IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Synthetic Minority Oversampling Technique Based on Adaptive Local Mean Vectors and Improved Differential Evolution

Junnan Li[1, 2]

[1]Chongqing Vocational Institute of Engineering, Big Data and Internet of Things School, Chongqing, 402260, China
[2]Chongqing University, Chongqing Key Laboratory of Software Theory & Technology, Chongqing, College of Computer Science, 400044, China

Corresponding author: Junnan Li (e-mail: JunnanLi@cqu.edu.cn).

ABSTRACT SMOTE is a classical oversampling method and aims to improve imbalanced classification by creating synthetic minority class samples. Overgeneralization is a great challenge in SMOTE and its improvements. Multiple variations of SMOTE are proposed against imbalances between classes and overgeneralization. However, they still have the following issues: a) most methods depend on too many parameters; b) most methods fail to detect suspicious noise effectively and modify them; c) interpolation of almost all methods is susceptible to abnormal samples. To overcome the above issues, a new synthetic minority oversampling technique based on adaptive local mean vectors and improved differential evolution (SMOTE-LMVDE) is proposed. First, a new noise detection technique based on the defined adaptive local mean vectors (NDALMV) is proposed to find suspicious noise. Second, an improved differential evolution is proposed to modify and improve detected suspicious noise. Finally, a new interpolation based on the defined adaptive local mean vectors is proposed to create synthetic minority class samples. Experiments prove that the proposed method superior to 7 popular oversampling approaches on extensive data sets in the training nearest neighbor classifier and the decision tree classifier.

INDEX TERMS Imbalanced learning; Imbalanced classification; Classification; Oversampling; Local means; Differential evolution

I. INTRODUCTION

Imbalanced classification has been favored by scholars in genetic engineering [1], text mining [2], image recognition [3], financial fraud [4], etc. In these practical applications, the number of negative cases is much more than that of positive cases due to the highly skewed class distribution. Negative and positive cases are regarded as the majority and minority classes, respectively. Under such circumstances, the minority class is more concerned, but it is easy to be misclassified due to the limited number.

Imbalanced classification [5] has been intensively studied and developed into cost-sensitive, algorithm-level and data-level approaches. In terms of the cost-sensitive approach [6], they generate the cost matrices by the imbalance ratio and misclassification costs. Then, the cost matrices are used for the imbalanced classification. The algorithm-level approach [7] usually modifies the theoretical model or cost function of the traditional classifiers. The algorithm-level approach aims to make the traditional classifiers adapt to imbalanced classification. The data-level approach [8] is the most dominant because of the wrapping advantage, i.e., it is independent of classifiers. Concretely, the data-level approach includes oversampling techniques [9, 10], undersampling techniques [11] and hybrid techniques [12].

Oversampling techniques improve the class distribution of data by creating synthetic minority class samples. By contrast, undersampling methods intend to remove redundant majority class samples. Hybrid techniques, such as S-SulfPred [12] and SSO_{Maj}-SMOTE-SSO_{Min} [13], are developed and combine oversampling techniques with undersampling techniques. Among oversampling techniques, the Synthetic Minority Over-sampling Technique (SMOTE) [5] is the most successful due to a lot of admiration and extensive practice, such as gender analysis [14], bioengineering [15], medical examination [16], Fraud identification [17].

Numerous experiments and studies [5, 8] show that overgeneralization is a great challenge in SMOTE and its improvements. Overgeneralization usually refers to noise generation in SMOTE-based methods [18, 19]. Synthetic minority class samples may become noise and cross the decision boundary due to interpolation among suspicious noisy samples and (or) improper values of parameters, leading to overgeneralization.

Multiple variations of SMOTE are proposed to handle imbalances between classes and overgeneralization. Representative examples are change-direction oversampling techniques and filtering-based oversampling techniques. Change-direction oversampling techniques overcome imbalances between classes and overgeneralization by creating synthetic minority class samples in high-density and (or) central regions. Safe-Level-SMOTE [20], ADASYN [21], DBSMOTE [22], MWMOTE [23], NI-MWMOTE [24], *k*-means SMOTE [25], Adaptive-SMOTE [26] and RSMOTE [27] belong to change-direction oversampling techniques. Filtering-based techniques deal with imbalances between classes and overgeneralization by employing noise detection approaches. Employed noise detection approaches can find and remove suspicious noise in filtering-based techniques. SMOTE-ENN [28], SMOTE-WENN [29], SMOTE-IPF [18], FRIPS-SMOTE [19] and SMOTE-NaN-DE [10] are with the idea of filtering-based techniques. Despite their effectiveness, they still have the following shortcomings:

(a) Most methods rely on too many parameters. ADASYN, DBSMOTE, Adaptive-SMOTE, SMOTE-ENN, SMOTE-WENN and FRIPS-SMOTE require 3 parameters. SMOTE-IPF, MWMOTE, NI-MWMOTE, *k*-means SMOTE and SMOTE-NaN-DE require 5 or more 5 parameters.

(b) Most methods fail to handle suspicious noise effectively. Although change-direction methods hardly use suspicious noise to generate synthetic samples, they fail to detect and (or) modify suspicious noise from the original and synthetic data. Filtering-based methods directly remove found suspicious noise rather than modifying or improving them, leading to information loss and distorting the real data distribution.

(c) Almost all methods use the *k* nearest neighbor-based interpolation to create synthetic minority class samples. As the study [5, 8] found, the *k* nearest neighbor-based interpolation heavily relies on parameter *k* and is susceptible to abnormal samples (e.g. outliers, noise or unsafe borderline samples). If one of the *k* nearest neighbors is the abnormal sample, the interpolation based on the selected abnormal will degrade.

To overcome the above issues of existing work while handling imbalances between classes and overgeneralization, a new synthetic minority oversampling technique based on adaptive local mean vectors and improved differential evolution (SMOTE-LMVDE) is proposed. First, a new noise detection technique based on the defined adaptive local mean vector (NDALMV) is proposed to find suspicious noise. Second, an improved differential evolution is proposed to modify and improve detected suspicious noise. Finally, a new interpolation based on the defined adaptive local mean vectors is proposed to create safer synthetic minority class samples. The main advantages of SMOTE-LMVDE are that a) it is parameter-free; b) it can modify found suspicious noisy samples rather than removing them; c) it can create safe synthetic minority class samples, avoiding overgeneralization. The chief contributions of this work are highlighted as follows:

- A new oversampling technique named SMOTE-LMVDE is proposed. It can eliminate imbalances between classes and avoid overgeneralization while overcoming the shortcoming of the existing work.
- A new concept, i.e., the parameter-free adaptive local mean vector is proposed. The defined adaptive local mean vectors help SMOTE-LMVDE detect suspicious noise and generate synthetic samples.
- A new noise detection technique (NDALMV) based on the defined adaptive local mean vector is proposed to find suspicious noise. Compared with existing noise detection techniques, NDALMV is parameter-free and reduces the bias towards the majority class.
- An improved differential evolution is proposed. Compared with related work [10, 30, 31], the proposed improved differential evolution is parameter-free and converges faster.
- A new interpolation based on the defined adaptive local mean vector is proposed to create synthetic minority class samples. The proposed interpolation is parameter-free and can reduce the error of synthetic minority class samples.
- Empirical results with 7 oversampling methods, the nearest neighbor classifier and the decision trees classifier on numerous data sets are reported.

The rest is organized as follows. Section II reviews related work and comparative methods in experiments. Section III shows preliminaries. Section IV introduces the proposed algorithm. Section V reports empirical results of intensive experiments and Section VI summarizes our work.

II. RELATED WORK

SMOTE was proposed by Chawla *et al*. [5]. Up to now, SMOTE has been favored in various practical applications due to its great value. Kamarulzalis *et al.* [14] apply SMOTE to gender analysis, in which J48 is used as the classifier. Liu *et al*. [15] apply SMOTE-TL to cancer risk prediction. Nakamura *et al*. [16] propose a novel SMOTE-based method using codebooks obtained by the learning vector quantization, and then apply the proposed SMOTE in biomedical data. Recently, BSMAIRS is proposed by Wang *et al*. [32]. BSMAIRS uses an oversampling method to improve the air algorithm, aiming to improve the classification of brain metastasis.

SMOTE is a wrapping algorithm that can train any supervised classifier in theory. SMOTEBoost [33] improves AdaBoost by employing SMOTE at each iteration of Adaboost. KSMOTE [34] improves the cost function of the support vector machine by combining SMOTE. In SMOTECSELM [35], ELM is modified by SMOTE, which improves ELM on imbalanced data.

Recent empirical studies [18, 19] indicate that overgeneralization is a great challenge in SMOTE and its improvements. SMOTE and its improvements may create synthetic minority class samples by the interpolation between suspicious noise or (and) harmful borderline samples. Hence, the generated synthetic minority class sample may also be noise and cross the decision boundary, resulting in overgeneralization. Additionally, inappropriate parameters of SMOTE-based methods also tend to increase the error of synthetic samples and the possibility of overgeneralization [8]. Among multiple variations of SMOTE, change-direction oversampling techniques and filtering-based oversampling techniques can overcome the imbalances between classes and overgeneralization at the same time.

Change-direction oversampling techniques employ heuristic models and statistical principles to create synthetic samples of the minority class in high-density and (or) central areas. Safe-Level-SMOTE is a classical change-direction oversampling technique and proposed by Bunkhumpornpat *et al.* [20]. A so-called safe level ratio is defined by a distance-based rule and the parameter k in Safe-Level-SMOTE. Then, Safe-Level-SMOTE uses the safe level ratio to create safe synthetic samples and compute the random differences between synthetic samples and base samples. ADASYN [21] is an improvement of Safe-Level-SMOTE. ADASYN employs k nearest neighbors to calculate the adaptive weight for each minority class sample. Samples that are hard to learn have higher adaptive weights. Then, more synthetic samples are created based on samples with higher adaptive weights. DBSMOTE [22], MWMOTE [23], NI-MWMOTE [24] and k-means SMOTE [25] are clustering-based change-direction oversampling techniques. DBSMOTE proposes a density-reachable graph by DBSCAN. Then, the shortest path algorithm is used to find the paths between cores points and minority class samples. Next, DBSMOTE generates synthetic samples by employing found paths. MWMOTE and NI-MWMOTE execute the agglomerative hierarchical clustering on minority class samples. Then, sampling weights based on the density factor and closeness factor are used to create synthetic samples and improve the minority class. k-means SMOTE performs k-means clustering on imbalanced data. Then, synthetic samples are generated based on the density of the filtered sub-cluster. Additionally, Adaptive-SMOTE [26] and RSMOTE [27] are the latest variants of change-direction oversampling techniques. Adaptive-SMOTE designs inner subsets and danger subsets by counting the neighbor's number in the majority and minority classes. Adaptive-SMOTE strengthens the distribution of the original data by using inner and danger subsets to create synthetic samples. RSMOTE employs homogeneous and heterogeneous k-nearest neighbors to compute density for each sample. Then, k-means clustering is used to partition the minority class into safe and borderline areas according to the density. Next, RSMOTE performs SMOTE in safe areas. Nevertheless, ADASYN, DBSMOTE and Adaptive-SMOTE depend on 3 parameters. MWMOTE, NI-MWMOTE, k-means SMOTE require 5, 6 and 9 parameters, respectively. Besides, the above methods fail to detect and (or) modify suspicious noise from the original and synthetic data.

