2] and Keming Mao[2]

[1]Shenyang Fire Science and Technology Research Institute, Ministry of Emergency Management of the People's Republic of China, Shenyang, 110034, China

[2]College of Software, Northeastern University, Shenyang, 110006, China

*Corresponding Author: Ming Zhang. Email: 18842534541@163.com

Received: 20 August 2024 Accepted: 12 October 2024 Published: 19 December 2024

ABSTRACT

Imbalance is a distinctive feature of many datasets, and how to make the dataset balanced become a hot topic in the machine learning field. The Synthetic Minority Oversampling Technique (SMOTE) is the classical method to solve this problem. Although much research has been conducted on SMOTE, there is still the problem of synthetic sample singularity. To solve the issues of class imbalance and diversity of generated samples, this paper proposes a hybrid resampling method for binary imbalanced data sets, RE-SMOTE, which is designed based on the improvements of two oversampling methods parameter-free SMOTE (PF-SMOTE) and SMOTE-Weighted Ensemble Nearest Neighbor (SMOTE-WENN). Initially, minority class samples are divided into safe and boundary minority categories. Boundary minority samples are regenerated through linear interpolation with the nearest majority class samples. In contrast, safe minority samples are randomly generated within a circular range centered on the initial safe minority samples with a radius determined by the distance to the nearest majority class samples. Furthermore, we use Weighted Edited Nearest Neighbor (WENN) and relative density methods to clean the generated samples and remove the low-quality samples. Relative density is calculated based on the ratio of majority to minority samples among the reverse k-nearest neighbor samples. To verify the effectiveness and robustness of the proposed model, we conducted a comprehensive experimental study on 40 datasets selected from real applications. The experimental results show the superiority of radius estimation-SMOTE (RE-SMOTE) over other state-of-the-art methods. Code is available at: https://github.com/blue9792/RE-SMOTE (accessed on 30 September 2024).

KEYWORDS

Imbalanced data sampling; SMOTE; radius estimation

1 Introduction

In recent years, machine learning techniques have played critical roles in the explosive data generated in various fields [1,2]. However, the imbalanced data distribution poses a significant challenge to traditional machine-learning techniques. Specifically, unbalanced datasets suffer from a skewed distribution of categories, with some classes significantly exceeding others. Various real-world applications encounter this issue, including fault diagnosis [3], fraud detection, bioinformatics, soil classification, and credit risk assessment [4].

The primary problem for unbalanced datasets is that it makes the model training unusable, which works well on balanced datasets by calibrating the loss function for optimal accuracy. For example, when the ratio of majority class to minority class is 98:2, the accuracy can still reach 98% even if all the samples are classified as majority class. On the other hand, all minority samples are ignored and misclassified, which makes it problematic in real-life applications. Accurate identification of cancer patients is of greater importance than that of non-cancer patients [5]. This study investigates binary imbalanced datasets where class relationships are clearly defined: one class represents the majority, while the other represents the minority.

The fundamental approach to addressing the binary class imbalance issue is to mitigate the bias toward the majority class and enhance the focus on the minority class, thereby achieving balanced performance across both classes [6]. It can be categorized into 3 types, data-level methods, algorithm-level methods, and cost-sensitive methods. Data-level methods balance the number of samples between majority and minority-based sampling, i.e., oversampling [7], cut sampling [8], and mixed [9–11]. Algorithm-level methods try to modify the classification model to improve the performance, such as changing the decision threshold for each class and training the classifier separately [12–14]. Cost-sensitive methods can be seen as a hybrid of data-level and algorithm-level. It incorporates misclassification costs or samples into the optimization process [15–18]. Among these, data-level methods are the most widely used compared to algorithm-level methods that rely on specific classifiers, or problem-specific cost-sensitive methods.

Chawla et al. [7] proposed the synthetic minority oversampling technique (SMOTE), which balanced the class distribution by adding synthetic minority samples. It reduced the possibility of overfitting and improved the generalization performance of the classifier on the test set. Unlike random oversampling with repeated samples, SMOTE generates synthetic samples by using the k-nearest neighbors of the considered minority class samples. In the last decade, various approaches have been studied to improve SMOTE at different levels, including (1) Improvements in the initial selection of samples, (2) Combination with undersampling, (3) Improvements in interpolation type, (4) Combination with feature selection or dimensionality reduction, (5) Adaptive sample generation, and (6) Filtering out noisy samples. Most of these SMOTE-based methods only focus on synthesizing a safe minority of samples and ignore other minority classes. It cannot overcome the data distribution of unbalanced datasets, which is prone to the distribution marginalization problem. Since the distribution of minority class samples dictates their available nearest neighbors, if a minority class sample lies at the boundary of the distribution, the interpolated samples generated from this sample and its neighbors will also be positioned near the edge, further marginalizing them. This results in a blurring of the boundary between majority and minority class samples, leading to more significant boundary ambiguity. While this process may balance the dataset, it also increases the complexity of the classification algorithm.

To solve the above problems, we propose radius estimation-SMOTE (RE-SMOTE) which is essentially an improved model based on parameter-free SMOTE (PF-SMOTE) [19] and SMOTE-Weighted Ensemble Nearest Neighbor (SMOTE-WENN) [20]. Specifically, the minority class is first divided into boundary minority and safe minority, as used in PF. For safe minority synthesis, the synthesized samples are interpolated into the region dominated by the minority class, while a Gaussian process is adopted to expand the boundaries of the minority class. Hence, boundary minority and safe minority samples are all reorganized. Then, data cleaning is performed based on WENN and relative density estimation. Different distance weights are applied to the majority and minority class samples by considering local imbalance and spatial sparsity. Relative density determines whether a sample is noisy by calculating a ratio between the number of majority samples and minority samples among reverse k-nearest neighbor samples. An extensive experimental study is conducted to evaluate the effectiveness of the RE-SMOTE method. In this study, 40 datasets are selected from the KEEL dataset repository. Commonly used evaluation metrics, such as the area under the curve (AUC) [21], F1 score, and the Wilcoxon signed-rank test [22], are employed for performance assessment.

In summary, the contributions of this paper are as follows:

(1) We propose RE-SMOTE, an advanced model that builds upon the foundational principles of PF-SMOTE and SMOTE-WENN. This hybrid approach leverages the strengths of both methods to more effectively tackle the problem of class imbalance.

(2) We classify minority class samples into boundary minority and safe minority categories. For the synthesis of safe minority samples, a Gaussian process is utilized to strategically expand the class boundaries, ensuring that these samples are interpolated within regions predominantly occupied by the minority class. This approach enhances the diversity and representativeness of the synthesized samples.

(3) The model incorporates advanced data-cleaning mechanisms using the WENN method and relative density estimation. By applying varying distance weights based on local class imbalance and spatial sparsity, this approach accurately identifies and eliminates noisy samples. The relative density is computed as the ratio of majority to minority class samples within the reverse k-nearest neighbors, ensuring a precise cleaning process that enhances the overall quality of the dataset.

(4) To demonstrate the robustness and effectiveness of the proposed RE-SMOTE method, comprehensive experiments are evaluated on 40 imbalanced data sets.

The rest of this paper is structured as follows: The related works are provided in Section 2. Section 3 describes RE-SMOTE in detail. Section 4 sets up the experiment. The experimental results and discussions are analyzed in Section 5. Finally, Section 6 concludes this paper.

2 Related Works

In this section, the generic SMOTE method is first introduced, which is illustrated in Fig. 1. Sample x_i is selected from the minority class as the root sample for synthesizing. Then one of the k (k is generally odd) nearest neighbor samples of x_i is randomly selected (x_{i3} is selected in this sample) as the auxiliary sample for synthesizing a new sample. Linear interpolation is performed between the root sample and auxiliary sample, as given in Eq. (1).

$$x_{new,attr} = x_{i,attr} + \gamma \times (x_{ij,attr} - x_{i,attr}) \tag{1}$$

where x_i and x_j represent the root sample and auxiliary sample. x_{new} is the new synthesized sample.

$x_i \in R^d$, and $x_{i,attr}$ is the value in the $attr$ dimension of $x_{i,attr} = 1, 2, \ldots, d, \gamma$ is a random variable between [0, 1].

As an effective method, many researches have been done based on SMOTE.

(1) Synthetic minority oversampling algorithm based on nearest neighbors (SMOM): A synthetic minority oversampling algorithm based on nearest neighbors SMOM is proposed in reference [23]. For the minority class sample, its k-nearest neighbor samples are set with different weights. A smaller weight is assigned to the sample's direction which may result in severe over-generalization.

Then Neighborhood-Based Density-Oriented Sampling (NBDOS) clustering and a double loop filter are applied to reduce the cost of distance computation. The security coefficient of the sample neighborhood for minority class oversampling (SSCMIO) is another way to avoid over-generalization [24], which makes oversampling based on the security coefficient of the neighborhood. A synthetic oversampling method with minority and majority class (SOMM) that combined samples by taking into account the neighbor features of both minority and majority classes is proposed in Reference [25]. It obtains better performance than SMOM. Heiringer Distance-guided SMOTE (HDSMOTE) guides sample synthesis and evaluation through Heiringer distance [26,27], to solve the problem of over generalization and class overlap.

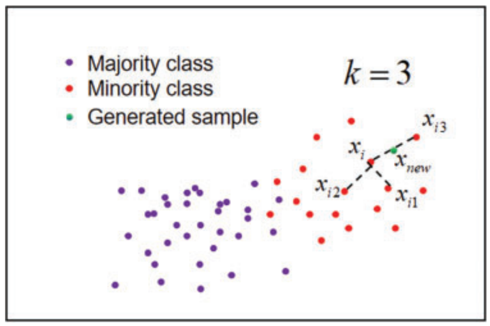


Figure 1: The basic principle of SMOTE

(2) Adaptive synthetic sampling approach (ADASYN) [28]: The basic idea is to use weighted distributions for different minority samples according to their learning difficulty, and to generate more synthetic data for minority samples compared to the easier-to-learn minority samples. Thus, the ADASYN method improves the learning ability of data distribution by reducing the bias due to class imbalance, and adaptively shifting the classification decision boundaries to difficult samples. In Borderline-SMOTE [29], different minority samples are given different weights for sample generation. The number of combinations for each minority sample is determined in Reference [30]. The adaptive synthetic sampling approach for nominal data (ADASYN-N) and adaptive synthetic sampling approach using k-nearest neighbors (ADASYN-KNN) make extensions to process nominal data types [31]. The nearest neighbor parameter k is estimated during class balancing [32].

(3) Sampling clustering and under-sampling technique (SCUT) [33]: This algorithm adopts undersampling and oversampling to reduce the imbalance between classes in a multiclass setup. Oversampling using SMOTE for minority classes generates synthetic samples. Under-sampling is used for the majority class, using a clustering-based under-sampling technique and the Expectation Maximization (EM) algorithm, which is suitable for scenarios with high imbalance ratios. Sampling clustering and under-sampling technique with under-sampling (SCUT-US) improve the SCUT by setting windows [34]. It balances the number of incoming samples of all classes and improves the recognition rate of minority class samples.

