Extended Natural Neighborhood for SMOTE and its Variants in Imbalanced Classification

Hongjiao Guan, Long Zhao*, Xiangjun Dong, Chuan Chen

Faculty of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Shandong Computer Science Center (National Supercomputer Center in Jinan), Jinan 250353, China

Abstract

Imbalanced data classification is a challenging issue encountered in many practical applications. Synthetic minority oversampling technique (SMOTE) and its variants are popular resampling methods. However, in most of these methods, the neighborhood determined by k-nearest neighbor (kNN) cannot reflect the local distribution precisely, leading to the generation of noisy examples. To solve this problem, we propose a neighborhood concept without parameter k called extended natural neighbor (ENaN), which is derived from natural neighbor (NaN). ENaN unites kNN and reverse kNN to determine neighbors adaptively according to the sample distribution. Compared to NaN, ENaN explores broad neighborhoods, which facilitates to improve the quality of generated examples. ENaN-based SMOTE (ENaNSMOTE) can improve the sample distribution obtained by SMOTE and NaNSMOTE. Extensive experiments using 30 synthetic and 20 real-world datasets prove the effectiveness of ENaN in SMOTE and its variants.

Keywords: imbalanced classification, SMOTE, extended natural neighbor, k-nearest neighbor, reverse k-nearest neighbor

1. Introduction

The problem of class imbalance occurs in datasets where one class has far more examples than the other class. The class with more samples is called the majority or negative class, and the remaining class is called the minority or

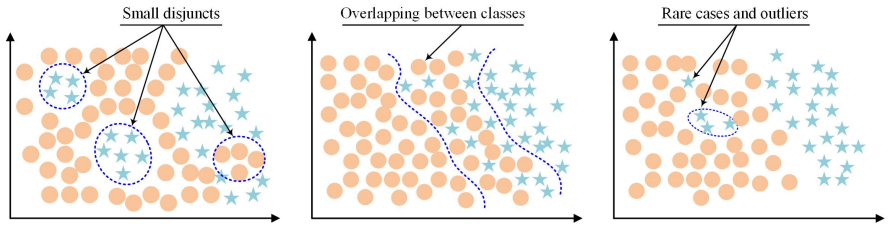


Figure 1 Illustrations of complex distribution characteristics in imbalanced data. Circles and stars respectively indicate majority and minority class examples.

positive class. Note that binary classification is considered in this study. Many practical applications, including fault diagnosis and credit assessment, suffer from this problem (Tian et al. (2021); Wang et al. (2021a)). Hence, imbalanced data classification is challenging and significant in machine learning and artificial intelligence.

Traditional classification methods have difficulty coping with imbalanced datasets (Wang et al. (2021b); Luo et al. (2021); Du et al. (2020)), with the minority class often having an extremely low recognition rate. Unfortunately, the misclassification cost of the minority class is typically higher than that of the majority class. Complicated distribution characteristics are the primary cause of the degraded classification performance (Guan et al. (2021b)). These characteristics include small disjunctions, overlap between classes, rare cases, and outliers in the minority class, as shown in Fig. 1.

Data resampling is an effective solution for imbalanced data classification, which aims to obtain a roughly balanced class distribution. Synthetic minority oversampling technique (SMOTE) (Chawla et al. (2002)) is a commonly used oversampling method, which generates new minority class examples between each minority class example and its k-nearest neighbors (kNNs). SMOTE has the following disadvantages. (a) The number of nearest neighbors k needs to be set manually, so SMOTE is parameter-dependent. (b) Each minority class example uses the same number of nearest neighbors and generates the same number of new samples, without considering the sample distribution. (c) SMOTE is

likely to generate noisy examples. For instance, the new minority class examples may lie in the space of the majority class, leading to overlap between classes.