Filtering-based oversampling techniques design noise filters, intending to detect and filter out suspicious noise. SMOTE-ENN [28], SMOTE-WENN [29], SMOTE-IPF [18], FRIPS-SMOTE [19] and SMOTE-NaN-DE [10] are competitive instances with the filtering-based idea. The edited nearest neighbor is employed in SMOTE-ENN and SMOTE-WENN to find mislabeled samples regarded as suspicious noise, in which SMOTE is executed to create synthetic samples. An ensemble classifier by bagging decision trees is employed in SMOTE-IPF to detect noise. SMOTE-IPF executes the noise filter based on the ensemble classifier k times. FRIPS-SMOTE calculates the membership degree of noise for each sample by statistics rough sets. After removing noise with a high membership degree of noise, SMOTE is performed in FRIPS-SMOTE. SMOTE-NaN-DE uses evolutionary algorithms to deal with noise in SMOTE. SMOTE-ENN, SMOTE-WENN and FRIPS-SMOTE rely on 3 parameters. SMOTE-IPF and SMOTE-NaN-DE depend on 5 or more 5 parameters. Also, most of them directly remove found suspicious noise rather than modifying or improving them.

In summary, change-direction and filtering-based oversampling techniques manage to combat imbalances between classes and overgeneralization, but they still have the following shortcomings: a) most methods require too many parameters; b) most methods fail to detect suspicious noise effectively and improve them; c) the k nearest neighbor-based interpolation employed in most methods heavily relies on the parameter k and is susceptible to abnormal samples (e.g. outliers, noise or unsafe borderline samples) [5, 8]. This paper proposes a new synthetic minority oversampling technique based on adaptive local mean vectors and improved differential evolution (SMOTE-LMVDE), aiming to overcome the above issues at the same time.

III. PRELIMINARIES

The Natural Neighbor (NaN) and Natural Neighbor Eigenvalue [36] are introduced in this section, which provides a theoretical basis for SMOTE-LMVDE.

A. NATURAL NEIGHBORS

The Natural Neighbor (NaN) [36] is a new technique of neighbors with a Natural Neighbor Eigenvalue (NaNE). The idea of the NaN comes from the understanding of the community in the real world. If two peoples are true friends, they should treat each other as a friend in a community. When everyone has a friend, a harmonious society will be

formed. For data objects, if two samples treat each other as a neighbor, they will be friends. When every sample has at least one friend, a Natural Stable Structure (NSS) will be formed in data objects. The relationship of neighbors formed in the NSS is called natural neighbors. The NSS is described in formula (1):

$$(\forall \boldsymbol{x}_i)(\exists \boldsymbol{x}_j)(\boldsymbol{x}_i \neq \boldsymbol{x}_j) \rightarrow (\boldsymbol{x}_i \in \boldsymbol{NN}_r(\boldsymbol{x}_j)) \wedge (\boldsymbol{x}_j \in \boldsymbol{NN}_r(\boldsymbol{x}_i)) \tag{1}$$

In formula (1), r is the search round and increased from 1 to λ, where λ is the Natural Neighbor Eigenvalue (NaNE). In other words, when $r=\lambda$, each sample has a friend and the NSS is formed in a given data set. The NaNE is defined in Definition 1.

Definition 1. (Natural Neighbor Eigenvalue): The Natural Neighbor Eigenvalue λ is equal to the search round r, when the Natural Neighbor Stable Structure is formed.

$$\lambda = r_{r\in n}\{r \mid (\forall \boldsymbol{x}_i)(\exists \boldsymbol{x}_j)(\boldsymbol{x}_i \neq \boldsymbol{x}_j) \rightarrow (\boldsymbol{x}_i \in \boldsymbol{NN}_r(\boldsymbol{x}_j)) \wedge (\boldsymbol{x}_j \in \boldsymbol{NN}_r(\boldsymbol{x}_i))\} \tag{2}$$

Based on Definition 1, NaN is defined as follows:

Definition 2. (Natural Neighbor): If sample $\boldsymbol{x}_j$ is a natural neighbor (NaN) of sample $\boldsymbol{x}_i$, sample $\boldsymbol{x}_j$ is one of λ nearest neighbors of sample $\boldsymbol{x}_i$ and sample $\boldsymbol{x}_i$ is one of λ nearest neighbors of sample $\boldsymbol{x}_j$.

$$\boldsymbol{x}_j \in \boldsymbol{NaN}(\boldsymbol{x}_i) \Leftrightarrow \boldsymbol{x}_i \in \boldsymbol{NN}_\lambda(\boldsymbol{x}_j) \ \&\& \ \boldsymbol{x}_j \in \boldsymbol{NN}_\lambda(\boldsymbol{x}_i) \tag{3}$$

Algorithm 1: Search for the NaN (*NaN_Search*)

Input: $\boldsymbol{X}$ (Input Data)
Output: λ (NaNE)

1: r=1, num_r=0, $\forall \boldsymbol{x}_i \in \boldsymbol{X}$, $\boldsymbol{Nb}(\boldsymbol{x}_i)$=0, $\boldsymbol{NN}_r(\boldsymbol{x}_i)=\varnothing$, $\boldsymbol{RNN}(\boldsymbol{x}_i)=\varnothing$, $\boldsymbol{NaN}(\boldsymbol{x}_i)=\varnothing$;
2: Create a *kd* tree from data set $\boldsymbol{X}$;
3: **while** $num_r \neq num_{r-1}$ && r>1
4: **for** each sample $\boldsymbol{x}_i$ in $\boldsymbol{X}$, finding its r-th neighbor $\boldsymbol{x}_j$ by using the created *kd* tree
5: $\boldsymbol{NN}_r(\boldsymbol{x}_i)=\boldsymbol{NN}_r(\boldsymbol{x}_i) \cup \{\boldsymbol{x}_j\}$;
6: $\boldsymbol{Nb}(\boldsymbol{x}_j)=\boldsymbol{Nb}(\boldsymbol{x}_j)+1$;
7: $\boldsymbol{RNN}(\boldsymbol{x}_j)=\boldsymbol{RNN}(\boldsymbol{x}_j) \cup \{\boldsymbol{x}_i\}$;
8: **end for**
9: Compute num_r; % num_r is the number of sample $\boldsymbol{x}_i$ with $\boldsymbol{Nb}(\boldsymbol{x}_i)$==0
10 $r=r+1$;
11: **end while**
12: $\lambda=r$-1;
13: **return** λ;

The searching algorithm for NaNs and NaNE is described in Algorithm 1 which returns λ. At Lines 2-8, the r-neighbor of each sample is searched until the NSS is formed. The stopping criteria of Algorithm 1 are that (1) every sample is considered as a neighbor; (2) the number of samples that are not considered as neighbors no longer changes since noise (i.e., outliers) can affect Algorithm 1. At Line 9, the value of *num* is calculated and *num* is the number of sample $\boldsymbol{x}_i$ with $\boldsymbol{Nb}(\boldsymbol{x}_i)$==0. $\boldsymbol{Nb}(\boldsymbol{x}_i)$ is the number of sample $\boldsymbol{x}_i$ that is considered as the neighbor of other samples. Hence, when *num* does not change at Line 3, the NSS is formed and the iteration stops. After NSS is formed, λ is calculated at Lines 12-13. In general, the time complexity is $O(N\log N)$ because *kd* tree [37] at Line 2 is employed to search for neighbors. N is the number of samples in $\boldsymbol{X}$. For more details on NaNs, please refer to the work [36]. Note that the Natural Neighbor Eigenvalue λ can be used to overcome the choice of parameter k [36]. Hence, we design an adaptive local mean vector based on the Natural Neighbor Eigenvalue λ in Section IV.A.

IV. PROPOSED ALGORITHM

$\boldsymbol{X}_{imb}=\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$ is an imbalanced training set with $\boldsymbol{X}_{min}$ and $\boldsymbol{X}_{maj}$. N is the sample number in $\boldsymbol{X}_{imb}$. $\boldsymbol{x}_i=\{\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, ..., \boldsymbol{x}_{i,D}\}$ is the ith sample in $\boldsymbol{X}_{imb}$ with D attributes. ω_i is the class label of sample $\boldsymbol{x}_i$. $\omega_i \in \{\omega_{min}, \omega_{maj}\}$. ω_{min} and ω_{maj} are the class label of minority and majority classes, respectively. $\boldsymbol{X}_{min}=\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{N_{min}}\}$ is the set of minority class samples. N_{min} is the number of minority class samples. $\boldsymbol{X}_{maj}=\{\boldsymbol{x}_{N_{min}+1}, \boldsymbol{x}_{N_{min}+2}, ..., \boldsymbol{x}_N\}$ is the set of majority class samples. N_{maj} is the number of majority class samples.

The pseudo-code of Algorithm 2 and Fig. 1 provide an overview of SMOTE-LMVDE. First, the Natural Neighbor Eigenvalue λ is computed by Algorithm 1 at Line 2. Second, a noise detection technique based on adaptive local mean vectors (NDALMV) is proposed to detect suspicious noise, as shown in Fig. 1 (a) and Line 3 of Algorithm 2. Third, an improved differential evolution is proposed to modify and optimize detected suspicious noise, as shown in Fig. 1 (b) and Lines 4-11 of Algorithm 2. Finally, a new interpolation based on the defined adaptive local mean vectors is proposed to create synthetic minority class samples, as shown in Fig. 1 (c) and Lines 12-22 of Algorithm 2.