(4) Complexity-based synthetic technique (COSTE) [35]: Unlike the proximity-based SMOTE, this method first normalizes the data $min - max$, calculates and ranks the complexity of each sample, and then selects samples that are similar in complexity to synthesize samples. Combining pairs of defective samples with similar complexity to generate synthetic samples increases the diversity within the data, maintains the predictive model's ability to find defects, and takes into account the different testing efforts required for different samples. COSTE is also applied to the problem of multi-class unbalance [36].

(5) SMOTE-least squares support vector machine (SMOTE-LSSVM) [37]: This method first decomposes the multi-class problem, applies SMOTE to balance the data, and then optimizes the parameters of the least squares support vector machine (LSSVM) classifier using a combination of particle swarm optimization and gravitational search algorithms. This approach leverages the global search capability and the local search capability to enhance classifier performance. This method is validated using the breast cancer malignancy dataset.

(6) SMOTE-local outlier factor (SMOTE-LOF) [38]: This method combines the Local Outlier Factor (LOF) to identify noise in synthetic minority samples, addressing the noise issue that may arise when handling imbalanced data. Experimental results show that, compared to traditional SMOTE, SMOTE-LOF performs better in terms of accuracy and F-measure. Additionally, when dealing with large datasets with a smaller imbalance ratio, SMOTE-LOF also outperforms SMOTE in terms of AUC.

(7) Refined neighborhood-SMOTE (RN-SMOTE) [39]: The method begins by applying SMOTE to oversample the minority class, generating synthetic instances. It then employs the density-based spatial clustering of applications with noise (DBSCAN) algorithm to detect and eliminate noisy instances. After cleaning, the synthetic instances are reintegrated into the original dataset. SMOTE is subsequently reapplied to ensure the dataset remains balanced before being introduced to the classifier.

(8) Feature-weighted-SMOTE (FW-SMOTE) [40]: This method introduces a feature-weighted oversampling approach aimed at addressing the limitations of using Euclidean distance to define neighborhoods in high-dimensional spaces, as in traditional SMOTE. FW-SMOTE utilizes a weighted Minkowski distance to define neighborhoods for minority classes, giving greater priority to features that are more relevant to the classification task. Another advantage is its built-in feature selection capability, where attributes with weights below a threshold are discarded. This ensures the method avoids unnecessary complexity while effectively mitigating issues such as class overlap and hubness.

(9) DeepSMOTE [41]: DeepSMOTE, a novel oversampling algorithm designed specifically for deep learning models. It leverages the successful features of the SMOTE algorithm, using an encoder/decoder framework to produce high-quality synthetic images. DeepSMOTE enhances minority class data through SMOTE-based oversampling techniques. Furthermore, it employs a specialized loss function augmented with a penalty term to optimize the generation, ensuring that the artificial images are both information-rich and suitable for visual inspection, without the need for a discriminator. This streamlined and effective design is particularly adept at addressing class imbalance issues in image data.

3 RE-SMOTE Model

This paper proposes a novel unbalanced data processing method based on PF-SMOTE and SMOTE-WENN. The minority class samples are first divided into safe minority and boundary minority as given in Section 3.1. Data synthesis and data cleaning are described in detail in Sections 3.2 and 3.3, respectively.

3.1 Data Division

The dataset is divided into safe minority and boundary minority categories, with the following definitions. For a given dataset $D \in R^d$, it consists of both minority class samples and majority class samples. The minority class dataset is denoted as D^+, and the majority class dataset is denoted as D^-.

If $x_i \in D$, $x_j \in D$, $d(x_i, x_j)$ indicates the distance between x_i and x_j, d_{min} represents the minimum distance between x_i and its nearest neighbor in D, the nearest neighbor of x_i in D is denoted as x_{nn}, $x_{nn} = \{x_j \in D \mid d(x_i, x_j) = d_{min}\}$.

Definition 1 (Boundary minority sample): If $x \in D^+$, $x_{nn} \in D^-$, then x is a boundary minority sample, as demonstrated in Fig. 2a.

Definition 2 (Safe minority sample): If $x \in D^+$, $x_{nn} \in D^+$, then x is a safe minority sample, as demonstrated in Fig. 2b.

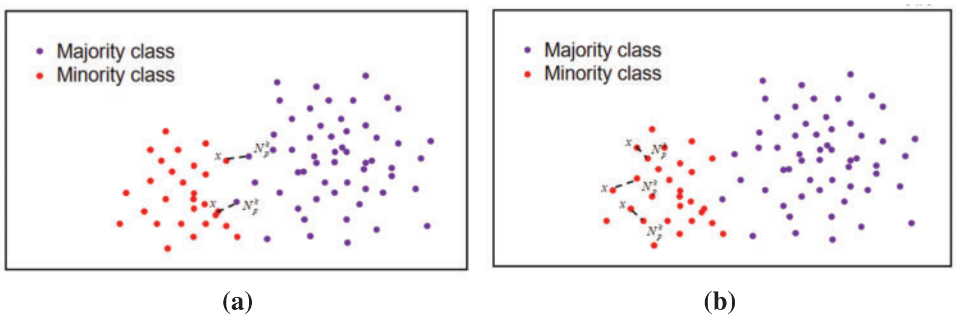


Figure 2: Examples of two types of minority samples. (a) Example diagram of boundary minority sample. (b) Example diagram of safe minority sample

For each of the minority class samples x_i, $x_i \in D^+$, if the nearest neighbor of the minority class sample $x_{nn} \in D^+$, then x_i is added to the safe minority class set D^+_{safe}. Otherwise, if $x_{nn} \in D^-$, then the x_i is added to the boundary minority class set $D^+_{boundary}$.

3.2 Data Synthesis

As can be seen from Fig. 2a,b, boundary minority samples and safe minority samples have heterogeneous characteristics, so multiple strategies should be taken into account for data sample synthesis. We aim to increase the diversity of synthesized samples and expand the boundary of the minority class. Meanwhile, for safe minority class samples, it is necessary to increase the local area as much as possible and avoid generating duplicate examples and noisy samples.

For the safety minority sample synthesis, the location of the nearest majority samples for one safety minority sample should be found. Then, a new safety minority sample is synthesized randomly within the formed circle by taking this safety minority sample as the center and the distance from the nearest majority class sample as the radius. This procedure is illustrated in Fig. 3a.

f^{safe}_{gen} represents the synthesized sample by safe minority class and is computed with Eq. (2). Unlike the linear interpolation synthesis of the safety minority sample in PF-SMOTE, we randomly synthesize a new sample within a local area, in which the minority class sample point is the center of a circle and the radius is the distance from the nearest majority class sample to the minority class sample.

$$f^{safe}_{gen} = r\sin\theta + x_i, \theta \in U(0, 2\pi), r \in U(0, d|x_i - x^-_{nn}|) \tag{2}$$

For the boundary minority sample synthesis, the boundary toward the majority class sample is extended, i.e., the position of the synthesized sample is biased towards the position of the majority class sample. Specifically, for a boundary minority sample, its nearest majority class sample is first found, and a boundary minority class sample is synthesized by interpolating along the line between boundary minority class sample point and the nearest majority class sample point. The Gaussian process is employed to enhance the diversity of the synthesized samples. This procedure is shown in Fig. 3b. $f_{gen}^{boundary}$ represents the synthesized sample by boundary minority class as given in Eqs. (3) through (6).

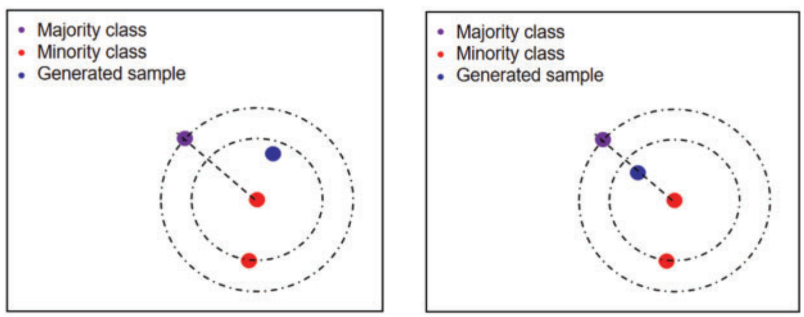
$$f_{gen}^{boundary} = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x_i - \mu)^2}{2\sigma^2}\right) \tag{3}$$

$$N((x_i + gap \times (x_{nn}^- - x_i)), \sigma^2) \tag{4}$$

$$w.r.t.\mu = x_i + gap \times (x_{nn}^- - x_i)), gap \sim U(0, 1) \tag{5}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{|D^+|}(x_i - \overline{x})^2}{|D^+| - 1}} \tag{6}$$

where x_{nn}^- is the nearest majority sample points of x_i. $|D^+|$ is the size of data set, and *gap* follows a uniform distribution. Algorithm 1 provides the pseudo-code for data synthesis.



(a) **(b)**

Figure 3: Examples of minority class synthesis for safe and boundary samples. (a) Example diagram of security minority class synthesis. (b) Example of sample synthesis of the boundary minority class

3.3 Data Cleaning

Noise is inevitably introduced during the synthesis of new samples, potentially degrading the quality of both sample attributes and class labels and adversely affecting model performance. To mitigate this, data cleaning is applied to all samples, including both majority and minority class samples, based on two key aspects: the WENN method and relative data density.

(1) WENN

WENN addresses class imbalance and the small sample problem through a distance scaling function. By applying different distance scaling for positive and negative candidate neighbors, WENN effectively preserves a higher proportion of safe minority and safe majority samples.