Many SMOTE variants have been proposed to overcome the shortcomings of SMOTE. For example, BL-SMOTE (Han et al. (2005)) and ADASYN (He et al. (2008)) generate different number of examples around each minority class example according to the local distribution. SMOTE-based hybrid resampling methods filter noisy examples after obtaining balanced samples using SMOTE, such as SMOTE-WENN (Guan et al. (2021b)) and SMOTE-RkNN (Zhang et al. (2022)). These methods focus on the latter two issues, i.e., undifferentiated oversampling and noise generation. Recently, Li et.al (Li et al. (2021)) proposed NaNSMOTE, which replaces kNN using natural neighbor (NaN). NaNSMOTE determines neighbors adaptively without setting parameter k. However, NaNSMOTE is likely to ignore rare cases and outliers in the minority class.

To solve these problems, we propose a new neighborhood concept called extended natural neighbor (ENaN) for SMOTE (named ENaNSMOTE) and its variants. ENaN expands the friend relationship of NaN by adding a concept called unilateral friendship. Experimental results demonstrate the effectiveness of ENaN in SMOTE and its variants. The proposed ENaN and ENaNSMOTE can overcome the shortcomings of SMOTE, and their advantages are mainly in three aspects. (a) ENaN does not depend on any parameters and determines neighbors adaptively according to the sample distribution. (b) ENaNSMOTE can improve the sample distribution obtained by original SMOTE and NaNSMOTE. (c) As a neighborhood method, ENaN can be used in any SMOTE-related methods.

The remainder of this paper is organized as follows. Section 2 reviews related work about imbalanced classification methods. Section 3 introduces the proposed ENaN and ENaNSMOTE in detail. Experimental results are reported and analyzed in Section 4. Finally, Section 5 concludes this study.

2. Related work

Imbalanced classification methods can be categorized into algorithmic level and data level methods. At the algorithmic level, cost-sensitive learning and ensemble learning are commonly used techniques (Mostafaei et al. (2022); Fernandes et al. (2020)). The idea of cost-sensitive learning is to assign a larger cost to the minority class than to the majority class (Jiang et al. (2015, 2014)). The main disadvantage of cost-sensitive learning is that the actual costs for each class are typically unknowable. Resampling-based ensemble methods have shown promising results and received significant attention (Guan et al. (2021a); Xu et al. (2021); Seng et al. (2021)). They incorporate resampling techniques with bagging or boosting, such as EasyEnsemble (Liu et al. (2009)), uNBBag (Błaszczyński and Stefanowski (2015)), and SPE (Liu et al. (2020)).

At the data level, the classical oversampling method is SMOTE, which generates synthetic instances between each positive seed and its k nearest neighbors. SMOTE has several drawbacks. First, SMOTE is parameter-dependent and the number of nearest neighbors k needs to be set artificially. An appropriate k value can improve the quality of the generated samples. However, the appropriate value of k differs between datasets, and finding a suitable k value for each dataset is inefficient. Second, SMOTE generates new samples without considering the sample distribution. All minority class examples use the same number of nearest neighbors and they are used as seeds to generate the same number of new samples. Third, SMOTE may generate noisy examples due to the former two issues.

Many SMOTE's variants have been proposed to improve SMOTE. Table 1 lists SMOTE and its variants involving three issues: which minority examples to choose as seeds, how to choose the nearest neighbors, and what sampling method to apply. Some methods use kNNs as neighbors or oversample all minority class examples with the same probability (i.e., hard sampling). Therefore, they are also parameter-dependent or do not consider sample distributions. NaNSMOTE uses NaNs as neighbors, which does not need to set parameter k. The number

Table 1 Summarization of previous oversampling methods.

Method	Seeds	Neighbors	Sampling
SMOTE	All minority examples	kNN	Hard
BL-SMOTE	Border minority examples	kNN	Hard
SL-SMOTE	All minority examples	kNN	Hard
LN-SMOTE	All minority examples	kNN	Hard
ADASYN	All minority examples	kNN	Soft(instance-level)
MWMOTE	Border minority examples	cluster	Soft(instance-level)
kmeans-SMOTE	Filtered cluster	kNN	Soft(cluster-level)
GDO	All minority examples	Gaussian-based	Soft(instance-level)
NaNSMOTE	All minority examples	NaN	Hard

of NaNs is different for each example, which is determined automatically based on the sample distribution. However, NaNSMOTE tends to ignore rare cases and outliers in the minority class.