Algorithm 2: *SMOTE-LMVDE*

Input: $\boldsymbol{X}_{min}$, $\boldsymbol{X}_{maj}$
Output: ***SyntheticSamples*** (The set of synthetic minority class samples)

		Time complexity
1	$\boldsymbol{X}_{imb}=\boldsymbol{X}_{min}\cup\boldsymbol{X}_{maj}$;	$O(1)$
	% First, computing the Natural Neighbor Eigenvalue λ	$O(NlogN)$
2	$\lambda=NaN\_Search(\boldsymbol{X}_{imb})$;	
	% Second, detecting noise by the proposed noise detection technique based on adaptive local mean vectors	
3	[***Noise***, ***Normal***]=$NDALMV(\boldsymbol{X}_{min}, \boldsymbol{X}_{maj}, \lambda)$;	$O(N)$
	% Third, modifying and improving found suspicious noise by the proposed improve differential evolution	
4	***OptimizedSample***=$ImprovedDiffierentialEvolution$(***Noise***, ***Normal***, λ);	$O(G_{max}\times NlogN)$
	% Using ***OptimizedSample*** to update $\boldsymbol{X}_{min}$ and $\boldsymbol{X}_{maj}$	
5	for each $\boldsymbol{x}_i\in$ ***OptimizedSample***	$O(N_{noise})$
6	if $\boldsymbol{x}_i\in\boldsymbol{X}_{min}$	$O(N_{noise})$
7	$\boldsymbol{X}_{min}=\boldsymbol{X}_{min}\cup\{\boldsymbol{x}_i\}$;	$O(N_{noise})$
8	else	
9	$\boldsymbol{X}_{maj}=\boldsymbol{X}_{maj}\cup\{\boldsymbol{x}_i\}$;	$O(N_{noise})$
10	end	
11	end	
	% Four, creating synthetic minority class samples by the proposed interpolation based on adaptive local mean vectors	
12	for each $\boldsymbol{x}_i\in\boldsymbol{X}_{min}$, computing the adaptive local mean vector $u(\boldsymbol{x}_i, \omega_{min})$ by $\boldsymbol{NN}_\lambda(\boldsymbol{x}_i)$ in $\boldsymbol{X}_{min}$;	$O(N_{min}logN_{min})$
13	for each $\boldsymbol{x}_i\in\boldsymbol{X}_{min}$	$O(N_{min})$
14	$Num=\left\lvert(N_{maj}-N_{min})/N_{min}\right\rvert$;	$O(N_{min}\times Num)$
15	***Base***=$\boldsymbol{x}_i$; % $\boldsymbol{x}_i$ is regarded as the base sample	$O(N_{min}\times Num)$
16	while $Num>0$	$O(N_{min}\times Num)$
17	for d=1: D	$O(N_{min}\times Num)$
18	Using formula (8) to create synthetic minority class sample ***New***;	$O(N_{min}\times Num)$
19	end for	
20	***SyntheticSamples***=***SyntheticSamples*** $\cup$ ***New***, $Num=Num$-1;	$O(N_{min}\times Num)$
21	end while	
22	end for	
23	return ***SyntheticSamples***;	$O(1)$

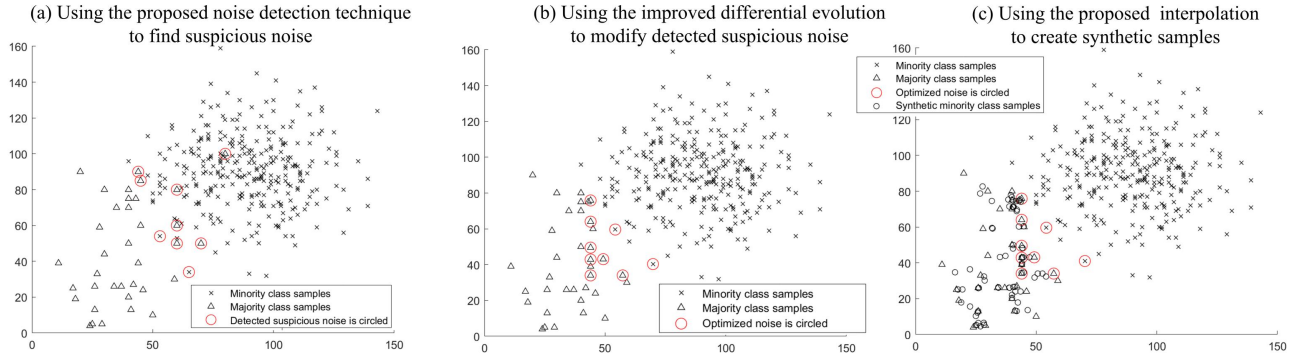

FIGURE 1. **Visualizing the main process of SMOTE-LMVDE on synthetic data.**

In the following, Section IV.A introduces the proposed noise detection technique based on adaptive local mean vectors. Section IV.B introduces the proposed improved differential evolution. Section IV.C introduces the proposed interpolation based on adaptive local mean vectors. The time complexity and characteristics of SMOTE-LMVDE are analyzed in Section IV.D.

A. NOISE DETECTION TECHNIQUE BASED ON ADAPTIVE LOCAL MEAN VECTORS

The proposed noise detection technique (NDALMV) is based on the defined adaptive local mean vector. The defined adaptive local mean vector is inspired by the Natural Neighbor Eigenvalue λ [36] (Algorithm 1), the local mean vector and the k nearest neighbors. The adaptive local mean vector is defined as follows:

Definition 3. (Adaptive Local Mean Vector): The adaptive local mean vectors of sample $\boldsymbol{x}_i$ are the local mean vectors from different classes in λ nearest neighbors $\boldsymbol{NN}_\lambda(\boldsymbol{x}_i)$. For each class ω_j, the adaptive local mean vector of sample $\boldsymbol{x}_i$ is formulated as follows:

$$\boldsymbol{u}(\boldsymbol{x}_i,\omega_j)=\frac{\sum_{\boldsymbol{x}_t\in\boldsymbol{NN}_\lambda(\boldsymbol{x}_i)\&\&\omega_t==\omega_j}\boldsymbol{x}_t}{\left\lvert\{\boldsymbol{x}_t\mid\boldsymbol{x}_t\in\boldsymbol{NN}_\lambda(\boldsymbol{x}_i)\&\&\omega_t==\omega_j\}\right\rvert},\ \omega_j\in\{\omega_{min},\omega_{maj}\} \tag{4}$$

|.| refers to the number. $|\{\boldsymbol{x}_t|\boldsymbol{x}_t \in \boldsymbol{NN}_\lambda(\boldsymbol{x}_i)\ \&\&\ \omega_t == \omega_j\}|$ is the sample's number of $\{\boldsymbol{x}_t|\boldsymbol{x}_t \in \boldsymbol{NN}_\lambda(\boldsymbol{x}_i)\ \&\&\ \omega_t == \omega_j\}$. ω_t is the class label of sample $\boldsymbol{x}_t$. $\boldsymbol{u}(\boldsymbol{x}_i, \omega_j)$ is the adaptive local mean vector of $\boldsymbol{x}_i$ in class ω_j. Next, the proposed NDALMV uses Definition 4 to detect suspicious noise.

Definition 4. (Noise): The set of noise is denoted as ***Noise***. If sample $\boldsymbol{x}_i$ belongs to ***Noise***, sample $\boldsymbol{x}_i$ has a different class label from the nearest adaptive local mean vector $\boldsymbol{u}(\boldsymbol{x}_i, \omega_j)$, where $\omega_j \in \{\omega_{min}, \omega_{maj}\}$.

$$\boldsymbol{x}_i \in \boldsymbol{Noise} \rightarrow \omega_i \neq \underset{\omega_j \in \{\omega_{min}, \omega_{max}\}}{arg\,min}\ (dist(\boldsymbol{x}_i, \boldsymbol{u}(\boldsymbol{x}_i, \omega_j))) \quad (5)$$

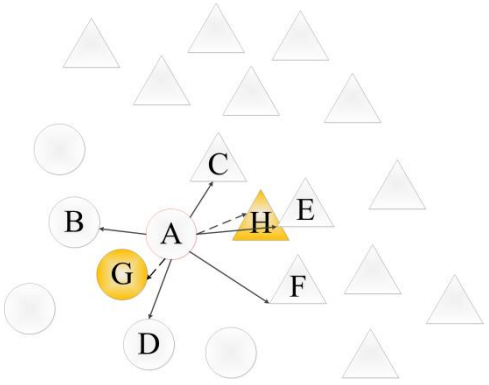

FIGURE 2. Determining noise by the proposed NDALMV and assuming λ=5 on synthetic data.

The function *dist*() returns the Euclidean distance between two samples and the function *argmin* returns the class label ω_j corresponding to the minimum value. Fig. 2 uses synthetic data to visualize the proposed NDALMV based on Definition 4. In Fig. 2, circles and triangles represent samples of minority and majority classes, respectively. $\boldsymbol{NN}_\lambda$(A)={B, C, D, E, F}. By employing Definition 3, Sample G and sample H are adaptive local mean vectors of sample A for minority and majority classes, respectively. Specifically, sample G is the adaptive local mean vector based on minority class samples B and D. Sample H is the adaptive local mean vector based on majority class samples C, E and F. By employing Definition 4, sample A is not noise because it is closer to sample G that has the same class label as sample A.

Most existing work [5, 8, 27] uses k nearest neighbors to determine noise with the majority voting. Take ENN [28] as an example. If ENN with k=5 is adopted in Fig. 2, sample A will be misjudged as a noisy sample because the majority class receives more votes than the minority class. Compared to existing work [5, 8, 27] with the majority voting, the proposed NDALMV is parameter-free by employing λ. Besides, it reduces the bias towards the majority class because there is only one local mean vector for a given sample in the majority class or the minority class.

The pseudo-code of the proposed NDALMV is described in Algorithm 3. At Lines 2-3, adaptive local mean vectors for each sample are calculated. After that, noise is determined by formula (5) at Lines 4-9. Please note that in Algorithm 3, several points need to be highlighted.

(a) As the analysis of column "Time complexity" in Algorithm 3, the time complexity of Algorithm 3 is $O(N)$.