Algorithm 1: Sample synthesis
Input: minority class samples D^+, majority class samples D^- **Output:** synthetic samples f_{gen}

(Continued)

Algorithm 1 (continued)

$D^{+}_{safe}, D^{+}_{boundary} \leftarrow \varnothing$
for all $x_i \in D^{+}$ do
    Find its nearest neighbor x_{nn}
    if $x_{nn} \in D^{+}$ then
        $D^{+}_{safe} \leftarrow D^{+}_{safe} \cup x_i$
    else if $x_{nn} \in D^{-}$ then
        $D^{+}_{boundary} \leftarrow D^{+}_{boundary} \cup x_i$
    end if
end for
$f^{safe}_{gen} \leftarrow \varnothing$
for all $x_i \in D^{+}_{safe}$ do
    Find its nearest majority class sample $x^{-}_{nn} \in D^{-}$
    generate synthetic samples x^{safe}_{gen} according to Eq. (2)
    $f^{safe}_{gen} \leftarrow f^{safe}_{gen} \cup x^{safe}_{gen}$
end for
$f^{boundary}_{gen} \leftarrow \varnothing$
for all $x_i \in D^{+}_{boundary}$ do
    Find its nearest majority class sample $x^{-}_{nn} \in D^{-}$
    generate synthetic samples $x^{boundary}_{gen}$ according to Eqs. (3)–(6)
    $f^{boundary}_{gen} \leftarrow f^{boundary}_{gen} \cup x^{boundary}_{gen}$
end for
return $f_{gen} \leftarrow f^{safe}_{gen} \cup f^{boundary}_{gen}$

In WENN, the distance between two samples is defined using the isomorphic value difference metric, as described in Eq. (7).

$$d_{HVDM}(x_1, x_2) = \sqrt{\sum_{a=1}^{m} d_a^2 \left(x_{1,a}, x_{2,a}\right)} \tag{7}$$

where x_1 and x_2 represent the feature vectors of two samples. a denotes the attribute index, and m is the number of attributes. The distance scaling function of WENN d is shown in Eq. (8), where N^{-}, N^{+} and N represent the sizes of the majority class sample, minority class sample, and total sample.

$$d(x_1, x_2) = \begin{cases} e^{\left(\frac{N^{+}}{N}\right)^{m}} \cdot d_{HVDM}(x_1, x_2), x_2 \in D^{+} \\ e^{\left(\frac{N^{-}}{N}\right)^{m}} \cdot d_{HVDM}(x_1, x_2), x_2 \in D^{-} \end{cases} \tag{8}$$

Here, x_2 is the k closest sample of x_1 (i.e., $k = 3$). k nearest neighbors of x_1 can be found by computing all distance scaling functions of these nearest samples. K_i^{+} denotes the number of majority class samples in the k nearest neighbors of x_i, K_i^{-} denotes the number of minority class samples in the k nearest neighbors of x_i. Data samples are cleaned up according to the rules given in Eq. (9).

$$\begin{cases} K_i^{+} > K_i^{-}, x_i \in D^{-} \\ K_i^{+} < K_i^{-}, x_i \in D^{+} \end{cases} \tag{9}$$

Fig. 4a shows an unscaled distance sample, where the k ($k = 3$) nearest neighbors of x_1 are x_2, x_3, x_4. x_2, x_3 are minority class samples and x_4 is a majority class sample. In this case, if x_1 is the minority class sample, then it should be retained. In Fig. 4b, after scaling with distance weights, the k ($k = 3$) neighbors of x_1 are x_3, x_4, x_5. x_3 is a minority class sample and x_4, x_5 are majority class samples. If x_1 is a majority class sample, then x_1 should be retained.

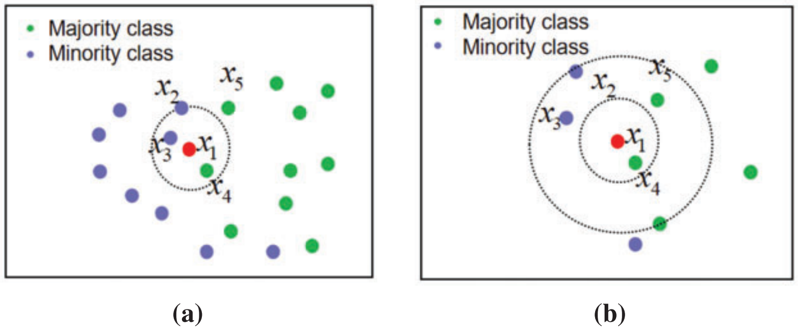


Figure 4: Example of distance scaling weights. (a) Unscaled distance sample. (b) Scaled distance sample with weights

(2) Relative data density

Besides WENN, we design another rule for data cleaning based on relative data density. A reverse k-nearest neighbor is adopted to determine the relative data density. The inverse k-nearest neighbor is defined as: The reverse nearest neighbor of query point q is the set of all data points in data set D whose distance from q does not exceed the kth nearest neighbor of q. It can be noted as $RkNN(q, k) = \{p \in D(p, q) <= dist(q, p')\}$, where p' is the kth nearest neighbor of q in dataset D.

Fig. 5 shows the reverse k neighbor sample diagram. As shown in Fig. 5a, when $k = 3$, the query sample has 3 nearest neighbors A, B, and D. In Fig. 5b, the query sample has 4 reverse nearest neighbors A, C, B and D for it belongs to the 3 nearest neighbors of these samples. Thus, there are two significant differences between kNN and $RkNN$: kNN contains a specific number of samples, while $RkNN$ may contain from 0 to an infinite number of samples. Data samples are cleaned up according to the rules given by Eq. (10). Algorithm 2 provides the pseudo-code for data cleaning.

$$\begin{cases} K_i^- > 2 \times K_i^+, x_i \in D^+ \\ K_i^- < 2 \times K_i^+, x_i \in D^- \end{cases} \tag{10}$$

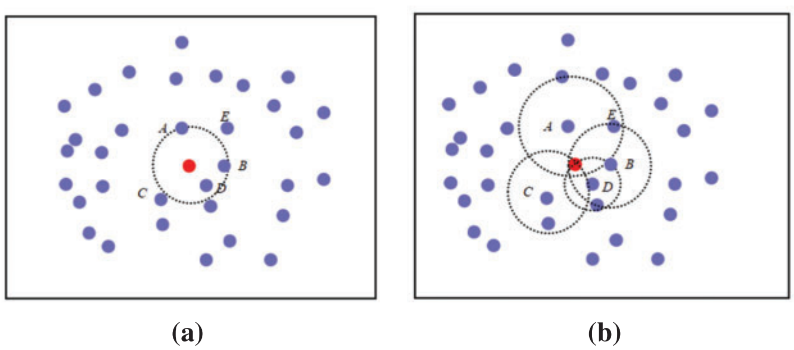


Figure 5: Illustration of k-nearest neighbor and reverse k-nearest neighbor samples. (a) k-nearest neighbor sample diagram. (b) Reverse k-nearest neighbor sample diagram

Algorithm 2: Data cleaning

Input: Synthesized data set $S \leftarrow D \cup f_{gen}$
Output: Cleaned dataset F

```
X_WENN, X_RKNN ← ∅
for all x_i ∈ S do
    Find its k nearest neighbors and calculate the distance according to Eq. (7)
    Compute weighted distances according to Eq. (8)
     if delete x_i according to Eq. (9)
       X_WENN ← X_WENN ∪ x_i
     end if
  end for
  for all x_i ∈ S do
      Find its reverse k nearest neighbor
      if delete x_i according to Eq. (10)
        X_RkNN ← X_RkNN ∪ x_i
      end if
    end for
    return F ← S − (X_WENN ∪ X_RkNN)
```

4 Experimental Framework

4.1 Data Set

In this section, 40 unbalanced datasets from the KEEL repository are used to evaluate the performance of RE-SMOTE. Table 1 provides details of the datasets, including the numbers of attributes (Attr.) and examples (NE), the number of each class (%Class(maj,min)), and the imbalance ratio (IR).

Table 1: Summary descriptions of the datasets

id	Datasets	#Attr.	#NE	%Class (maj,min)	#IR
D1	ecoli1	7	336	(259,77)	3.364
D2	ecoli2	7	336	(284,52)	5.462
D3	ecoli3	7	336	(301,15)	8.600
D4	ecoli4	7	336	(316,20)	15.800
D5	ecoli-0$_v s_1$	7	220	(143,77)	1.857
D6	glass-0-1-2-3$_v s_4 - 5 - 6$	9	214	(163,51)	3.196
D7	glass0	9	214	(144,70)	2.057
D8	haberman	3	306	(225,81)	2.778
D9	vehicle2	18	846	(628,218)	2.881
D10	yeast6	8	1484	(1449,35)	41.400
D11	wisconsin	9	683	(444,239)	1.858
D12	new-thyroid1	5	215	(180,35)	5.143
D13	glass6	9	214	(185,29)	6.379
D14	page-blocks0	10	5472	(4913,559)	8.789

(Continued)

Table 1 (continued)

id	Datasets	#Attr.	#NE	%Class (maj,min)	#IR
D15	yeast1	8	1484	(1055,429)	2.459
D16	australian	14	690	(383,307)	1.248
D17	bupa	6	345	(200,145)	1.379
D18	heart	13	270	(150,120)	1.250
D19	vehicle0	18	846	(649,199)	3.251
D20	yeast3	8	1484	(1321,163)	8.104
D21	new-thyroid2	5	215	(180,35)	5.143
D22	glass1	9	214	(138,76)	1.816
D23	vowel0	13	988	(898,90)	9.978
D24	vehicle3	18	846	(634,212)	2.991
D25	yeast-2$_{v}s_{8}$	8	482	(462,200)	23.100
D26	segment0	19	2308	(1979,329)	6.015
D27	yeast4	8	1484	(1433,51)	28.098
D28	ring	20	740	(373,367)	1.016
D29	dermatology-6	34	358	(338,20)	16.900
D30	vehicle1	18	846	(629,217)	2.899
D31	pima	8	768	(500,268)	1.866
D32	yeast5	8	1484	(1440,44)	32.727
D33	poker-8$_{v}s_{6}s$	10	1477	(1460,17)	85.882
D34	magic	10	1902	(1234,668)	1.847
D35	shuttle-2$_{v}s_{5}$	9	3316	(3627,49)	66.673
D36	winequality-red-4	11	1599	(1546,53)	29.170
D37	hepatitis	19	80	(67,13)	5.154
D38	ecoli-0-6-7$_{v}s_{5}$	6	220	(200,20)	10.000
D39	shuttle-6$_{v}s_{2}-$ 3	9	230	(220,10)	22.000
D40	winequality-red-8$_{v}s_{6}-$ 7	11	855	(837,18)	46.500

Repeated stratified k-fold cross-validation is employed, using ten replicates of 10-fold cross-validation, resulting in 100 models being fitted and evaluated. The dataset is divided into 10 subsets, each containing 10% of the samples. In each iteration, one subset is used as the test set, while others are used for training. The average of the 10 repetitions is considered as the final performance metric.

4.2 Performance Metrics

While classification accuracy is often used to evaluate algorithm performance, it is not an ideal metric for imbalanced datasets due to the skewed class distribution. Unlike standard metrics, which assume equal importance for all classes, imbalanced classification problems typically prioritize minimizing classification errors in the minority class over the majority class. As a result, performance metrics must focus on the minority class, which poses a challenge due to the limited representation of minority class observations, making it harder to train an effective model. Therefore, we employ two widely recognized metrics, AUC and F1 score [21]. To better explain AUC and F1, the confusion matrix of a dichotomous problem is shown in Table 2, and the corresponding concepts are given as follows:

TP (True Positive). The number of positive class samples that were predicted as positive class.

FN (False Negative). The number of positive class samples that were predicted as negative class.

FP (False Positive). The number of negative class samples that were predicted as positive class.

TN (True Negative). The number of negative class samples that were predicted as negative class.