In addition to the oversampling methods mentioned above, SMOTE-based hybrid resampling methods aim to detect and delete noisy samples after obtaining a balanced dataset using SMOTE. These hybrid methods include SMOTE-ENN (Batista et al. (2004)), SMOTE-WENN, and SMOTE-RkNN. SMOTE-ENN and SMOTE-WENN detect noise by evaluating heterogeneity in local neighborhoods. SMOTE-RkNN uses density information obtained by RkNN to clear noise. Inevitably, these hybrid methods depend on the k parameter because they employ SMOTE.

SMOTE uses kNN to search nearest neighbors and then generates new examples between the seed and its nearest neighbors. Therefore, the precision of the nearest neighbors influences the quality of generated examples (Faisal and Tutz (2022)). The nearest neighbors obtained by kNN are affected by three parameters, namely the number of neighbors k, the distance metric, and the weights of the attributes (Kahraman (2016); Bian et al. (2022)). Many methods have been proposed to enhance the performance of kNN by improving these three issues, such as determining the optimal neighbor number k, using fuzzy distance metrics and genetic algorithm-based weight-tuning methods. In this paper, we focus on the issue about the number of neighbors and propose a new neighbor-

hood method for determining neighbors adaptively. The main contributions of this paper are as follows:

(a) A new neighborhood concept named extended natural neighbor (ENaN) is proposed. On the one hand, ENaN can determine nearest neighbors adaptively based on the sample distribution, so it does not require to set the number of neighbors artificially. On the other hand, ENaN improves natural neighbor (NaN) and facilitates the generation of high-quality minority class examples.

(b) ENaN-based SMOTE (ENaNSMOTE) is proposed to deal with imbalanced datasets, which overcomes the shortcomings of SMOTE and its variants. ENaNSMOTE can improve the sample distribution obtained by original SMOTE and NaNSMOTE.

(c) Extensive experiments are conducted on 30 synthetic and 20 real-world datasets. ENaN is proved to be better than kNN and NaN in SMOTE and its variants.

3. Extended natural neighbor

3.1. Natural neighbor

Natural neighbor (NaN) (Zhu et al. (2016)) is a new approach that was first proposed in 2016. In contrast to kNN and RkNN (Radovanović et al. (2015); Sadhukhan and Palit (2019)), NaN is independent of hyperparameters, and captures the data distribution adaptively. The strength of NaN has led to its wide use in machine learning tasks, such as classification (Li et al. (2021, 2019)), clustering (Cheng et al. (2019, 2017)), instance reduction (Yang et al. (2018); Zhao and Li (2020)), and outlier detection (Huang et al. (2017)).

NaN is inspired by friendships in human society, and uses the term "true friends" to refer to two people who each consider the other to be a friend. A natural stable structure (NSS) holds if everyone (except strangers) has at least

one true friend (Zhu et al. (2016)). Similarly, if example x is one of the λ-nearest neighbors of example y **and** y is one of the λ-nearest neighbors of x, then x is called the natural neighbor of y, and vice versa (Definition 1). An NSS is obtained when every example (except noise) has at least one NaN. $NaN(y)$ is defined as the intersection of $NN_\lambda(y)$ and $RNN_\lambda(y)$. Note that if example y is one of the r-nearest neighbors of example x, x is called one of the reverse r-nearest neighbors of y; that is, $y \in NN_r(x) \iff x \in RNN_r(y)$.

Definition 1: (Natural Neighbor)

$$\begin{aligned} x \in NaN(y) &\iff x \in NN_\lambda(y) \wedge y \in NN_\lambda(x) \\ &\iff x \in NN_\lambda(y) \wedge x \in RNN_\lambda(y) \end{aligned}$$

In Definition 1, λ is called the natural neighbor eigenvalue (NaNE), which is the minimum number of searches r required to build the NSS. That is, all examples (except noise) have NaN neighbors until the number of neighbors r increases from 1 to λ. Therefore, the value of λ is related to the data distribution and varies from one dataset to another. In Definition 2, x and y denote two examples and N^+ denotes the set of $\{1, 2, 3, \dots\}$.