(b) Compared to existing noise detection techniques, the proposed NDALMV in SMOTE-LMVDE is parameter-free due to the Natural Neighbor Eigenvalue λ.

(c) Most existing noise detection techniques are based on k nearest neighbors with the majority voting. Hence, they are biased towards the majority class because of $|\boldsymbol{X}_{maj}|>|\boldsymbol{X}_{min}|$. The proposed NDALMV in SMOTE-LMVDE can reduce bias towards the majority class by employing the adaptive local mean vector, since $|\boldsymbol{u}(\boldsymbol{x}_i, \omega_{min})|==|\boldsymbol{u}(\boldsymbol{x}_i, \omega_{maj})|$ for the sample $\boldsymbol{x}_i$ to be tested.

Algorithm 3: Noise detection based on adaptive local mean vectors (*NDALMV*)

Input: $\boldsymbol{X}_{min}$ (The set of minority class samples), $\boldsymbol{X}_{maj}$ (The set of majority class samples), λ (Natural neighbor eigenvalue) — Time complexity

Output: ***Noise*** (The set of noise), ***Normal*** (The set of normal samples)

		Time complexity
1	$\boldsymbol{X}_{imb}=\boldsymbol{X}_{min} \cup \boldsymbol{X}_{maj}$, $\boldsymbol{Noise}=\varnothing$, $\boldsymbol{Normal}=\varnothing$;	$O(1)$
2	for $\boldsymbol{x}_i \in \boldsymbol{X}_{imb}$;	$O(N)$
3	Using formula (4) to calculate $\boldsymbol{u}(\boldsymbol{x}_i, \omega_{maj})$ and $\boldsymbol{u}(\boldsymbol{x}_i, \omega_{min})$;	$O(N)$
4	if $\omega_i \neq argmin(dist(\boldsymbol{x}_i \boldsymbol{u}(\boldsymbol{x}_i, \omega_j)))$, $\omega_j \in \{\omega_{maj}, \omega_{min}\}$ % formula (5) and Defintion 4	$O(N)$
5	$\boldsymbol{Noise}=\boldsymbol{Noise} \cup \{\boldsymbol{x}_i\}$;	$O(N)$
6	else	$O(N)$
7	$\boldsymbol{Normal}=\boldsymbol{Normal} \cup \{\boldsymbol{x}_i\}$;	
8	end	$O(N)$
9	end	
10	return ***Noise***, ***Normal***;	$O(1)$

B. IMPROVED DIFFERENTIAL EVOLUTION

As analyzed in previous sections, most change-direction and filtering-based oversampling techniques fail to improve and modify detected suspicious noise. The differential evolution [30] is a numerical optimization algorithm and can optimize the attributes of given samples. However, existing differential evolution algorithms [10, 30, 31] rely on parameters. Besides, most of them optimize all vectors at each iteration, which increases unnecessary time consumption and leads to slow convergence. Hence, an improved differential evolution is proposed to improve and modify detected suspicious in SMOTE-LMVDE. The chief ideas of the proposed improved differential evolution are that a) suspicious noise is optimized by the random difference between it and one of its λ nearest neighbors with

the same class; b) if a suspicious noise is correctly classified by its nearest neighbor from normal samples in the optimization process, it will not be optimized at the next iteration; c) when all suspicious noise is correctly classified by its nearest neighbor from normal samples, the iteration stops. The improved differential evolution contains the initialization step, the mutation step and the selection step.

In the initialization step, each suspicious noisy sample $\boldsymbol{x}_i \in \boldsymbol{Noise}$ (the set of noise found by Algorithm 3) is regarded as an optimized vector $\boldsymbol{v}_i$.

$$\boldsymbol{V}_g = \{\boldsymbol{v}_{1,g}, \boldsymbol{v}_{1,g}, ...\boldsymbol{v}_{N_{noise},g}\} \tag{6}$$

$\boldsymbol{V}_g$ is the set of optimized vectors. $\boldsymbol{v}_{i,g}$ is the ith optimized vector at gth iteration, where $i \in \{1, 2, ..., N_{noise}\}$ and $g \in \{1, 2, ..., G_{max}\}$. When g=1, $\boldsymbol{v}_{i,g}$=$\boldsymbol{x}_i$ (i=1, 2, ..., N_{noise}). N_{noise} is the number of detected suspicious noisy samples and G_{max} is the maximum number of iterations.

In the mutation step, $\boldsymbol{v}_{i,g}$ is optimized by formula (7).

$$\boldsymbol{v}_{i,g}[d] = \boldsymbol{v}_{i,g}[d] + rand(0,1) \times (\boldsymbol{v}_{i,g}[d] - \boldsymbol{x}_r[d]) \tag{7}$$

$\boldsymbol{v}_{i,g}[d]$ and $\boldsymbol{x}_r[d]$ is the dth attribute of $\boldsymbol{v}_{i,g}$ and $\boldsymbol{x}_r$, where $d \in \{1, 2, ..., D\}$. $\boldsymbol{x}_r$ is one of $\boldsymbol{NN}_\lambda(\boldsymbol{v}_{i,g})$. $\boldsymbol{NN}_\lambda(\boldsymbol{v}_{i,g})$ is the λ nearest neighbors searched on $\{\boldsymbol{x}_j | \boldsymbol{x}_j \in \boldsymbol{Normal}$ && $\omega_j == \omega_i\}$, where ω_j or ω_i is the class label of $\boldsymbol{x}_j$ or $\boldsymbol{v}_{i,g}$. ***Normal*** is the set of normal samples found by Algorithm 3. The function *rand*(0, 1) returns a random number between 0 and 1.

The selection step is implied at Lines 12-14 of Algorithm 4. If the vector $\boldsymbol{v}_{i,g}$ is classified correctly by its nearest neighbor from ***normal***, then $\boldsymbol{v}_{i,g}$ will not be optimized at the next iteration and $OptmizedTag(\boldsymbol{v}_{i,g})$=*True*. Otherwise, $OptmizedTag(\boldsymbol{v}_{i,g})$=*False*.

The pseudo-code of the proposed improved differential evolution is described in Algorithm 4. Lines 1-2 is the initialization step, where each suspicious noisy sample $\boldsymbol{x}_i \in \boldsymbol{Noise}$ is regarded as an optimized vector $\boldsymbol{v}_i$ and its *OptimizedTag* is equal to *False*. Lines 3-11 is the mutation step, where each optimized vector $\boldsymbol{v}_i$ is improved and modified by the random difference between it and one of its λ nearest neighbors with the same class. Lines 12-14 are the selection Step. If an optimized vector is classified correctly by its nearest neighbor from ***Normal***, then its *OptimizedTag* is equal to *True* and will not be optimized at the next iteration. When all optimized vectors are classified correctly by their nearest neighbor from ***Normal***, then the iteration (Lines 4-17) stop. After that, Algorithm 4 outputs the set of optimized noisy samples. Please note that in Algorithm 4, several points need to be highlighted.

(a) Let the number of suspicious noisy samples and normal samples be denoted as N_{noise} and N_{normal}, respectively. As analyzed by Algorithm 3, the time complexity of the improved differential evolution is $O(G_{max} \times N_{noise} \times N_{normal} log N_{normal})$. Because $N_{normal} \approx N$ and $N_{noise} << N$ in most cases, the time complexity of the improved differential evolution is $O(G_{max} \times N log N)$.

(b) Compared to existing variations of the differential evolution [10, 30, 31], the improved differential evolution is parameter-free.

(c) Compared to existing variations of the differential evolution [10, 30, 31], the improved differential evolution can converge faster. The improved differential evolution only optimizes suspicious noise misclassified by its nearest neighbor from ***Normal*** (instead of all suspicious noise) at each iteration, which save time.

Algorithm 4: Improved differential evolution (*ImprovedDiffierentialEvolution*)

Input: ***Noise*** (The set of noisy samples), ***Normal*** (The set of normal samples), λ (Natural neighbor eigenvalue)
Output: ***OptimizedSample*** (The set of optimized noisy samples)

		Time complexity
	% Intialization Step	
1	$\boldsymbol{V}_g = \{\boldsymbol{v}_{1,g}, \boldsymbol{v}_{2,g}, ...\boldsymbol{v}_{N_{noise},g}\}$ is formed by ***Noise***;	$O(N_{noise})$
2	$\forall \boldsymbol{v}_{i,g} \in \boldsymbol{V}_g, OptimizedTag(\boldsymbol{v}_{i,g}) == False$;	$O(N_{noise})$
	% Mutation Step	$O(N_{noise})$
3	g=1;	
4	while $OptimizedTag(\exists \boldsymbol{v}_{i,g}) == False$	$O(G_{max})$
5	for $\boldsymbol{v}_{i,g} \in \boldsymbol{V}_g$	$O(G_{max} \times N_{noise})$
6	if $OptimizedTag(\boldsymbol{v}_{i,g}) == False$	$O(G_{max} \times N_{noise})$
7	$\boldsymbol{x}_r$ is a random sample of $\boldsymbol{NN}_\lambda(\boldsymbol{v}_{i,g})$ in $\{\boldsymbol{x}_j | \boldsymbol{x}_j \in \boldsymbol{Normal}$ && $\omega_j == \omega_i\}$;	$O(G_{max} \times N_{noise} \times N_{normal} log N_{normal})$
8	for d=1 to D	$O(G_{max} \times N_{noise})$
9	$\boldsymbol{v}_{i,g}[d] = \boldsymbol{v}_{i,g}[d] + rand(0,1) \times (\boldsymbol{v}_{i,g}[d] - \boldsymbol{x}_r[d])$;	$O(G_{max} \times N_{noise})$
10	end	
11	end	
	% Selection Step	
12	if $\boldsymbol{v}_{i,g}$ is classified correctly by its nearest neighbor from ***Normal***	$O(G_{max} \times N_{noise} \times N_{normal} log N_{normal})$
13	$OptimizedTag(\boldsymbol{v}_{i,g}) = True$;	$O(G_{max} \times N_{noise})$
14	end	
15	end	
16	g=g+1;	
17	end	
18	$\boldsymbol{OptimizedSample} = \boldsymbol{V}_g$;	$O(1)$