Table 2: Confusion matrix

	Positive prediction	Negative prediction
Positive class	TP	FN
Negative class	FP	TN

Based on the confusion matrix, ROC curves can be drawn on different thresholds. ROC curves, also called subject working characteristic curves, are composed of True Positive Rate (TPR) and False Positive Rate (FPR) at different classification thresholds, as given in Eqs. (11) and (12). TPR is the vertical axis and FPR is the horizontal axis. Each threshold corresponds to a (FPR, TPR) point, which is depicted as the ROC curve. The closer the ROC curve is to the upper left corner, the higher the model's accuracy.

$$TPR = \frac{TP}{TP + FN} \tag{11}$$

$$FPR = \frac{FP}{FP + TN} \tag{12}$$

AUC is the area under the ROC curve. The larger the area under the ROC curve, the better the model. There are two obvious advantages of AUC. First, AUC does not focus on specific scores. It reflects relative results such as ranking relations. Second, AUC is an overall indicator and does not focus on the local characteristics of the model, so it is not sensitive to the sample. AUC can be represented as Eq. (13), where (x_i, y_i) is the coordinate and n is the number of points.

$$AUC = \frac{1}{2} \sum_{i=1}^{n-1} (x_{i+1} - x_i) \cdot (y_i + y_{i+1}) \tag{13}$$

The F1 value is the summed average of precision and recall, as given in Eq. (14). It is close to the smaller of these two values. If the F1 value is large, then precision and recall must be large. F1 can reflect the algorithm's overall performance.

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{14}$$

4.3 Classification Algorithm

To evaluate the proposed RE-SMOTE model, 3 well-known classifiers are selected in this experiment, which are described in detail as follows:

• Decision Tree (DT) [42,43]. It can be applied to both classification and regression, with the leaf nodes being the final decision result. It starts with a root node containing all the training data and then continuously refines the internal nodes using specific division criteria until the stopping conditions are satisfied. Thus, classification rules can be obtained inductively. The training is to construct a tree based on the given dataset and select the most valuable feature-slicing nodes. Decision tree is easy to understand and interpret. The data preparation is simple. It can be used constructed for data containing many attributes. Moreover, the decision tree scales well to large databases while its size is independent of the dataset.

• Support Vector Machine (SVM) [44]. The basic idea is to find the best-separating hyperplane in the feature space to maximize the interval between positive and negative samples in the training set, and with the power of kernel functions, SVM can also be used to solve nonlinear problems. The SVM classifier can be adapted to small training datasets and easy-to-fit high-dimensional samples. In addition, it can also handle the problem of neural network structure selection and local minima prevention.

• K-Nearest Neighbor (KNN) [45]. KNN is used for classification by measuring the distance between different feature values. It is based on the idea that a sample belongs to a class if the majority of the k most similar (i.e., most neighboring) samples in the feature space belong to that class. k is usually an integer no greater than 20. The neighbors selected in the KNN algorithm are correctly classified objects. It relies only on the categories of the nearest neighbors to determine the classification of the samples. KNN is simple and low-cost, and the training time and space are linearly related to the size of the training data set. It is more suitable for the set of samples to be divided with more crossover or overlap of class domains since it relies on a limited number of neighboring samples around.

4.4 Comparison Models

To verify the performance of the RE-SMOTE method, three SMOTE variants are selected for experimental comparisons. A brief descriptions of these methods are given as follows:

• SMOTE-ENN [46]: It starts with oversampling samples of minority class using SMOTE, and then performs local data cleaning using ENN. If the predicted label of k-nearest neighbors (KNN) is different from the true label, the sample is considered noisy and deleted, otherwise, the sample is retained.

• SMOTE-WENN [20]: It designs a new data cleaning method WENN. WENN uses a weighted distance function and KNN rules to detect and remove unsafe majority and minority samples. The weighted distance function extends a suitable distance by considering local imbalance and spatial sparsity.

• PF-SMOTE [19]: It is a parameter-free variant of SMOTE that generates a sufficient number of representative synthetic samples based on bounded minority and the safe minority classes while avoiding the generation of interpolated noisy samples.

• SMOTE-RkNN [47]: This method introduces an improved SMOTE hybrid algorithm called SMOTE-reverse k-nearest neighbors (SMOTE-RkNN). The algorithm identifies noise based on probability density rather than relying on local neighborhood information.

5 Experimental Results and Discussions

To demonstrate the effectiveness of RE-SMOTE, the experiments are conducted in two aspects. (1) In Section 5.1, visualization results on the synthesis of samples are provided. (2) In Section 5.2, comparisons of other well-known variants of the SMOTE methods are given.

5.1 Visualization of Synthetic Samples

Among the 40 datasets in Table 1, we randomly select two datasets for visualization, numbered D1 and D23, for visualization. For these comparative methods, we set the k-nearest neighbors parameter to 3 based on empirical experience. Other parameters will be adjusted and calculated according to the characteristics of different datasets.

The original dataset, the balanced dataset after the SMOTE, and the balanced dataset after RE-SMOTE are listed, respectively. In this way, the regions where RE-SMOTE generates samples can be visually displayed. The final visualization plot of applying SMOTE and RE-SMOTE to a two-dimensional data set is shown in Fig. 6, where blue points represent the minority class samples and red points represent the majority class samples. Original data sample plots are given in Fig. 6a,d. Balanced data sample plots after SMOTE are given in Fig. 6b,e.

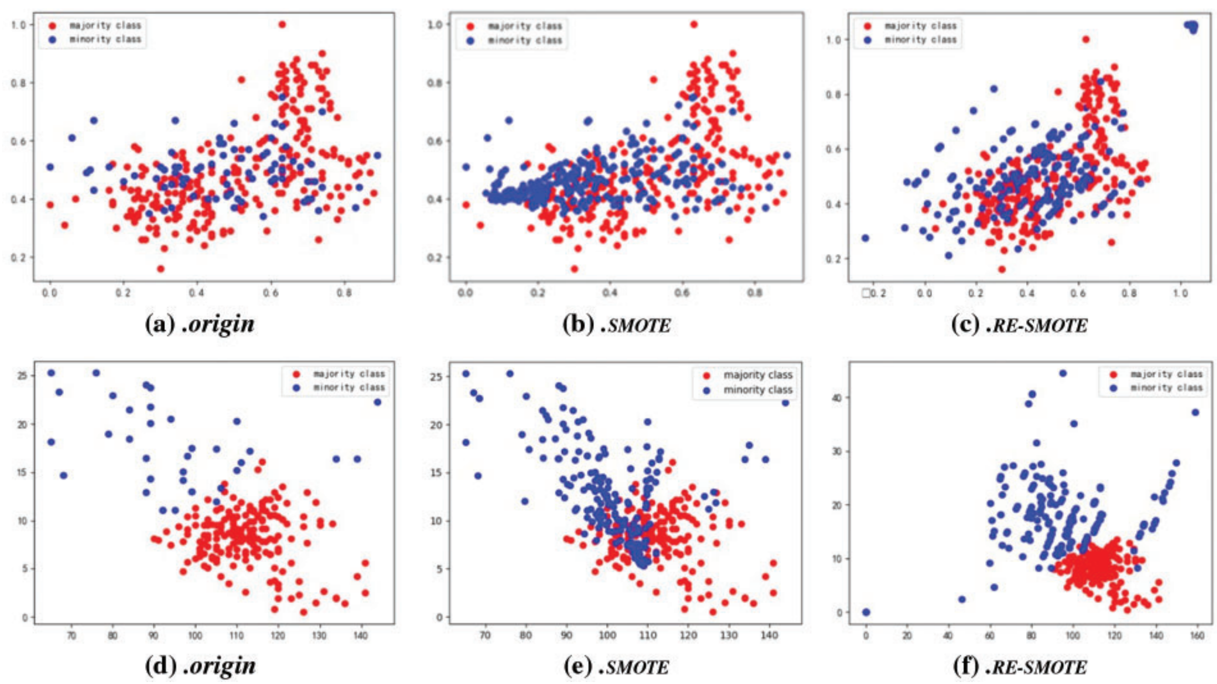


Figure 6: Visualization comparison of RE-SMOTE with SMOTE on 2 datasets. (a) Visualization of the original dataset (D1). (b) Balanced dataset after SMOTE (D1). (c) Balanced dataset after RE-SMOTE (D1). (d) Visualization of the original dataset (D23). (e) Balanced dataset after SMOTE (D23). (f) Balanced dataset after RE-SMOTE (D23)

We randomly select a minority class sample and interpolate between the nearest minority class neighbors to synthesize a new minority class sample. SMOTE is a linear interpolation to synthesize the new sample. This leads to a more convergent distribution of the synthesized minority class sample than the original one. Balanced data samples performed by RE-SMOTE are given in Fig. 6c,f. In contrast, RE-SMOTE focuses on the diversity of the synthesized samples and favors the synthesis of minority-class samples. In general, it can be concluded that RE-SMOTE is more effective in sample synthesis.

5.2 Comparison of Different Methods

The comparison results on AUC and F1 with DT-based, KNN-based and SVM-based classifiers are shown in Tables 3–12, respectively. Through these results, it can be observed that the proposed RE-SMOTE outperforms other variants in most cases. Specifically, the proportion of best results with RE-SMOTE on 40 data sets is shown in Fig. 7.

Table 3: Summary table of comparison results

	SMOTE	SMOTE-ENN	SMOTE-WENN	PF-SMOTE	SMOTE-RkNN	RE-SMOTE
AUC with DT-based classifier	0.9000	0.9248	0.9352	0.9142	**0.9599**	0.9588
F1 with DT-based classifier	0.9005	0.9212	0.9371	0.9073	0.9423	**0.9582**
AUC with KNN-based classifier	0.9114	0.9424	0.9691	0.9388	0.9730	**0.9882**
F1 with KNN-based classifier	0.8701	0.8984	0.9450	0.9109	0.9596	**0.9614**
AUC with SVM-based classifier	0.8932	0.9157	0.9378	0.9215	0.9447	**0.9626**
F1 with SVM-based classifier	0.8003	0.7828	0.8832	0.8020	**0.8843**	0.8679

Table 4: Comparison results on AUC with DT-based classifier (The best results in each dataset are shown in bold)

id	SMOTE	SMOTE-ENN	SMOTE-WENN	PF-SMOTE	SMOTE-RkNN	RE-SMOTE
D1	0.8782	0.9154	0.9454	0.9225	0.9620	**0.9756**
D2	0.9075	0.9276	0.9423	0.9557	0.9685	**0.9759**
D3	0.8979	0.9319	0.9649	0.9434	0.9815	**0.9842**
D4	0.9643	0.9631	0.9676	0.9825	0.9860	**0.9876**
D5	0.8822	0.9183	0.9282	0.9813	**0.9950**	0.9929
D6	0.8927	0.9333	0.9196	0.9637	0.9600	**0.9759**
D7	0.8496	0.9052	**0.9343**	0.8524	0.9235	0.9091
D8	0.7355	0.8861	0.9277	0.7471	0.9210	**0.9328**
D9	0.9520	0.9617	0.9588	0.9678	**0.9750**	0.9646

(Continued)