Definition 2: (Natural Neighbor Eigenvalue)

$$\lambda = argmin_r\{(\forall r \in N^+)(\forall y)(\exists x \neq y), (y \in NN_r(x)) \wedge (x \in NN_r(y))\}$$

3.2. Extended natural neighbor

Generally, four types (safe, borderline, rare, or outlying) of examples are defined according to the sample distribution. Safe examples are those located in the homogeneous regions of each class. Borderline examples exist near class boundaries and are therefore more significant for learning classifiers than safe examples (Yang et al. (2020)). Rare cases and outliers are often the result of an underrepresented minority class, and they are likely to be identified as noise. Typically, minority class examples are absolutely scarce, so we declare an assumption that each minority class example is valuable and should not be abandoned without due care, especially for rare cases and outliers.

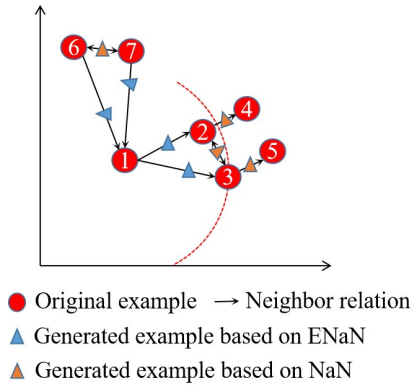


Figure 2 Illustration of empty NaN. Take two nearest neighbors for example. The NaN set of x_1 is empty whereas the ENaN set of x_1 contains x_2, x_3, x_6, and x_7.

An example's NaN is the intersection of its nearest neighbors and reverse nearest neighbors. Because minority class examples are inherently sparse, the NaN set of a minority class example may be empty or contain few instances. If the NaN set is empty, this example is considered noise, and will not be used as the bases for any synthetic examples. Figure 2 presents an illustration of an empty NaN set. In this figure, each example is assigned two nearest neighbors, which are indicated by the arrows. The two nearest neighbors of the example x_1 are x_2 and x_3. The reverse nearest neighbors of x_1 are x_6 and x_7. Therefore, the NaN set of x_1 is empty and x_1 is regarded as a noisy example. NaNSMOTE will not generate new instances in the vicinity of x_1. However, according to our assumption, each minority class example is valuable and should be treated prudently owing to the absolute scarcity of minority class examples (Guan et al. (2021b)). NaNSMOTE tends to generate new examples in safe regions, such as x_2–x_5. Therefore, when generating synthetic positive instances in SMOTE, using NaNs as neighbors does not fit the original data distribution well.

To overcome this problem, we propose the extended natural neighbor. If example x is one of the λ-nearest neighbors of y **or** y is one of the λ-nearest neighbors of x, then x (y) is called the extended natural neighbor of y (x). From Definition 3, $ENaN(y)$ is the union of $NN_\lambda(y)$ and $RNN_\lambda(y)$. Therefore,

$NaN(y) \subseteq ENaN(y)$. As shown in Fig. 2, the ENaNs of x_1 are x_2, x_3, x_6, and x_7. ENaN expands the NaN by combining the concepts of unilateral friendship and true friendship. A person x is a unilateral friend of y if y is a friend of x but x is not a friend of y. ENaN is a looser relation than NaN. Despite being unilateral friends, x and y know each other and have similarities. Therefore, ENaN embodies the social relationship. Note that the Euclidean distance was used in the three neighborhood methods kNN, NaN, and ENaN, with reference to previous literature (Zhu et al. (2016); Li et al. (2021)).