C. INTERPOLATION BASED ON ADAPTIVE LOCAL MEAN VECTORS

Most variations of SMOTE use the k nearest neighbor-based interpolation to create synthetic minority class samples. Nevertheless, the k nearest neighbor-based interpolation heavily relies on the parameter k and is susceptible to abnormal samples (e.g. outliers, noise or unsafe borderline samples). In the proposed SMOTE-LMVDE, the interpolation based on adaptive local mean vectors is proposed to create synthetic samples without parameters. The proposed interpolation is implied in formula (8).

$$\begin{aligned} &\boldsymbol{New}[d] = \boldsymbol{x}_i[d] + rand(0,1) \times (\boldsymbol{x}_i[d] - \boldsymbol{u}(\boldsymbol{x}_i, \omega_{min})[d]), \\ &d = 1, 2, ..., D \end{aligned} \tag{8}$$

In formula (8), ***New*** is a new synthetic minority class sample based on the base sample $\boldsymbol{x}_i$. $\boldsymbol{u}(\boldsymbol{x}_i, \omega_{min})$ is the adaptive local mean vector calculated by $\boldsymbol{NN}_\lambda(\boldsymbol{x}_i)$ in the minority class. $\boldsymbol{New}[d]$, $\boldsymbol{x}_i[d]$ or $\boldsymbol{u}(\boldsymbol{x}_i, \omega_{min})[d]$ are the dth attribute of ***New***, $\boldsymbol{x}_i$ and $\boldsymbol{u}(\boldsymbol{x}_i, \omega_{min})$. The pseudo-code of the proposed interpolation is described in Lines 11-21 of Algorithm 2. At Line 11, the adaptive local mean vector of each minority class sample is calculated again because imbalanced data is updated by modifying noise. At Line 13, the variable *Num* is the average number of synthetic samples for each minority class sample. Each minority class sample is regarded as the base sample at Line 14. The proposed interpolation uses formula (8) to create synthetic minority class samples at Lines 15-21. Fig. 3 uses synthetic data to visualize the proposed interpolation.

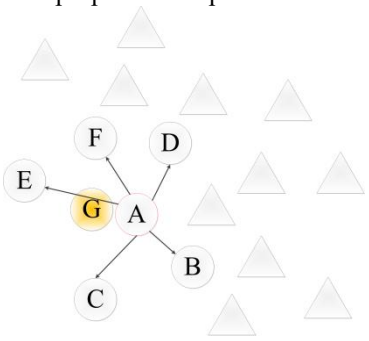

FIGURE 3. Illustrating the proposed interpolation on synthetic data and assuming λ=5.

In Fig. 3, the base sample is sample A, where circles or triangles are samples of minority or majority classes, respectively. $\boldsymbol{NN}_\lambda$(A)={B, C, D, E, F} are searched in the minority class. The adaptive local mean vector of sample A from the minority class (i.e., $\boldsymbol{u}$(A, ω_{min})) is sample G. Sample G is the local mean vector of samples B-E. Sample G can alleviate the negative effect of unsafe borderline samples B and D by the mean vector. The proposed interpolation implied formula (8) will employ samples G and A to create safer synthetic samples and alleviate the effect of unsafe samples B and D.

If k nearest neighbor-based interpolation [5] is used to create synthetic samples, unsafe samples B or D are likely to be employed in the process of interpolation, which increases the error of synthetic samples. Compared to the k nearest neighbor-based interpolation, the proposed interpolation is parameter-free and reduces the error of synthetic samples.

D. TIME COMPLEXITY AND CHARACTERISTIC ANALYSIS

As analyzed by Algorithm 2, the time complexity of computing the Natural Neighbor Eigenvalue λ (Line 2), the proposed noise detection (Line 3), the improved differential evolution (Lines 4-11) and the proposed interpolation (Lines 12-23) are $O(NlogN)$, $O(N)$, $O(G_{max} \times NlogN)$ and $O(N_{min} \times Num) + O(N_{min} logN_{min})$. Because of *Num* and $N_{min} << N$, the time complexity of SMOTE-LMVDE is $O(G_{max} \times NlogN)$. The main characteristics of SMOTE-LMVDE need to be emphasized.

(a) SMOTE-LMVDE is without parameters because the process of the noise detection, the improved differential evolution and the proposed interpolation is parameter-free.

(b) SMOTE-LMVDE can modify found suspicious noisy samples by the proposed improved differential evolution, as shown in Figs. 1, 4 and 5.

(c) SMOTE-LMVDE can create safer synthetic minority class samples by the interpolation based on adaptive local mean vectors, which reduces the effect of unsafe samples (as shown in Figs. 1, 4 and 5).

V. EXPERIMENTS

A Server with Intel(R) Core(TM) i5-1035G4 CPU, 16G memory and 64-bit Windows 10 operating system is used for experiments. Matlab 2021 is used for coding.

A. EXPERIMENTAL DATA SETS

To validate the effectiveness of SMOTE-LMVDE, extensive real data sets are adopted from UCI Machine Learning Repository (http:// archive. ics. uci. edu/ ml/ index. php). Table 1 describes experimental adopted real data sets. #Minority, #Majority, #Attribute and IR represent the number of samples in the minority class, the number of samples in the majority class, the number of attributes and the imbalanced ratio, respectively. The imbalanced ratio is equal to #Majority divided by #Minority. On each data set, the class with the least number of samples is considered the minority class, while others are regarded as the majority class. For some data sets (Sonar, Australian Credit Approval, Wilt and Heart), the minority class samples are removed randomly in order to obtain a higher imbalance ratio.

TABLE I
EXPERIMENTAL REAL DATA SETS

Data sets	#Minority	#Majority	#Attribute	IR	Application Areas
Spambase	1428	2253	57	1.6	Computer
Sonar	22	111	60	5.0	Physical
Australian Credit Approval	77	383	14	5.0	Finance
Vertebral Column	100	210	7	2.1	Biology
Wilt	85	4265	5	50.0	Life
Sani Z-Alizadeh	87	216	55	2.5	Life
USPS	708	8590	256	12.1	Image
Heart	30	150	13	5.0	Medical Science
Vehicle	199	647	18	3.3	Physical
Cardiotocography	176	1950	22	11.1	Medical Science
Abalone	287	2870	8	10.0	Life
Isolet5	59	1500	617	25.4	Life
Biodegradation	356	699	41	2.0	Biology
Pima Indians Diabetes	206	408	8	2.0	Medical Science
German	249	551	24	2.2	Finance
Cervical Cancer	42	645	35	15.4	Medical Science

The stratified k-fold cross-validation with k=5 is adopted to divide each real data set into the training set and the test set. All experiments are repeated 5 times due to the stratified k-fold cross-validation with k=5.

B. EVALUATION METRICS OF IMBALANCED CLASSIFICATION

Imbalanced classification, the minority and majority classes are regarded as positive and negative cases, respectively. F-measure and G-mean are common evaluation metrics for imbalanced classification in existing work [5, 8]. F-measure is a combination of *Precision* and *Recall*. *Precision* and *recall* can evaluate the classification accuracy of positive cases. Hence, if a given classifier achieves a higher F-measure, then the classifier can predict positive cases more accurately. Formula (7)-(9) introduces F-measure.

$$F\text{-}measure = \frac{(1+\beta)\times Recall \times Precision}{\beta^2 \times Recall + Precision} \tag{9}$$

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

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

β=1 in formula (9). TP (True Postive), FP (False Positive) and FN (False Negative) are from the confusion matrix for binary classification. G-mean is fomulated by weighing *Specificty* and *Recall*. Formulas (12) and (13) indicates G-mean.

$$G\text{-}mean = \sqrt{Recall \times Specifity} \tag{12}$$

$$Specifity = \frac{TN}{FP+TN} \tag{13}$$

Specificity can evaluate the classification accuracy of negative cases, while *Recall* can evaluate the classification accuracy of positive cases. A higher G-mean indicates that a given classifier can predict positive and negative cases more accurately. Hence, G-mean can evaluate the overall classification performance in the imbalanced binary classification.

C. COMPARATIVE OVERSAMPLING TECHNIQUES

TABLE II
COMPARATIVE OVERSAMPLING TECHNIQUES

ID	Comparison methods	Parameters
1	SMOTE	k=5, N=2
2	Safe-Level-SMOTE	k=5, N=2
3	MWMOTE	$k1$=5, $k2 \in$ {5, 10, 20}, $k3 = \|\boldsymbol{Smin}\|/2$, C_p = 3, $Cf(th)$=5, $CMAX$ = 2
4	k-means SMOTE	$k \in$ {2, 4, 20, 50, 100}, knn=5, irt=1, de=the number of features, N=2
5	Adaptive-SMOTE	k=5, C=5, N=2
6	RSMOTE	k=5, N=2
7	SMOTE-IPF	The number of iteration=3, the number of partitions=5, k=5 in SMOTE, N=2

The proposed SMOTE-LMVDE aims to overcome imbalances between classes and overgeneralization. Hence, related oversampling techniques with the above objective are used for comparison and described in Table 2. SMOTE is a classical oversampling method. Safe-Level-SMOTE, MWMOTE, k-means SMOTE, Adaptive-SMOTE and RSMOTE are change-direction improvements of SMOTE. SMOTE-IPF is a competitive filtering-based improvement of SMOTE. Their algorithmic ideas have been introduced in Section 1 and Section 2. Parameters of comparative oversampling technique are set as their suggestions.

Additionally, the nearest neighbor classifier and the decision tree classifier (classification and regression tree, CART) [38] are used as the trained classifiers because they are popular in a large number of practical applications [39, 40] and are often used to evaluate the performance of existing oversampling methods.

D. VALIDATING COMPARATIVE METHODS ON SYNTHETIC DATA

Figs. 4 and 5 visualize the results that comparative oversampling methods are performed on synthetic data. Figs. 4 (a) and 5 (a) show the distribution of synthetic imbalanced data with noise, minority class samples, majority class samples. Note that noise is usually located in

overlappings and has a different class label from samples around it in Figs. 4 and 5.