Table 4 (continued)

id	SMOTE	SMOTE-ENN	SMOTE-WENN	PF-SMOTE	SMOTE-RkNN	RE-SMOTE
D10	0.9741	0.9801	0.9868	0.9802	0.9920	**0.9964**
D11	0.9584	**0.9855**	0.9811	0.9604	0.9840	0.9777
D12	0.9728	0.9686	0.9783	**0.9892**	0.9800	0.9638
D13	0.9562	0.9669	0.9739	0.9705	0.9810	**0.9856**
D14	0.9668	0.9763	0.9875	0.9857	**0.9950**	0.9931
D15	0.7649	0.8451	0.8917	0.7963	0.9010	**0.9020**
D16	0.8266	0.9103	0.8996	0.8484	0.9120	**0.9181**
D17	0.6453	0.6613	0.6623	0.7089	0.8600	**0.8698**
D18	0.7536	0.7496	0.7514	0.7812	0.8420	**0.8501**
D19	0.9553	0.9658	**0.9756**	0.9551	0.9850	0.9683
D20	0.9741	0.9712	0.9826	0.9556	0.9911	**0.9919**
D21	0.9485	0.9483	0.9804	**0.9882**	0.9800	0.9683
D22	0.7899	0.7922	0.8555	0.8389	0.8630	**0.8701**
D23	0.9897	0.9884	0.9908	**0.9910**	0.9900	0.9860
D24	0.8125	0.9047	0.8900	0.8229	0.9030	**0.9050**
D25	0.9467	0.9689	0.9580	0.9652	0.9995	**1.0000**
D26	0.9926	0.9945	0.9940	0.9961	0.9965	**0.9969**
D27	0.9470	0.9549	0.9722	0.9558	0.9880	**0.9922**
D28	0.8414	0.7844	0.7813	**0.8571**	0.8550	0.8226
D29	0.9954	0.9970	**0.9997**	0.9962	0.9990	0.9945
D30	0.8126	0.8952	0.8953	0.8159	**0.9320**	0.9210
D31	0.7137	0.8412	0.8692	0.7593	0.9300	**0.9151**
D32	0.9844	0.9874	0.9977	0.9894	0.998	0.9952
D33	0.9956	0.9957	0.9877	0.9252	0.9985	**0.9987**
D34	0.7656	0.8356	0.8457	0.8086	0.925	**0.9231**
D35	0.9997	0.9996	0.9997	**1.0000**	**1.0000**	0.9997
D36	0.9564	0.9673	0.9621	0.8379	0.9980	**0.9985**
D37	0.8849	0.8946	0.9364	0.8697	0.9710	**0.9712**
D38	0.9625	0.9579	0.9697	0.9759	0.9810	**0.9869**
D39	0.9889	0.9923	0.9947	**1.0000**	0.9980	0.9957
D40	0.9652	0.9723	0.9691	0.9270	0.9980	**0.9984**

Table 5: Comparison results on F1 with DT-based classifier (The best results in each dataset are shown in bold)

id	SMOTE	SMOTE-ENN	SMOTE-WENN	PF-SMOTE	SMOTE-RkNN	RE-SMOTE
D1	0.8760	0.9166	0.9528	0.9187	0.9497	**0.9770**
D2	0.9071	0.9262	0.9492	0.9521	0.9476	**0.9754**
D3	0.8995	0.9308	0.9671	0.9282	0.9648	**0.9820**
D4	0.9643	0.9648	0.9706	0.9801	0.9749	**0.9872**

(Continued)

Table 5 (continued)

id	SMOTE	SMOTE-ENN	SMOTE-WENN	PF-SMOTE	SMOTE-RkNN	RE-SMOTE
D5	0.8829	0.9179	0.9319	0.9768	0.9693	**0.9928**
D6	0.8949	0.9353	0.9252	0.9641	0.9596	**0.9735**
D7	0.8474	0.9110	**0.9391**	0.8519	0.9291	0.9130
D8	0.7368	0.8740	**0.9425**	0.7222	0.9197	0.9313
D9	0.9508	0.9603	0.9607	0.9671	**0.9798**	0.9642
D10	0.9740	0.9804	0.9884	0.9725	0.9846	**0.9962**
D11	0.9568	**0.9851**	0.9806	0.9603	0.9743	0.9777
D12	0.9728	0.9656	0.9801	**0.9884**	0.9747	0.9658
D13	0.9580	0.9668	0.9747	0.9700	0.9729	**0.9847**
D14	0.9663	0.9756	0.9879	0.9863	0.9842	**0.9931**
D15	0.7637	0.8382	0.8987	0.7862	0.8896	**0.9032**
D16	0.8269	0.9029	0.8922	0.8647	0.8893	**0.9122**
D17	0.6499	0.6700	0.6709	0.7431	0.6598	**0.8690**
D18	0.7967	0.7993	0.8043	0.8477	0.7997	**0.9093**
D19	0.9555	0.9649	**0.9778**	0.9541	0.9698	0.9688
D20	0.9476	0.9711	0.9847	0.9485	0.9797	**0.9924**
D21	0.9469	0.9472	0.9855	**0.9862**	0.9746	0.9695
D22	0.7961	0.7818	0.8555	0.8406	0.8398	**0.8621**
D23	0.9897	0.9883	0.9899	**0.9917**	0.9878	0.9863
D24	0.8132	0.8983	0.9002	0.8020	0.8949	**0.9014**
D25	0.9478	0.9690	0.9704	0.9523	0.9648	**1.0000**
D26	0.9925	0.9944	0.9940	0.9959	0.9947	**0.9966**
D27	0.9468	0.9549	0.9787	0.9329	0.9748	**0.9925**
D28	0.8358	0.6427	0.6482	**0.8888**	0.8391	0.7526
D29	0.9955	0.9966	**0.9997**	0.9956	0.9979	0.9948
D30	0.8135	0.8927	0.9071	0.7957	0.8997	**0.9210**
D31	0.7090	0.8383	0.8771	0.7605	0.8699	**0.9179**
D32	0.9845	0.9877	**0.9979**	0.9871	0.9948	0.9949
D33	0.9955	0.9958	0.9917	0.8804	0.9897	**0.9988**
D34	0.7655	0.8224	0.8425	0.8115	**0.9293**	0.9191
D35	0.9997	0.9996	0.9997	**1.0000**	0.9993	0.9996
D36	0.9566	0.9666	0.9758	0.7093	0.9699	**0.9986**
D37	0.8872	0.8943	0.9442	0.8396	0.9598	**0.9726**
D38	0.9632	0.9587	0.9756	0.9755	0.9746	**0.9876**
D39	0.9885	0.9916	0.9947	**1.0000**	0.9939	0.9957
D40	0.9652	0.9727	0.9786	0.8670	0.9697	**0.9982**

Table 6: Results of Wilcoxon signed-rank tests for comparing RE-SMOTE and the well-known variants of SMOTE when DT is used as the classifier

Comparison	AUC			F1		
	R^+	R^-	p-value	R^+	R^-	p-value
RE-SMOTE *vs*. SMOTE	693	127	2.6124e–07	693	127	1.8109e–08
RE-SMOTE *vs*. SMOTE-ENN	745	75	5.8444e–09	722	98	2.0111e–07
RE-SMOTE *vs*. SMOTE-WENN	631	189	2.2117e–05	623	197	0.000149
RE-SMOTE *vs*. PF-SMOTE	624	196	9.5838e–07	624	196	3.1946e–06
RE-SMOTE *vs*. SMOTE-RkNN	620	200	1.2345e–05	615	205	2.3456e–05

Table 7: Comparison results on AUC with KNN-based classifier (The best results in each dataset are shown in bold)

id	SMOTE	SMOTE-ENN	SMOTE-WENN	PF-SMOTE	SMOTE-RkNN	RE-SMOTE
D1	0.9473	0.9829	0.9955	0.9664	0.9951	**0.9998**
D2	0.9539	0.9652	0.9921	0.9837	0.9918	**0.9998**
D3	0.9468	0.9693	0.9973	0.9676	0.9972	**0.9998**
D4	0.9844	0.9867	**1.0000**	0.9945	0.9999	**1.0000**
D5	0.9562	0.9644	0.9936	0.9927	0.9924	**0.9996**
D6	0.9502	0.9654	0.9889	0.9846	0.9886	**0.9994**
D7	0.9169	0.9858	0.9841	0.9134	0.9837	**0.9880**
D8	0.7587	0.8628	0.9774	0.8559	0.9768	**0.9870**
D9	0.9570	0.9749	0.9915	0.9810	0.9907	**0.9960**
D10	0.9834	0.9869	0.9984	0.9904	0.9983	**1.0000**
D11	0.9882	**0.9988**	0.9983	0.9871	0.9976	0.9969
D12	0.9648	0.9638	0.9990	0.9989	0.9990	**0.9997**
D13	0.9713	0.9755	0.9939	0.9876	0.9938	**0.9989**
D14	0.9786	0.9839	0.9980	0.9978	0.9980	**0.9996**
D15	0.8338	0.9411	0.9820	0.8711	0.9819	**0.9931**
D16	0.6802	0.9514	0.9672	0.7709	0.9664	**0.9823**
D17	0.6894	0.6891	0.6886	0.7881	0.7762	**0.9760**
D18	0.6246	0.7175	0.7287	0.7726	0.7271	**0.9668**
D19	0.9618	0.9769	0.9959	0.9821	**0.9970**	0.9965
D20	0.9743	0.9878	0.9992	0.9815	0.9990	**0.9995**
D21	0.9246	0.9307	0.9936	**1.0000**	0.9934	**1.0000**
D22	0.8990	0.9338	0.9883	0.9342	0.9879	**0.9931**
D23	0.9987	0.9987	**1.0000**	**1.0000**	0.9999	**1.0000**
D24	0.8530	0.9342	0.9798	0.8710	0.9795	**0.9897**
D25	0.9762	0.9757	0.9981	0.9810	0.9979	**1.0000**
D26	0.9930	0.9939	0.9989	0.9975	0.9988	**0.9998**
D27	0.9694	0.9742	0.9981	0.9770	0.9980	**0.9994**

(Continued)

Table 7 (continued)

id	SMOTE	SMOTE-ENN	SMOTE-WENN	PF-SMOTE	SMOTE-RkNN	RE-SMOTE
D28	0.7424	0.7089	0.6935	**0.8146**	0.6928	0.7137
D29	0.9743	0.9779	0.9927	0.9938	0.9925	**0.9985**
D30	0.8696	0.9558	0.9748	0.8727	0.9746	**0.9859**
D31	0.7739	0.9215	0.9784	0.8518	0.9779	**0.9839**
D32	0.9887	0.9911	0.9996	0.9954	**1.0000**	0.9999
D33	0.9904	0.9900	0.9973	0.9467	0.9969	**1.0000**
D34	0.7909	0.8821	0.9673	0.8703	**0.9970**	0.9887
D35	0.9998	0.9997	0.9998	**1.0000**	0.9997	**1.0000**
D36	0.9235	0.9300	0.9940	0.8394	0.9938	**1.0000**
D37	0.8514	0.8439	0.9498	0.9232	0.9989	**0.9992**
D38	0.9770	0.9836	0.9984	0.9859	0.9982	**0.9994**
D39	**1.0000**	0.9985	**1.0000**	**1.0000**	0.9999	**1.0000**
D40	0.9402	0.9441	0.9944	0.9328	0.9939	**1.0000**