Definition 3: (Extended Natural Neighbor)

$$\begin{aligned} x \in ENaN(y) &\iff x \in NN_\lambda(y) \vee y \in NN_\lambda(x) \\ &\iff x \in NN_\lambda(y) \vee x \in RNN_\lambda(y) \end{aligned}$$

Algorithm 1 shows the pseudocode for searching for ENaNs. It is straightforward, and the process of searching for ENaNs is to build the NSS. Once the NSS is constructed based on sample set S, the NaNE value λ and ENaNs of each example are determined. It should be noted that searching for NaN and ENaN reaches the same λ. The difference lies in line 11 of Algorithm 1, specifically that the NaNs of an example are the intersection of its λ-nearest neighbors and reverse λ-nearest neighbors, whereas the ENaNs are their union. The ENaN (and NaN) neighborhood of each example are different from each other. According to previous definitions, the number of NaNs of an example is less than λ whereas the number of ENaNs is greater than λ.

$$\begin{aligned} &|NN_\lambda| = \lambda, \ \ |RNN_\lambda| \leq \lambda, \\ &NaN = NN_\lambda \cap RNN_\lambda, \ \ 0 < |NaN| \leq \lambda, \\ &ENaN = NN_\lambda \cup RNN_\lambda, \ \ \lambda \leq |ENaN| \leq |S|. \end{aligned}$$

The time complexity of Algorithm 1 is $O(n\log n)$, where n denotes the number of examples in S. In Algorithm 1, the nearest neighbors can be obtained by creating a KD-tree of sample set S, and its time complexity is $O(n\log n)$. Subsequently, for each while-loop, the time complexity is $O(n\log n)$. The while-loop is generally performed a few times, the number of which is far smaller than n.

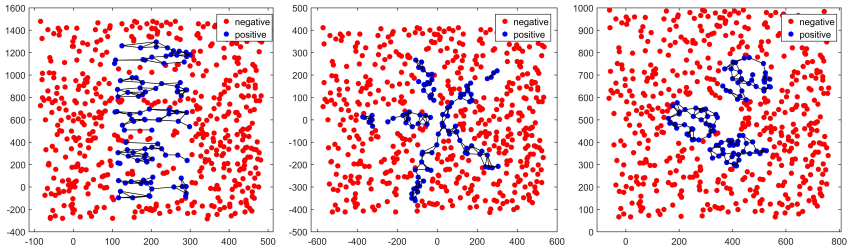
Algorithm 1: Searching for extended natural neighbors (ENaN_Search)

Input: S, Sample set.

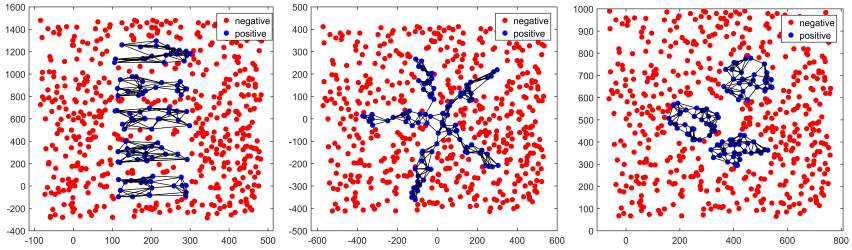
Output: $ENaN$, set of extended natural neighbors of each example in S.

```
Initialization: r = 1, ∀x_i ∈ S, NN_r(x_i) = ∅, RNN_r(x_i) = ∅, nb(x_i) = 0;
while r < |S| do
    foreach x_i ∈ S do
        Find r-th nearest neighbor x_j of x_i in S;
        NN_r(x_i) = NN_r(x_i) ∪ {x_j}, RNN_r(x_j) = RNN_r(x_j) ∪ {x_i}, nb(x_j) =
          nb(x_j) + 1;
    end
    n(r) = {|x_i|, nb(x_i) == 0};
    if r > 1 & n(r) == n(r − 1) then
        λ = r − 1;
        foreach x_i ∈ S do
            ENaN(x_i) = NN_λ(x_i) ∪ RNN_λ(x_i);
        end
        return ENaN;
    else
        r = r + 1;
    end
end
```