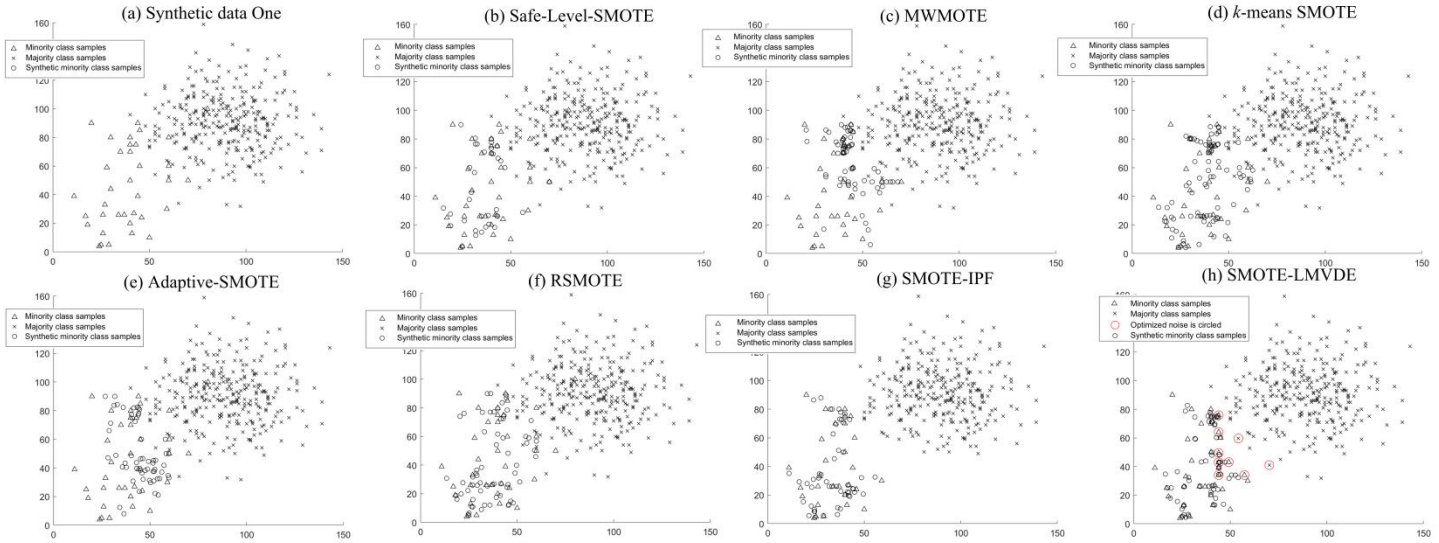

FIGURE 4. Comparative oversampling methods are performed on synthetic data one.

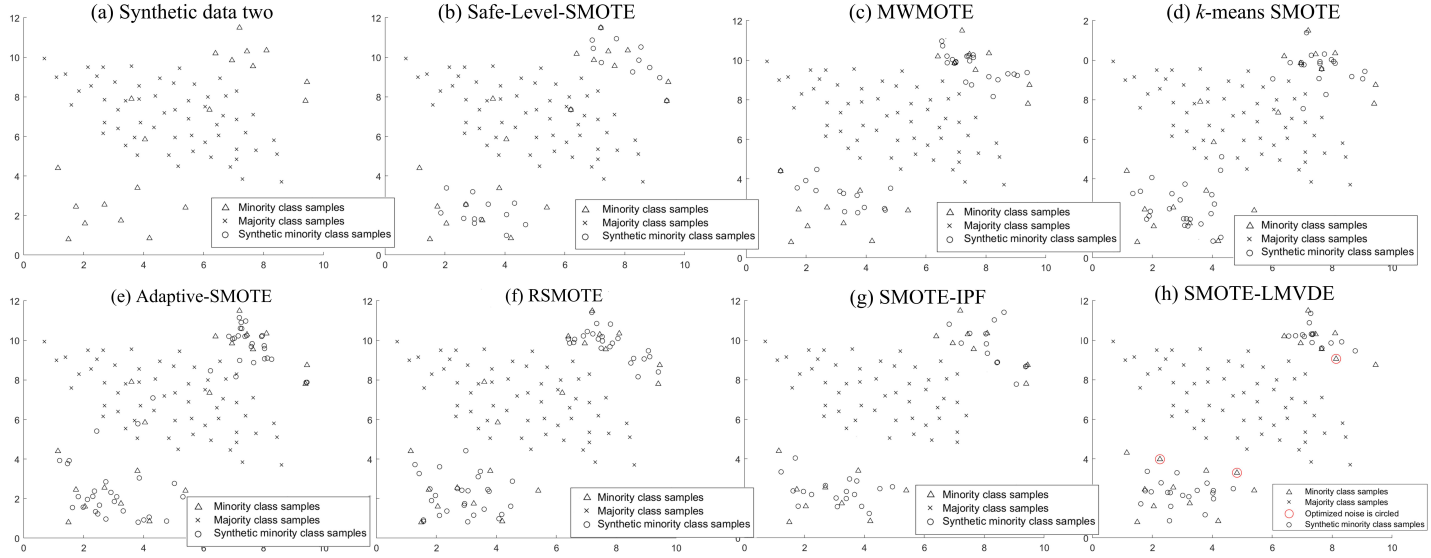

FIGURE 5. Comparative oversampling methods are performed on synthetic data two.

In Figs. 4 and 5, MWMOTE, *k*-means SMOTE and Adaptive-SMOTE creates many unsafe synthetic samples close to the class boundary, which complicates the decision boundary (especially in Fig. 4). Besides, *k*-means SMOTE and Adaptive-SMOTE fail to handle noise in the original and synthetic data. Although MWMOTE can detect and remove a few noises from the minority class, it hardly handles noise from majority class and synthetic data.

In Figs. 4-5 (b) and (f), Safe-Level-SMOTE and RSMOTE create more synthetic samples in central areas. Nevertheless, they also can not deal with noise from the original data. Besides, the *k* nearest neighbor-based interpolation in Safe-Level-SMOTE and RSMOTE lead to noise generation in Fig. 4 (b) and (f). Additionally, SMOTE-IPF removes a large number of suspicious noise instead of modifying them, leading to the loss of information and the destruction of the class boundary in Figs. 4 (g) and 5 (g).

In Figs. 4 (h) and 5 (h), compared to others, the proposed SMOTE-LMVDE can effectively detect suspicious noise and optimize them, which improves the distribution and the class boundary of imbalanced data. Besides, the interpolation based on adaptive local mean vectors can create safer synthetic minority class samples, which reduces the error of synthetic samples and the possibility of overgeneralization.

In general, Figs. 4 and 5 prove that SMOTE-LMVDE, compared to others, can overcome imbalances between classes and overgeneralization more effectively by creating

safer synthetic samples and improving detected suspicious noise.

E. VALIDATING COMPARATIVE METHODS ON SYNTHETIC REAL DATA SETS

The proposed SMOTE-LMVDE is compared with the comparative oversampling technique in training the nearest neighbor classifier and CART classifier. Table 3-6 shows the average *F*-measure and *G*-mean of test classifiers improved by comparative methods. The highest value in each row of Tables 3-6 is bold.

SMOTE-LMVDE achieves the highest *F*-measure in 10 of 16 data sets (Table 3), the highest *G*-mean in 10 of 16 data sets (Table 4), the highest *F*-measure in 10 of 16 data sets (Table 5) and the highest *G*-mean in 8 of 16 data sets (Table 6). This result shows that SMOTE-LMVDE is better than 7 comparative methods on most imbalanced data sets in improving the nearest neighbor classifier and CART classifier.

TABLE III
AVERAGE *F*-MEASURE OF THE NEAREST NEIGHBOR CLASSIFIER IMPROVED BY COMPARATIVE METHODS (%)

Data sets	SMOTE	Safe-Level-SMOTE	MWMOTE	*k*-means SMOTE	Adaptive-SMOTE	RSMOTE	SMOTE-IPF	SMOTE-LMVDE
Spambase	77.17	77.94	79.12	77.97	79.03	78.19	78.87	**79.51**
Sonar	63.32	65.61	62.77	63.00	62.97	65.27	60.97	**65.97**
Australian Credit Approval	36.96	24.39	31.33	26.51	26.51	24.39	21.43	**37.25**
Vertebral Column	71.83	71.01	72.71	70.49	66.28	71.53	75.28	**76.94**
Wilt	78.94	76.21	78.55	77.95	81.33	80.32	80.93	**81.55**
Sani Z-Alizadeh	35.14	36.06	**43.54**	36.67	38.36	35.88	39.41	39.64
USPS	95.12	96.75	96.75	96.75	96.77	96.43	96.75	**97.09**
Heart	33.33	31.11	**46.15**	39.13	31.11	37.50	45.12	40.00
Vehicle	82.76	**84.34**	82.76	83.33	83.33	83.33	83.72	83.11
Cardiotocography	83.58	84.85	82.35	84.85	83.58	83.58	84.48	**86.57**
Abalone	45.40	44.14	44.04	46.32	44.28	47.14	**52.37**	45.31
Isolet5	77.50	79.34	76.89	76.05	73.30	78.36	77.31	**82.82**
Biodegradation	71.10	72.47	72.28	71.79	70.98	71.47	71.34	**73.03**
Pima Indians Diabetes	59.77	59.47	60.09	59.90	60.97	60.11	60.32	**62.32**
German	45.65	42.86	46.74	44.37	42.07	44.10	**47.38**	45.42
Cervical Cancer	26.67	25.00	28.57	25.00	25.00	23.53	**34.78**	30.14
Average	61.51	60.72	62.79	61.26	60.37	61.32	63.15	**64.17**
Mean Rank	3.53	3.81	4.63	3.97	3.69	3.97	5.41	7.00
Wilcoxon signed-rank test	+	+	=	+	+	+	=	N/A

TABLE IV
AVERAGE *G*-MEAN OF THE NEAREST NEIGHBOR CLASSIFIER IMPROVED BY COMPARATIVE METHODS (%)