Table 8: Comparison results on F1 with KNN-based classifier (The best results in each dataset are shown in bold)

id	SMOTE	SMOTE-ENN	SMOTE-WENN	PF-SMOTE	SMOTE-RkNN	RE-SMOTE
D1	0.9116	0.9534	0.9850	0.9374	0.9643	**0.9944**
D2	0.9317	0.9403	0.9819	0.9704	0.9763	**0.9966**
D3	0.9275	0.9437	0.9946	0.9411	0.9853	**0.9956**
D4	0.9738	0.9745	0.9966	0.9918	0.9981	**0.9984**
D5	0.9014	0.9211	0.9794	0.9890	0.9763	**0.9966**
D6	0.9001	0.9378	0.9650	0.9714	0.9728	**0.9885**
D7	0.8535	0.9482	0.9612	0.8561	0.9638	**0.9645**
D8	0.7028	0.7892	0.9534	0.7854	0.9534	**0.9589**
D9	0.9121	0.9389	0.9726	0.9482	0.9719	**0.9830**
D10	0.9697	0.9748	0.9982	0.9844	0.9781	**0.9987**
D11	0.9790	**0.9978**	0.9950	0.9785	0.9876	0.9878
D12	0.9238	0.9096	0.9859	0.9964	0.9963	**0.9975**
D13	0.9465	0.9507	0.9802	0.9777	**0.9950**	0.9944
D14	0.9545	0.9626	0.9913	0.9905	0.9780	**0.9984**
D15	0.7730	0.8867	0.9566	0.8147	0.9700	**0.9704**
D16	0.6405	0.8861	0.9214	0.7488	0.9214	**0.9354**
D17	0.6457	0.6415	0.6526	0.7539	0.7419	**0.9430**
D18	0.7114	0.7641	0.7753	0.7976	0.9591	**0.9602**
D19	0.9390	0.9509	0.9825	0.9630	0.9762	**0.9865**
D20	0.9496	0.9681	0.9957	0.9636	0.9956	**0.9984**
D21	0.8816	0.8931	0.9821	0.9928	0.9969	**0.9970**
D22	0.8264	0.8846	0.9515	0.8743	0.9700	**0.9703**

(Continued)

Table 8 (continued)

id	SMOTE	SMOTE-ENN	SMOTE-WENN	PF-SMOTE	SMOTE-RkNN	RE-SMOTE
D23	0.9981	0.9982	0.9996	**0.9998**	0.9996	**0.9998**
D24	0.8009	0.8820	0.9511	0.7874	0.9498	0.7874
D25	0.9494	0.9533	0.9928	0.9769	0.9898	**1.0000**
D26	0.9840	0.9868	0.9953	0.9939	**0.9989**	**0.9982**
D27	0.9432	0.9458	0.9929	0.9622	0.9963	**0.9966**
D28	0.3251	0.1970	0.2255	**0.6672**	0.6660	0.2123
D29	0.9480	0.9526	0.9893	0.9886	0.9866	**0.9968**
D30	0.8185	0.9021	0.9469	0.8032	0.9469	**0.9562**
D31	0.7193	0.8607	0.9513	0.7978	0.8931	**0.9633**
D32	0.9801	0.9836	**0.9992**	0.9864	0.9885	0.9987
D33	0.9745	0.9738	0.9957	0.9090	0.9894	**0.9996**
D34	0.7292	0.8086	0.9201	0.8210	0.9201	**0.9579**
D35	0.9995	0.9993	0.9996	**0.9999**	0.9998	**0.9999**
D36	0.8733	0.8778	0.9820	0.7917	0.8919	**0.9995**
D37	0.7849	0.7702	0.9301	0.8376	0.9824	**0.9827**
D38	0.9409	0.9476	0.9882	0.9747	0.9850	**0.9952**
D39	0.9925	0.9871	**1.0000**	**1.0000**	0.9999	**1.0000**
D40	0.8901	0.8934	0.9849	0.9118	0.9734	**0.9996**

Table 9: Results of Wilcoxon signed-rank tests for comparing RE-SMOTE and the well-known variants of SMOTE when KNN is used as the classifier

Comparison	AUC			F1		
	R^+	R^-	p-value	R^+	R^-	p-value
RE-SMOTE *vs*. SMOTE	753	67	1.6666e–07	792	28	2.2937e–09
RE-SMOTE *vs*. SMOTE-ENN	809	11	1.2732e–09	809	11	9.0949e–12
RE-SMOTE *vs*. SMOTE-WENN	743	743	2.3753e–05	710	110	1.3279e–06
RE-SMOTE *vs*. PF-SMOTE	674	146	1.4131e–07	695	125	2.0881e–06
RE-SMOTE *vs*. SMOTE-RkNN	710	110	1.5234e–06	730	90	8.5432e–07

Table 10: Comparison results on AUC with SVM-based classifier (The best results in each dataset are shown in bold)

id	SMOTE	SMOTE-ENN	SMOTE-WENN	PF-SMOTE	SMOTE-RkNN	RE-SMOTE
D1	0.9680	0.9874	**0.9971**	0.9769	0.9951	0.9960
D2	0.9794	0.9889	0.9974	0.9879	0.9974	**1.0000**
D3	0.9610	0.9809	**0.9981**	0.9815	0.9980	0.9940
D4	0.9933	0.9934	0.9986	0.9998	0.9981	**1.0000**

(Continued)

Table 10 (continued)

id	SMOTE	SMOTE-ENN	SMOTE-WENN	PF-SMOTE	SMOTE-RkNN	RE-SMOTE
D5	0.9609	0.9702	0.9872	0.9977	0.9970	**1.0000**
D6	0.9251	0.9303	0.9395	0.9738	0.9394	**0.9857**
D7	0.8696	0.9142	**0.9378**	0.7619	0.9377	0.8029
D8	0.7305	0.8359	0.8928	0.8070	**0.9327**	0.9235
D9	0.8567	0.8600	0.8910	0.9016	0.8909	**0.9176**
D10	0.9855	0.9885	0.9952	0.9943	0.9950	**0.9957**
D11	0.9920	**0.9999**	0.9998	0.9904	0.9997	0.9996
D12	0.9303	0.9515	0.9721	**0.9872**	0.9720	0.9865
D13	0.8882	0.9068	0.9355	0.9794	0.9793	**0.9938**
D14	0.9021	0.9140	**0.9250**	0.9223	0.9249	0.9155
D15	0.8457	0.9229	**0.9496**	0.8558	0.9495	0.9413
D16	0.7155	0.8547	0.8521	0.7838	0.8520	**0.9118**
D17	0.7522	0.7656	0.7566	0.7121	0.7565	**0.9071**
D18	0.7204	0.7633	0.7607	0.8171	0.8170	**0.9455**
D19	0.9237	0.9345	0.9662	**0.9664**	0.9661	0.9610
D20	0.9789	0.9919	**0.9985**	0.9880	0.9984	0.9960
D21	0.8377	0.8117	0.9062	0.9905	**0.9961**	0.9901
D22	0.6103	0.6684	**0.7127**	0.6091	0.7126	0.6557
D23	0.9927	0.9925	0.9932	0.9996	0.9996	**0.9997**
D24	0.7803	0.8454	0.8544	0.8303	0.8543	**0.9164**
D25	0.9245	0.9188	0.9480	0.9825	0.9824	**1.0000**
D26	0.9940	0.9941	0.9969	0.9987	0.9968	**0.9990**
D27	0.9400	0.9480	**0.9823**	0.9599	0.9822	0.9691
D28	0.9918	0.9767	**1.0000**	0.9843	0.9999	**1.0000**
D29	0.9993	0.9996	**1.0000**	0.9993	1.0000	**1.0000**
D30	0.7701	0.8254	0.8227	0.8152	0.8226	**0.9158**
D31	0.7709	0.8828	0.9252	0.8662	0.9251	**0.9595**
D32	0.9850	0.9880	**0.9973**	0.9898	0.9972	0.9916
D33	0.9910	0.9910	**0.9989**	0.9438	0.9988	0.9987
D34	0.8137	0.8613	0.9031	0.8816	0.9030	**0.9516**
D35	0.9964	0.9963	0.9972	**0.9982**	0.9971	**0.9982**
D36	0.8142	0.8205	0.8947	0.8713	0.8946	**0.9991**
D37	0.8156	0.8131	0.9472	0.9219	0.9471	**0.9997**
D38	0.9912	0.9916	0.9932	0.9925	0.9931	**0.9999**
D39	**1.0000**	0.9996	**1.0000**	**1.0000**	1.0000	**1.0000**
D40	0.8342	0.8494	0.8892	0.8426	0.8891	**0.9900**

Table 11: Comparison results on F1 with SVM-based classifier (The best results in each dataset are shown in bold)

id	SMOTE	SMOTE-ENN	SMOTE-WENN	PF-SMOTE	SMOTE-RkNN	RE-SMOTE
D1	0.9076	0.9527	**0.9876**	0.9411	0.9875	0.9719
D2	0.9319	0.9460	0.9763	0.9744	0.9762	**0.9971**
D3	0.9358	0.9466	**0.9882**	0.9494	0.9881	0.9864
D4	0.9531	0.9561	0.9897	0.9884	0.9896	**0.9984**
D5	0.8529	0.8678	0.9541	0.9927	0.9540	**0.9963**
D6	0.4413	0.0000	**0.6827**	0.9738	0.6816	0.2823
D7	0.4068	0.6720	**0.6902**	0.1692	0.6901	0.2046
D8	0.6921	0.5501	0.8201	0.6662	0.8200	**0.8614**
D9	0.7676	0.7761	0.7993	0.8163	0.7992	**0.8362**
D10	0.9426	0.9461	0.9743	0.9690	0.9742	**0.9827**
D11	0.9785	**0.9959**	0.9935	0.9777	0.9934	0.9912
D12	0.7859	0.7554	**0.9036**	0.8714	0.9030	0.8692
D13	0.3273	0.0000	**0.6923**	0.0663	0.6921	0.5502
D14	0.7732	0.7826	**0.7912**	0.7863	0.7901	0.7464
D15	0.7679	0.8414	**0.8910**	0.6943	0.8872	0.8310
D16	0.4939	0.6465	0.6676	0.7154	0.6675	**0.7634**
D17	0.7004	0.7125	0.7147	0.7548	0.7146	**0.8444**
D18	0.7629	0.7818	0.7854	0.8253	0.7853	**0.8821**
D19	0.8238	0.8338	**0.8691**	0.8413	0.8670	0.8590
D20	0.9464	0.9548	**0.9855**	0.9475	0.9854	0.9663
D21	0.7154	0.6641	0.8444	0.8721	**0.8943**	0.8879
D22	0.5525	0.0000	**0.6762**	0.2491	0.6761	0.2435
D23	0.9791	0.9790	0.9829	**0.9831**	0.9828	0.9812
D24	0.7383	0.7793	**0.8233**	0.7001	0.8232	0.8322
D25	0.8278	0.8260	0.9026	0.9773	0.9025	**1.0000**
D26	0.9695	0.9707	0.9748	0.9879	0.9747	**0.9889**
D27	0.8783	0.8828	0.9515	0.8687	0.9514	**0.9571**
D28	0.9740	0.9575	**0.9947**	0.9537	0.9944	0.9946
D29	0.9566	0.9579	0.9747	0.9864	0.9746	**0.9952**
D30	0.7352	0.7867	0.8002	0.6858	0.7901	**0.8186**
D31	0.7224	0.8170	**0.8834**	0.7612	0.8833	0.8725
D32	0.9565	0.9571	**0.9902**	0.9668	**0.9911**	0.9789
D33	0.9499	0.9515	0.9843	0.8659	0.9832	**0.9966**
D34	0.7531	0.7696	0.8420	0.8079	0.8419	**0.8858**
D35	0.9215	0.9210	0.9494	**0.9989**	0.9493	**0.9989**
D36	0.7686	0.7619	0.8923	0.4679	0.8922	**0.9802**
D37	0.7465	0.7156	0.8818	0.8346	0.8817	**0.9573**
D38	0.9404	0.9379	0.9775	0.9762	0.9764	**0.9884**