The difference between NaN and ENaN can be presented intuitively using the natural neighbor graph (NaNG) and extended natural neighbor graph (ENaNG). Each vertex of the ENaNG represents an example in S. There is an edge between two vertexes if one example is an extended natural neighbor of the other. The NaNG is defined similarly. Figure 3 shows the NaNG and ENaNG of minority class examples in three synthetic datasets. The three datasets were produced using small disjunctions of different shapes. The details of the three synthetic datasets are presented in Section 4.1. The red and blue dots represent majority and minority class examples, respectively. As shown in Fig. 3, ENaNG presents a better affinity relationship among the minority class examples than NaNG, particularly in the boundary areas. Specifically, some borderline examples are the ENaN neighbors of other borderline examples but



(a) sub-cluster(NaN) (b) clover(NaN) (c) paw(NaN)



(d) sub-cluster(ENaN) (e) clover(ENaN) (f) paw(ENaN)

Figure 3 NaNG and ENaNG of minority class examples of three synthetic datasets.

not NaN neighbors. This facilitates the generation of new instances in broader neighborhoods and promotes the quality of generated minority class examples.

Definition 4: (Extended Natural Neighbor Graph)

$$ENaNG =< V, E >, V = S, E = \{(x_i, x_j), x_i \in ENaN(x_j)\}$$

3.3. Extended natural neighbor-based SMOTE

The standard SMOTE yields synthetic examples by linear interpolation between each minority class instance and its kNNs in the same class. SMOTE has the problem of generating noisy examples owing to parameter dependency and blindness in choosing neighbors. Thus, synthetic minority class examples may appear in the majority class space, resulting in overlap between classes and blurring of class boundaries. Using kNN in SMOTE does not capture the local structure correctly.

Algorithm 2: Dealing with imbalanced data using ENaNSMOTE

Input: Tr, imbalanced training set.

Output: Bal_Tr, balanced training set.

1 Divide Tr into positive and negative subsets: $Tr = Pos \cup Neg$;

2 Compute the number n_{gen} of minority class examples that should be generated artificially: $n_{gen} = n_{neg} - n_{pos}$;

3 Determine the ENaNs of each example: $ENaN$ = ENaN_Search(Pos);

4 Generate synthetic minority class examples between each original positive instance and its ENaNs using linear interpolation: Gen_Pos = SMOTE($Pos, ENaN, n_{gen}$);

5 $Bal\_Tr = Gen\_Pos \cup Tr$.

NaNSMOTE (Li et al. (2021)) is a parameterless SMOTE method that uses NaNs. NaNSMOTE alleviates the problem of noise generation owing to the adaptivity of natural neighbors. However, the use of NaN is insufficient. NaNSMOTE tends to generate synthetic examples around safe examples, not unsafe (borderline, rare, and outlying) examples, as illustrated in Fig. 2, so the generated instances may not fit the original data distribution well. ENaN-based SMOTE (ENaNSMOTE) generates synthetic examples between the seed and its ENaNs. Compared to NaNSMOTE, ENaNSMOTE expands the boundary region of the minority class, where synthetic samples will be produced. In brief, one advantage of ENaN is that it does not need to set the number of neighbors. Moreover, using ENaN in SMOTE not only eases the problem of overgeneralization (or generating noise) but also captures the sample structure of the minority class well. Algorithm 2 shows the pseudocode for ENaNSMOTE.

Figure 3 shows that ENaN explores wider local regions and captures more neighbors than NaN while preserving the data structure. Therefore, ENaNSMOTE can generate a larger number of effective borderline positive examples than NaNSMOTE. This is demonstrated by analyzing the distribution of the four types of examples in Table 4. In general, there are two types of borderline examples (Sáez et al. (2015)): examples in overlapping regions and examples close to the boundary. Overgeneralization refers to the former, whereas effective borderline examples are the latter. The preservation of the local structure

reduces overgeneralization in ENaNSMOTE, and a larger number of effective borderline examples helps construct more precise class boundaries when generating the same number of positive examples.