Data sets	SMOTE	Safe-Level-SMOTE	MWMOTE	*k*-means SMOTE	Adaptive-SMOTE	RSMOTE	SMOTE-IPF	SMOTE-LMVDE
Spambase	81.26	81.91	82.90	81.97	81.94	82.16	82.74	**82.95**
Sonar	64.06	65.28	65.89	65.76	65.92	66.74	64.19	**66.92**
Australian Credit Approval	45.38	35.59	41.71	37.52	37.52	35.59	32.80	**48.02**
Vertebral Column	80.22	79.08	81.18	79.55	75.86	80.37	82.71	**84.29**
Wilt	85.05	82.31	**86.61**	82.35	86.46	85.65	85.70	85.31
Sani Z-Alizadeh	49.29	50.85	**55.96**	51.12	53.13	50.79	53.04	52.46
USPS	98.76	98.39	98.39	98.39	98.73	98.70	98.39	**99.37**
Heart	41.56	40.96	52.09	48.67	40.96	45.88	**55.46**	53.14
Vehicle	90.64	90.45	90.64	90.08	90.08	90.08	**91.02**	90.41
Cardiotocography	91.15	91.27	91.04	91.27	91.15	91.15	91.27	**92.95**
Abalone	59.62	57.18	57.05	58.90	57.30	60.43	58.98	**62.98**
Isolet5	87.86	89.77	87.09	81.94	84.38	87.91	89.41	**94.80**
Biodegradation	80.01	80.43	81.15	80.01	79.44	79.78	79.99	**81.48**
Pima Indians Diabetes	66.67	66.88	66.72	66.90	68.04	67.26	68.07	**70.35**
German	58.01	56.18	57.75	57.28	55.47	57.14	**59.87**	58.66
Cervical Cancer	45.83	44.92	45.97	45.92	43.92	44.82	**54.12**	48.98
Average	70.33	69.47	71.38	69.85	69.39	70.28	71.73	**73.32**
Mean Rank	3.88	3.22	5.00	3.72	3.44	4.03	5.59	**7.13**
Wilcoxon signed-rank test	+	+	+	+	+	+	=	N/A

TABLE V
AVERAGE *F*-MEASURE OF THE CART CLASSIFIER IMPROVED BY COMPARATIVE METHODS (%)

Data sets	SMOTE	Safe-Level-SMOTE	MWMOTE	*k*-means SMOTE	Adaptive-SMOTE	RSMOTE	SMOTE-IPF	SMOTE-LMVDE
Spambase	88.98	89.23	89.17	88.92	89.62	89.79	88.63	**90.21**
Sonar	61.69	64.87	63.50	59.14	58.00	64.13	58.04	**65.71**
Australian Credit Approval	68.85	70.88	60.46	63.74	65.97	70.73	69.10	**71.18**
Vertebral Column	72.25	76.93	75.55	67.44	75.22	75.91	77.53	**77.81**
Wilt	71.82	75.99	62.88	75.56	74.18	72.95	65.13	**77.41**
Sani Z-Alizadeh	63.02	58.99	65.63	65.00	60.23	64.58	69.88	**71.57**
USPS	75.20	75.90	72.55	73.46	74.05	74.95	77.83	**78.25**
Heart	65.46	65.74	54.61	61.32	62.39	60.49	**68.15**	66.11
Vehicle	83.53	80.91	82.97	82.97	**84.16**	82.78	83.59	83.99
Cardiotocography	99.18	98.92	99.11	98.62	98.92	99.21	98.75	**99.58**
Abalone	25.48	31.48	27.07	32.96	33.31	**35.41**	34.80	34.16
Isolet5	48.23	47.09	55.18	43.38	54.04	48.67	53.16	**57.46**
Biodegradation	72.19	74.09	**77.63**	72.22	77.30	75.40	76.58	76.48
Pima Indians Diabetes	63.43	59.76	64.70	65.39	61.65	63.11	**66.38**	65.63
German	51.26	55.89	46.61	49.51	52.56	49.65	52.42	**56.04**
Cervical Cancer	60.13	47.06	40.75	42.86	**63.16**	62.50	57.14	63.11
Average	66.92	67.11	64.90	65.16	67.80	68.14	68.57	**70.92**
Mean Rank	3.63	4.28	3.47	2.84	4.53	4.69	5.13	**7.44**
Wilcoxon signed-rank test	+	+	+	+	+	+	=	N/A

The row labeled "Average" indicates the average value of all data sets. Observing the row labeled "Average" in Table 3-6, SMOTE-LMVDE achieves the highest average *F*-measure and *G*-mean of all data sets. The row labeled "Mean Rank" indicates the mean rank of the Friedman test. If a comparative method performs better, then it has a higher mean rank. Take the spambase dataset as an example in Table 3, the ranks of comparative methods are 1, 2, 7, 3, 6, 4, 5 and 8. The mean rank is the average value of ranks for all data sets. Observing the row labeled "Mean Rank" in Table 3-6, SMOTE-LMVDE achieves the highest mean ranks. These results prove the overall superiority of SMOTE-LMVDE in adapting to different data distributions.

The two-sided Wilcoxon signed-rank test with the default 5% significance level is used to analyze the significant differences between SMOTE-LMVDE and comparative methods. The row labeled "Wilcoxon signed-rank test" indicates the results of the two-sided Wilcoxon signed-rank test. The cell labeled "+" refers to that SMOTE-LMVDE is significantly better than the comparative method of a given column. The cell labeled "=" refers to that there is no significant difference between the proposed algorithm and the comparative method of a given column. Observing the row labeled "Wilcoxon signed-rank test" in Table 3-6, SMOTE-LMVDE is significantly better than most comparative methods.

Additionally, SMOTE-LMVDE can not achieve the highest performance on all data sets. The performance of SMOTE-LMVDE is slightly lower than that of the comparative methods in data sets, such as Sani Z-Alizadeh, Heart, German and Cervical Cancer. Different oversampling techniques have their own adaptive data distribution. SMOTE-LMVDE is more suitable for data sets with overlappings and noise. This is because SMOTE-LMVDE, compared to others, can improve and modify found suspicious noise while generating safer synthetic samples.

TABLE VI
AVERAGE *F*-MEAN OF THE CART CLASSIFIER IMPROVED BY COMPARATIVE METHODS (%)

Data sets	SMOTE	Safe-Level-SMOTE	MWMOTE	*k*-means SMOTE	Adaptive-SMOTE	RSMOTE	SMOTE-IPF	SMOTE-LMVDE
Spambase	90.87	90.93	90.94	90.84	91.35	91.47	90.56	**91.81**
Sonar	72.54	74.04	65.50	68.50	70.28	**74.05**	72.89	73.56
Australian Credit Approval	87.69	86.81	81.97	90.41	90.13	86.52	85.81	**90.91**
Vertebral Column	78.36	81.96	81.05	74.58	**82.77**	81.76	84.73	80.98
Wilt	85.09	84.61	76.81	83.35	85.57	81.50	78.98	**88.68**
Sani Z-Alizadeh	70.65	66.98	72.55	72.02	67.93	71.00	76.01	**77.02**
USPS	87.52	85.70	87.38	85.57	86.97	87.84	**88.34**	87.57
Heart	67.67	68.55	59.18	64.88	65.12	64.18	**70.48**	65.61
Vehicle	90.41	87.47	89.88	89.92	90.30	89.70	**90.66**	90.45
Cardiotocography	98.97	99.17	99.01	98.85	99.33	99.19	99.32	**99.41**
Abalone	35.67	35.60	26.22	31.08	30.21	32.65	28.45	**36.38**
Isolet5	70.91	64.69	**75.62**	63.35	73.07	65.99	67.39	74.08
Biodegradation	79.12	80.70	83.19	78.84	82.96	81.27	82.53	**83.44**
Pima Indians Diabetes	71.04	67.90	72.03	72.76	69.49	70.85	**73.51**	71.12
German	63.12	**66.93**	58.98	61.30	64.06	61.71	64.14	64.06
Cervical Cancer	88.12	82.42	79.86	78.04	89.62	80.66	89.06	**89.34**
Average	77.36	76.53	75.01	75.27	77.45	76.27	77.68	**79.03**
Mean Rank	4.34	4.03	3.38	2.88	5.03	4.19	5.44	**6.72**
Wilcoxon signed-rank test	+	+	+	+	+	+	=	N/A

F. VALIDATINg AVERAGE RUNNING TIME

The average running time of 5 executions of comparative oversampling techniques is shown in Table 7. The results of Table 7 also are analyzed by the mean rank of the Friedman test in the column labeled "Mean Rank". A faster method is with a lower mean rank. On most data sets, the time efficiency of SMOTE-LMVDE is better than MWMOTE and SMOTE-IPF. This is because (a) the adopted hierarchical clustering in MWMOTE is complex and time-consuming (with the time complexity $O(n^2 logn)$); (b) the adaptive noise filter in SMOTE-IPF is an iterative ensemble algorithm and relatively time-consuming. In general, the average running time of SMOTE-LMVDE is suitable for the field of oversampling methods and acceptable.

TABLE VII
AVERAGE RUNNING TIME OF COMPARATIVE OVERSAMPLING TECHNQUES (SEC.)

Data sets	SMOTE	Safe-Level-SMOTE	MWMOTE	*k*-means SMOTE	Adaptive-SMOTE	RSMOTE	SMOTE-IPF	SMOTE-LMVDE
Spambase	0.60	0.84	23.37	0.97	0.96	0.91	4.06	2.85
Sonar	0.09	0.12	0.40	0.08	0.06	0.13	1.76	0.39
Australian Credit Approval	0.89	0.19	1.92	0.19	0.15	0.37	2.23	0.76
Vertebral Column	0.08	0.10	0.35	0.07	0.05	0.09	0.71	0.41
Wilt	0.08	0.10	0.50	0.11	0.07	0.14	0.78	0.66
Sani Z-Alizadeh	0.09	0.15	0.55	0.12	0.07	0.19	0.92	0.90
USPS	0.57	1.11	12.06	7.67	1.49	3.11	6.72	3.99
Heart	0.07	0.12	0.55	0.09	0.05	0.14	0.86	0.40
Vehicle	0.12	0.12	0.97	0.13	0.10	0.17	0.73	0.52
Cardiotocography	0.10	0.13	0.76	0.11	0.15	0.19	0.85	0.51
Abalone	0.09	0.31	27.51	0.12	0.52	0.25	1.93	1.56
Isolet5	0.10	0.13	0.58	0.61	0.10	0.34	2.40	1.05
Biodegradation	0.12	0.19	1.67	0.14	0.20	0.17	0.94	0.58
Pima Indians Diabetes	0.06	0.13	1.29	0.07	0.11	0.11	0.79	0.52
German	0.09	0.17	1.92	0.09	0.08	0.16	0.81	0.86
Cervical Cancer	0.05	0.11	0.28	0.09	0.01	0.10	0.69	1.56
Mean Rank	**2.03**	3.63	7.06	3.25	2.19	4.16	7.38	6.31

VI. CONCLUSIONS

Although SMOTE and its improvements can overcome imbalances between classes, overgeneralization is a great challenge in them. Recently, change-direction and filtering-based oversampling SMOTE-based improvements are proposed against overgeneralization. Yet, they still have the following issues: a) most methods depend on too many parameters; b) most methods fail to detect suspicious noise effectively and modify them; c) interpolation of almost all methods is susceptible to abnormal samples. To overcome imbalances between classes and overgeneralization while improving the above shortcomings of related work, a new synthetic minority oversampling technique based on adaptive local mean vectors and improved differential evolution (SMOTE-LMVDE) is proposed. First, a new noise detection technique based on the defined adaptive local mean vectors (NDALMV) is proposed to find suspicious noise. Second, a new improved differential evolution is proposed to modify and improve detected suspicious noise. Finally, a new interpolation based on the defined adaptive local mean vectors is proposed to create synthetic minority class samples. The main advantages of SMOTE-LMVDE are (a) it is parameter-free; (b) it can modify found suspicious noisy samples rather than removing them; (c) it can create safe synthetic minority class samples, avoiding overgeneralization. The time complexity of SMOTE-LMVDE is $O(G_{max} \times NlogN)$.