(Continued)

Table 11 (continued)

id	SMOTE	SMOTE-ENN	SMOTE-WENN	PF-SMOTE	SMOTE-RkNN	RE-SMOTE
D39	0.9823	0.9835	0.9808	**1.0000**	0.9999	**1.0000**
D40	0.7541	0.7760	0.8658	0.2156	0.8657	**0.9402**

Table 12: Results of Wilcoxon signed-rank tests for comparing RE-SMOTE and the well-known variants of SMOTE when SVM is used as the classifier

Comparison	AUC			F1		
	R^+	R^-	p-Value	R^+	R^-	p-Value
RE-SMOTE *vs*. SMOTE	774	46	3.2639e–07	771	49	2.2706e–05
RE-SMOTE *vs*. SMOTE-ENN	780	40	6.4606e–08	773	47	1.0366e–07
RE-SMOTE *vs*. SMOTE-WENN	520	300	0.00386	563	257	0.16178
RE-SMOTE *vs*. PF-SMOTE	480	140	7.8917e–07	675	145	8.5000e–07
RE-SMOTE *vs*. SMOTE-RkNN	550	270	0.00098	600	220	0.0098

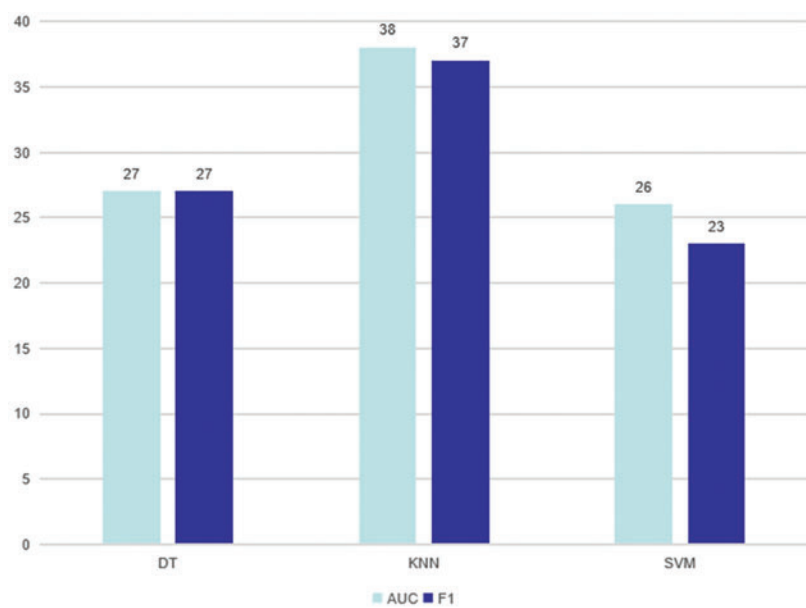


Figure 7: Best results of RE-SMOTE on 40 data sets

Table 3 presents the average results of comparative methods across 40 datasets, tested on three different classifiers. From the table, it is evident that our method consistently outperforms the comparative approaches across all classifiers, demonstrating its superior performance. This highlights the effectiveness and robustness of our approach to handling diverse datasets.

In addition to the above comparisons, "The Wilcoxon Signed Rank Test" is also used for statistical analysis of the proposed RE-SMOTE. The Wilcoxon Signed Rank Test (also known as Wilcoxon Signed Rank Sum Test) is non-parametric, and it is often used to determine the matching degree of the overall data distributions, especially for non-normal conditions. The statistical test results on AUC and F1 with DT-based, KNN-based, and SVM-based classifiers are given in Tables 6, 9, and 12, respectively.

R^+ and R^- represent the value of sign rank test for RE-SMOTE and compared models. The ratio of R^+ and R^- can be used to express the performance criteria, and it is expected for a big value. We can find from the result tables that R^+ value is greater than R^- for all cases.

For *p*-value, we have zero hypothesis: there is no difference between the performance of RE-SMOTE and other models. A smaller *p*-value means this zero hypothesis can be rejected. As can be seen from the results, the *p*-value is smaller for most cases (significantly less than alpha-value = 0.05). There is only one exceptional case in Table 13. *p*-value = 0.16178 for RE-SMOTE *vs.* SMOTE-WENN on F1. Therefore, for comprehensive consideration, there are significant differences between RE-SMOTE and other compared models.

Table 13: Runtime for processing 40 datasets

id	Datasets	#Attr.	#NE	%Class (maj,min)	#IR	Time (s)
D1	ecoli1	7	336	(259,77)	3.364	3.199
D2	ecoli2	7	336	(284,52)	5.462	3.763
D3	ecoli3	7	336	(301,15)	8.600	4.163
D4	ecoli4	7	336	(316,20)	15.800	4.723
D5	ecoli-$0_v s_1$	7	220	(143,77)	1.857	1.121
D6	glass-0-1-2-$3_v s_4-5-6$	9	214	(163,51)	3.196	1.362
D7	glass0	9	214	(144,70)	2.057	1.056
D8	haberman	3	306	(225,81)	2.778	2.630
D9	vehicle2	18	846	(628,218)	2.881	19.593
D10	yeast6	8	1484	(1449,35)	41.400	104.411
D11	wisconsin	9	683	(444,239)	1.858	10.396
D12	new-thyroid1	5	215	(180,35)	5.143	1.561
D13	glass6	9	214	(185,29)	6.379	1.755
D14	page-blocks0	10	5472	(4913,559)	8.789	1184.220
D15	yeast1	8	1484	(1055,429)	2.459	54.393
D16	australian	14	690	(383,307)	1.248	8.261
D17	bupa	6	345	(200,145)	1.379	2.890
D18	heart	13	270	(150,120)	1.250	2.425
D19	vehicle0	18	846	(649,199)	3.251	21.510
D20	yeast3	8	1484	(1321,163)	8.104	82.374
D21	new-thyroid2	5	215	(180,35)	5.143	1.510
D22	glass1	9	214	(138,76)	1.816	0.972
D23	vowel0	13	988	(898,90)	9.978	38.463
D24	vehicle3	18	846	(634,212)	2.991	19.700
D25	yeast-$2_v s_8$	8	482	(462,200)	23.100	9.961

(Continued)

Table 13 (continued)

id	Datasets	#Attr.	#NE	%Class (maj,min)	#IR	Time (s)
D26	segment0	19	2308	(1979,329)	6.015	192.016
D27	yeast4	8	1484	(1433,51)	28.098	95.635
D28	ring	20	740	(373,367)	1.016	17.524
D29	dermatology-6	34	358	(338,20)	16.900	4.283
D30	vehicle1	18	846	(629,217)	2.899	20.507
D31	pima	8	768	(500,268)	1.866	13.863
D32	yeast5	8	1484	(1440,44)	32.727	96.822
D33	poker-8$_{v}s_{6}s$	10	1477	(1460,17)	85.882	103.412
D34	magic	10	1902	(1234,668)	1.847	109.207
D35	shuttle-2$_{v}s_{5}$	9	3316	(3627,49)	66.673	497.811
D36	winequality-red-4	11	1599	(1546,53)	29.170	114.416
D37	hepatitis	19	80	(67,13)	5.154	0.725
D38	ecoli-0-6-7$_{v}s_{5}$	6	220	(200,20)	10.000	1.952
D39	shuttle-6$_{v}s_{2}-3$	9	230	(220,10)	22.000	2.184
D40	winequality-red-8$_{v}s_{6}-7$	11	855	(837,18)	46.500	32.105

5.3 Runtime and Complexity Analysis

In this section, the runtime for processing all 40 datasets has been recorded and presented in Table 13. Each dataset is processed using the methods described in this study, and the total time for each dataset is measured to provide a comprehensive overview of the computational performance. Based on the results, it can be observed that the processing time increases as the dataset size and imbalance ratio (IR) grow. Due to operations such as reverse k-nearest neighbor searches, the time consumption grows significantly as the dataset size increases.

Regarding the complexity, two core algorithms are analyzed. The time complexity of Algorithm 1 is $O(d\,|D^{+}|\,|D^{-}|)$, dominated by the nearest neighbor search between minority-class samples D^{+} and majority class samples D^{-}.

For Algorithm 2, the time complexity of Algorithm 2 is $O\left(d\,|S|^{2}\,k\right)$, primarily driven by the nearest and reverse nearest neighbor searched within the synthesized dataset S. The primary computational burden arises from search operations, which become increasingly intensive as the size of the synthesized dataset grows.

6 Conclusions

In this paper, we propose a novel hybrid resampling method RE-SMOTE to solve the class imbalance and diversity of synthetic samples. Different sample synthesis rules are adopted for the safe minority and the boundary minority class, and noisy samples are judged by WENN and the relative density for data cleaning. To demonstrate the effectiveness of RE-SMOTE, a variety of experiments on 40 datasets are tested. Different SMOTE variants equipment with different classifiers are adopted for evaluation. The experimental results demonstrate that the proposed RE-SMOTE significantly outperforms baseline methods.

The proposed method addresses the binary imbalance problem. In future work, we will focus on the more complex multivariate imbalance problem. Currently, minority classes are categorized into safe minority and boundary minority; we plan to explore personalized sample synthesis rules for various minority classes. Additionally, we will investigate further noise filters for data cleaning.

Acknowledgement: We thank all the members who have contributed to this work with us.

Funding Statement: This research is supported by the National Key R&D Program of China, No. 2022YFC3006302.

Author Contributions: Study conception and design: Dazhi E, Jiale Liu, Ming Zhang; Data collection: Dazhi E, Huiyuan Jiang, Keming Mao; Analysis and interpretation of methods: Dazhi E, Ming Zhang, Keming Mao; Draft manuscript preparation: Dazhi E, Huiyuan Jiang; Review and editing: Ming Zhang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The datasets and materials used in this study are available at https://github.com/blue9792/RE-SMOTE (accessed on 30 September 2024) and have been made publicly accessible for reproducibility and further research.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

[1] Z. Li, J. Tang, and T. Mei, "Deep collaborative embedding for social image understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2070–2083, 1 Sep. 2019. doi: 10.1109/TPAMI.2018.2852750.

[2] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li and Y. Gao, "A survey on federated learning," *Knowl.-Based Syst.*, vol. 216, 2021, Art. no. 106775. doi: 10.1016/j.knosys.2021.106775.

[3] B. Zhao, X. Zhang, H. Li, and Z. Yang, "Intelligent fault diagnosis of rolling bearings based on normalized CNN considering data imbalance and variable working conditions," *Knowl.-Based Syst.*, vol. 199, 2020, Art. no. 105971. doi: 10.1016/j.knosys.2020.105971.

[4] L. Wang, Y. Chen, H. Jiang, and J. Yao, "Imbalanced credit risk evaluation based on multiple sampling, multiple kernel fuzzy self-organizing map and local accuracy ensemble," *Appl. Soft Comput.*, vol. 91, no. 3, 2020, Art. no. 106262. doi: 10.1016/j.asoc.2020.106262.

[5] V. S. Sheng and C. X. Ling, "Thresholding for making classifiers cost-sensitive," in *Proc. 21st Natl. Conf. Artif. Intell.*, 2006, vol. 1, no. 3, pp. 476–481. doi: 10.1016/j.asoc.2020.106262.

[6] G. LemaÃŽtre, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *J. Mach. Learn. Res.*, vol. 18, no. 17, pp. 1–5, 2017.

[7] 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, 2002. doi: 10.1613/jair.953.

[8] S. -J. Yen and Y. -S. Lee, "Cluster-based under-sampling approaches for imbalanced data distributions," *Expert. Syst. Appl.*, vol. 36, no. 3, pp. 5718–5727, 2009. doi: 10.1016/j.eswa.2008.06.108.

[9] E. Ramentol, Y. Caballero, R. Bello, and F. Herrera, "SMOTE-RSB*: A hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using smote and rough sets theory," *Knowl. Inf. Syst.*, vol. 33, no. 2, pp. 245–265, 2012. doi: 10.1007/s10115-011-0465-6.

[10] Z. -J. Lee, C. -Y. Lee, S. -T. Chou, W. -P. Ma, F. Ye and Z. Chen, "A hybrid system for imbalanced data mining," *Microsyst. Technol.*, vol. 26, no. 9, pp. 3043–3047, 2020. doi: 10.1007/s00542-019-04566-1.

[11] D. Gyoten, M. Ohkubo, and Y. Nagata, "Imbalanced data classification procedure based on SMOTE," *Total Qual. Sci.*, vol. 5, no. 2, pp. 64–71, 2020. doi: 10.17929/tqs.5.64.

[12] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Special issue on learning from imbalanced data sets," *ACM SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 1–6, 2004. doi: 10.1145/1007730.1007733.

[13] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)*, vol. 42, no. 4, pp. 463–484, 2011. doi: 10.1109/TSMCC.2011.2161285.

[14] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Handling imbalanced datasets: A review," *GESTS Int. Trans. Comput. Sci. Eng.*, vol. 30, no. 1, pp. 25–36, 2006.

[15] X. -Y. Liu and Z. -H. Zhou, "The influence of class imbalance on cost-sensitive learning: An empirical study," in *Sixth Int. Conf. Data Min. (ICDM'06)*, Hong Kong, China, 2006, pp. 970–974. doi: 10.1109/ICDM.2006.158.

[16] M. A. Maloof, "Learning when data sets are imbalanced and when costs are unequal and unknown," in *Int. Conf. Mach. Learn.*, 2003.

[17] K. McCarthy, B. Zabar, and G. Weiss, "Does cost-sensitive learning beat sampling for classifying rare classes?" in *Proc. 1st Int. Workshop Util.-Based Data Min.*, 2005, pp. 69–77.

[18] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE," *Inform. Sci.*, vol. 465, no. 1, pp. 1–20, 2018. doi: 10.1016/j.ins.2018.06.056.

[19] Q. Chen, Z. -L. Zhang, W. -P. Huang, J. Wu, and X. -G. Luo, "PF-SMOTE: A novel parameter-free SMOTE for imbalanced datasets," *Neurocomputing*, vol. 498, no. 16, pp. 75–88, 2022. doi: 10.1016/j.neucom.2022.05.017.

[20] 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," *Appl. Intell.*, vol. 51, no. 3, pp. 1394–1409, 2021. doi: 10.1007/s10489-020-01852-8.

[21] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognit.*, vol. 30, no. 7, pp. 1145–1159, 1997. doi: 10.1016/S0031-3203(96)00142-2.

[22] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.

[23] T. F. Zhu, Y. P. Lin, and Y. H. Liu, "Synthetic minority oversampling technique for multiclass imbalance problems," *Pattern Recognit.*, vol. 72, no. 9, pp. 327–340, Dec. 2017. doi: 10.1016/j.patcog.2017.07.024.

[24] M. Dong, M. Liu, and C. Jing, "Sampling safety coefficient for multi-class imbalance oversampling algorithm,"(in Chinese), *J. Front. Comput. Sci. Technol.*, vol. 14, no. 10, pp. 1776–1786, 2020.

[25] H. A. Khorshidi and U. Aickelin, "A synthetic over-sampling method with minority and majority classes for imbalance problems," 2020, *arXiv:2011.04170*.

[26] M. Dong, Z. Jiang, and C. Jing, "Multi-class imbalanced learning algorithm based on Hellinger Distance and SMOTE algorithm," (in Chinese), *Comput. Sci.*, vol. 47, no. 1, pp. 102–109, 2020.

[27] D. A. Cieslak, T. R. Hoens, N. V. Chawla, and W. P. Kegelmeyer, "Hellinger distance decision trees are robust and skew-insensitive," *Data Min. Knowl. Discov.*, vol. 24, no. 1, pp. 136–158, Jan. 2012. doi: 10.1007/s10618-011-0222-1.

[28] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intell.)*, Hong Kong, China, 2008, pp. 1322–1328. doi: 10.1109/IJCNN.2008.4633969.

[29] H. Han, W. Y. Wang, and B. H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," *Adv. Intell. Comput.*, vol. 3644, pp. 878–887, 2005. doi: 10.1007/11538059.

[30] R. Alejo, V. García, and J. Pacheco-Sánchez, "An efficient over-sampling approach based on mean square error back-propagation for dealing with the multi-class imbalance problem," *Neural Process. Lett.*, vol. 42, no. 3, pp. 603–617, Dec. 2015. doi: 10.1007/s11063-014-9376-3.

[31] Y. E. Kurniawati, A. E. Permanasari, and S. Fauziati, "Adaptive synthetic-nominal (ADASYN-N) and adaptive synthetic-KNN (ADASYN-KNN) for multiclass imbalance learning on laboratory test data," in *2018 4th Int. Conf. Sci. Technol. (ICST)*, Yogyakarta, Indonesia, 2018, pp. 1–6. doi: 10.1109/ICSTC.2018.8528679.

[32] S. Rahayu, J. A. Putra, and Y. M.Z, "Effect of giving N value on ADA N method for classification of imbalanced nominal data," in *2019 4th Int. Conf. Inform. Technol. Inform. Syst. Elect. Eng. (ICITISEE)*, Yogyakarta, Indonesia, 2019, pp. 290–294. doi: 10.1109/ICITISEE48480.2019.9003757.

[33] A. Agrawal, H. L. Viktor, and E. Paquet, "SCUT: Multi-class imbalanced data classification using SMOTE and cluster-based undersampling," in *7th Int. Joint Conf. Knowl. Discov. Knowl. Eng. Knowl. Manage. (IC3K)*, Lisbon, Portugal, 2015, pp. 226–233.

[34] O. M. Olaitan and H. L. Viktor, "SCUT-DS: Learning from multi-class imbalanced Canadian weather data," in *24th Int. Symp. Methodol. Intell. Syst. (ISMIS)*, Limassol, Cyprus, 2018, vol. 11177, pp. 291–301. doi: 10.1007/978-3-030-01851-1_28.

[35] S. Feng *et al.*, "COSTE: Complexity-based OverSampling TEchnique to alleviate the class imbalance problem in software defect prediction," *Inf. Softw. Tech.*, vol. 129, Jan. 2021, Art. no. 106432. doi: 10.1016/j.infsof.2020.106432.

[36] S. Hartono, A. Lestari, A. Rahmadsyah, R. Maya Faza Lubis, and M. Gunawan, "HAR-MI with COSTE in handling multi-class imbalance," in *2020 8th Int. Conf. Cyber IT Serv. Manage. (CITSM)*, Pangkal, Indonesia, 2020, pp. 1–4. doi: 10.1109/CITSM50537.2020.9268804.

[37] S. W. Purnami and R. K. Trapsilasiwi, "SMOTE-least square support vector machine for classification of multiclass imbalanced data," in *Proc. 9th Int. Conf. Mach. Learn. Comput.*, 2017, pp. 107–111.

[38] N. U. Asniar, N. U. Maulidevi, and K. Surendro, "SMOTE-LOF for noise identification in imbalanced data classification," *J. King Saud Univ.-Comput. Inform. Sci.*, vol. 34, no. 6, pp. 3413–3423, Jun. 2022. doi: 10.1016/j.jksuci.2021.01.014.

[39] A. Arafa, N. El-Fishawy, M. Badawy, and M. Radad, "RN-SMOTE: Reduced noise SMOTE based on DBSCAN for enhancing imbalanced data classification," *J. King Saud Univ.-Comput. Inform. Sci.*, vol. 34, no. 8, pp. 5059–5074, Sep. 2022. doi: 10.1016/j.jksuci.2022.06.005.

[40] S. Maldonado, C. Vairetti, A. Fernandez, and F. Herrera, "FW-SMOTE: A feature-weighted oversampling approach for imbalanced classification," *Pattern Recogn.*, vol. 124, Apr. 2022, Art. no. 108511. doi: 10.1016/j.patcog.2021.108511.

[41] D. Dablain, B. Krawczyk, and N. V. Chawla, "DeepSMOTE: Fusing deep learning and SMOTE for imbalanced data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 9, pp. 6390–6404, Sep. 2023. doi: 10.1109/TNNLS.2021.3136503.

[42] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*, 1st ed. New York, USA: Chapman and Hall/CRC; 1984.

[43] S. L. Salzberg, "C4.5: Programs for machine learning by J. Ross Quinlan," in *Machine Learning*, 1st ed. Boston, MA, USA: Kluwer Academic Publishers, 1994, vol. 16, pp. 235–240.

[44] V. Vapnik, "The support vector method of function estimation," in *Int. Workshop Adv. Black-Box Tech. Nonlinear Model.-Theory Appl.*, Catholic University of Louvain, Louvain, Belgium, 1998, pp. 55–85.

[45] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, 1967. doi: 10.1109/TIT.1967.1053964.

[46] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 20–29, 2004. doi: 10.1145/1007730.1007735.

[47] 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," *Inform. Sci.*, vol. 595, no. 2, pp. 70–88, May 2022. doi: 10.1016/j.ins.2022.02.038.