The main contributions are (a) the proposed SMOTE-LMVDE; (b) the proposed noise detection technique based on adaptive local mean vectors; (c) the improved differential evolution; and (d) the proposed interpolation based on adaptive local mean vectors.

Intensive experiments are performed on extensive real data sets and two synthetic samples. Experiments prove that (a) SMOTE-LMVDE can overcome imbalances between classes and overgeneralization more effectively by creating safer synthetic samples and improving detected suspicious noise; (b) SMOTE-LMVDE outperforms comparative oversampling technique in training nearest neighbor classifier and CART on extensive data sets with the relatively high imbalance ratio; (c) the average running time of SMOTE-LMVDE is suitable for the field of oversampling methods and acceptable.

REFERENCES

[1] Y. Liu, Z. Yu, C. Chen, Y. Han, and B. Yu, "Prediction of protein crotonylation sites through LightGBM classifier based on SMOTE and elastic net," *Analytical Biochemistry*, vol. 609, Aug. 2020.

[2] Y. Li, H. Guo, Q. Zhang, M. Gu, and J. Yang, "Imbalanced text sentiment classification using universal and domain-specific knowledge," *Knowledge-Based Systems*, vol. 160, pp. 1-15, Jul. 2018.

[3] R. Panigrahi, S. Borah, "Dual-stage intrusion detection for class imbalance scenarios," *Computer Fraud and Security*, vol. 12, pp. 12-19, Dec. 2019.

[4] L. Wang, and C. Wu, "Dynamic imbalanced business credit evaluation based on Learn++ with sliding time window and weight sampling and FCM with multiple kernels," *Information Science*, vol. 520, pp. 305-323, Feb. 2020.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority oversampling Technique," *J. Artificial Intelligence Research*, vol. 16, pp. 321-357, Jun. 2002.

[6] W. Fan, S. Stolfo, J. Zhang, and P. Chan, "Adacost: misclassification cost-sensitive boosting," *in: ICML*, vol. 99, pp. 97-105, May. 1999.

[7] H. Dubey, and V. Pudi, "Class Based Weighted K-Nearest Neighbor over Imbalance Dataset," *in: Advances in Knowledge Discovery and Data Mining*, pp 305-316, Apr. 2013.

[8] D. Elreedy, and A. F. Atiya, "A Comprehensive Analysis of Syntheic Minority Oversampling Technique (SMOTE) for handling class imbalance," *Information Science*, vol. 505, pp. 32-64, Jul. 2019.

[9] J. Li, Q. Zhu, Q. Wu, and F. Zhu, "A novel oversampling technique for class-imbalanced learning based on SMOTE and natural neighbors," *Information Sciences*, vol. 565. pp. 438-455, Jul. 2021.

[10] J. Li, Q. zhu, Q. Wu, Z. Zhang, Y. Gong, Z. He, and F. Zhu, "SMOTE-NaN-DE: Addressing the noisy and borderline examples problem in imbalanced classification by natural neighbors and differential evolution," *Knowledge-Based Systems*, vol. 223: 107056, Apr. 2021.

[11] J. Li, Q. Zhu, Q. Wu, "A parameter-free hybrid instance selection algorithm based on local sets with natural neighbors," *Applied Intelligence*, vol. 50, no. 15, pp. 1527-1541, May 2020.

[12] C. Jia, and Y. Zuo, "S-SulfPred: A sensitive predictor to capture S-sulfenylation sites based on a resampling one-sided selection undersampling-synthetic minority oversampling technique," *Journal of Theoretical Biology*, vol. 7, pp. 84-89, Apr. 2017.

[13] S. Susan, and A. Kumar, "SSOMaj-SMOTE-SSOMin: Three-step intelligent pruning of majority and minority samples for learning from imbalanced datasets," *Applied Soft Computing*, vol. 78, pp. 141-149, Feb. 2019.

[14] A. H. Kamarulzalis, M.H.M. Razali, and B. Moktar, "Data pre-processing using smote technique for gender classification with imbalance hu's moments features," *IISA 2018: Advances in Intelligent, Interactive Systems and Applications*, pp. 351-355, Jan. 2018

[15] C. Liu, J. Wu, L. Mirador, Y. Song, and W. Hou, "Classifying dna methylation imbalance data in cancer risk prediction using smote and tomek link methods," *International Conference of Pioneering Computer Scientists, Engineers and Educators*, pp. 1-9, Feb. 2018.

[16] M. Nakamura, Y. Kajiwara, A. Otsuka, and H. Kimura, "Lvq-smote-learning vector quantization based synthetic minority over-sampling technique for biomedical data," *BioData Mining*, vol. 6, no. 1, pp. 16, Oct. 2013.

[17] J. Zhang, and X. Li, "Phishing detection method based on borderline-smote deep belief network," *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, pp. 45-53, Jan. 2017.

[18] J. A. Sáeza, J. Luengob, J. Stefanowskic, and F. Herreraa, "SMOTE-IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering," *Information Sciences*, vol. 291, pp. 184-203, Jan. 2015.

[19] N. Verbiest, E. Ramentol, C. Cornelis, and F. Herrera, "Preprocessing noisy imbalanced datasets using SMOTE enhanced with fuzzy rough prototype selection," *Applied Soft Computing*, vol. 22, pp. 511-517, Sep. 2014.

[20] C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap, "Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem," *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 475-482, Apr. 2009.

[21] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning," *Proc. Int' l Joint Conf. Neural Networks*, pp. 1322-1328, Jul. 2008.

[22] C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap, "DBSMOTE: Densitybased synthetic minority over-sampling TEchnique," *Applied intelligence*, vol. 36, no. 3, pp. 664-684, Apr. 2011.

[23] S. Barua, M. M. Islam, X. Yao, and K. Murase, "MWMOTE-majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, pp. 405-425, Feb. 2014.

[24] J. Wei, H. Huang, L. Yao, Y. Hu, Q. Fan, and D. Huang, "NI-MWMOTE: An improving noise-immunity majority weighted minority oversampling technique for imbalanced classification problems," *Expert Systems with Applications*, vol. 158, May. 2020.

[25] D. Georgios, B. Fernando, and L. Felix, "Improving imbalanced learning through a heuristic oversampling method based on k-means and smote," *Information Science*, 465: 1-20, Jul. 2018.

[26] T. Pan, J. Zhao, W. Wu, and J. Yang, "Learning imbalanced datasets based on SMOTE and Gaussian distribution," *Information Sciences*, vol. 512, pp. 1214-1233, Oct. 2020.

[27] B. Chen, S. Xia, Z. Chen, B. Wang, and G. Wang, "RSMOTE: A self-adaptive robust SMOTE for imbalanced problems with label noise," *Information Science*, vol. 553, pp. 397-428, Oct. 2020.

[28] 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 Explorations Newsletter*, vol. 6, pp. 20-29, Jun. 2004.

[29] H. Guan, Y. Zhang, M. Xian, H. D. Cheng, X. Tang, "Smote-wenn: solving class imbalance and small sample problems by oversampling and distance scaling," *Applied Intelligence*, 51: 1394-1409, Mar. 2021.

[30] I. Triguero, S. García, and F. Herrera, "Differential evolution for optimizing the positioning of prototypes in nearest neighbor classification," *Pattern Recognition*, vol. 44, pp. 901-916, Apr. 2011.

[31] I. Triguero, S. García, F. Herrera, "Seg-ssc: a framework based on synthetic examples generation for self-labeled semi-supervised classification," *IEEE Transactions on Cybernetics*, vol. 45, no. 4, pp. 622-634, 2015.

[32] K. J. Wang, A. M. Adrian, K. M. Chen, and K. M. Wang, "A hybrid classifier combining borderline-smote with airs algorithm for estimating brain metastasis from lung cancer: a case study in taiwan," Computer Methods and Programs in Biomedicine, vol. 119, pp. 63-76, May 2015.

[33] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTEBoost: Improving Prediction of the Minority Class in Boosting," *Knowledge Discovery in Databases: PKDD 2003, 7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pp. 22-26, Jan. 2003.

[34] Z. Q. Zeng, and J. Gao, "Improving SVM Classification with Imbalance Data Set," *Conference: Proceedings of the 16th International Conference on Neural Information Processing: Part I*, pp. 389-398, Dec. 2009.

[35] B. S. Raghuwanshi, and S. Shukla, "SMOTE based class-specific extreme learning machine for imbalanced learning," *Knowledge-Based Systems*, vol. 187, Jun. 2020.

[36] Q. Zhu, J. Feng, J. Huang, "Natural neighbor: a self-adaptive neighborhood method without parameter k," *Pattern Recognition Letters*, vol. 80, no.1, pp. 30-36, Sep. 2016.

[37] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, pp. 509-517, Jan. 1975.

[38] C. Munyati, "Comparative performance of regression tree and parametric classification of savannah woody cover on SPOT 6 NAOMI imagery," *Remote Sensing Applications: Society and Environment*, vol. 13, pp. 171-182, Oct. 2019.

[39] J. López, S. Maldonado, "Redefining nearest neighbor classification in high-dimensional settings," *Pattern Recognition Letters*, vol. 110, pp. 36-43, Mar. 2018.

[40] K. Skedgell, C. A. Kearney, "Predictors of school absenteeism severity at multiple levels: A classification and regression tree analysis," *Children and Youth Services Review*, vol. 86, pp. 236-245, Feb. 2018.