4. Experiments

4.1. Datasets

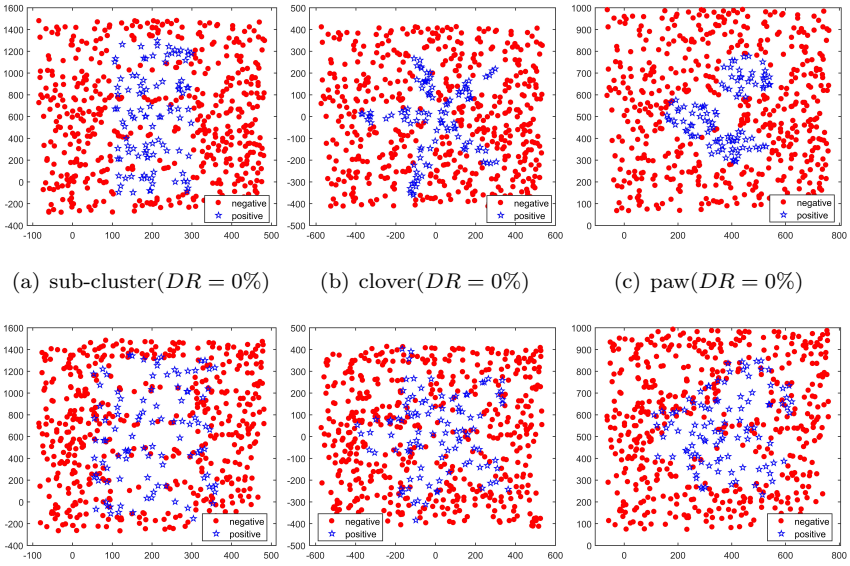
Three groups of synthetic datasets and 20 real-world datasets were used in the experiments. The synthetic datasets were derived from three datasets with small disjunctions in sub-cluster, clover, and paw shapes. Among each group, ten datasets were produced with two factors (Guan et al. (2021b)): imbalance ratio (IR) and disturbance ratio (DR). The imbalance ratio quantifies the degree of class imbalance, defined as the number of majority class samples divided by the number of minority class samples. The disturbance ratio represents the degree of overlap between classes or the disturbance of subregion borders in the minority class (Stefanowski (2013); Napierała et al. (2010)). Two IRs were used: datasets with $IR = 5$ had 600 samples and datasets with $IR = 7$ had 800 samples. Five DRs were used: 0%, 30%, 50%, 60%, and 70%. Figure 4 shows synthetic datasets with $IR = 5$, $DR = 0\%$ and $IR = 5$, $DR = 70\%$. Table 2 lists the characteristics of the 20 real datasets in terms of number of examples, number of minority class examples, imbalance ratio, and number of attributes. These datasets are available in the KEEL repository [1].

4.2. Settings

Two experiments were performed in this study.

- Comparison of three different neighborhoods. We compared the results of kNN, NaN, and ENaN used in SMOTE.

[1] http://www.keel.es/



(a) sub-cluster($DR = 0\%$) (b) clover($DR = 0\%$) (c) paw($DR = 0\%$)

(d) sub-cluster($DR = 70\%$) (e) clover($DR = 70\%$)) (f) paw($DR = 70\%$)

Figure 4 Synthetic datasets with IR=5, $DR = 0\%$ and $IR = 5$, $DR = 70\%$.

Table 2 Characteristics of real datasets.

Dataset	Ins	Min	IR	Att	Dataset	Ins	Min	IR	Att
breast-w	699	239	1.9	9	cmc	1473	333	3.42	9
new-thyroid	215	35	5.14	5	cleveland	303	35	7.66	13
vehicle0	846	199	3.25	18	abalone	4177	335	11.47	8
nursery	12960	328	38.51	8	solar-flare2	1066	43	23.79	12
satimage	6435	626	9.3	36	transfusion	748	178	3.2	4
car-good	1728	69	24.04	6	yeast4	1484	51	28.1	8
credit-g	1000	300	2.33	20	balance	625	49	11.76	4
ecoli3	336	35	8.6	7	winequality	1599	53	29.17	11
haberman	306	81	2.78	3	pima-full	768	268	1.87	8
breastcancer	286	85	2.36	9	postopera	87	24	2.63	8

- Validation of ENaN in SMOTE's variants. We replaced conventional SMOTE with ENaNSMOTE in five SMOTE's variants and compared the original kNN versions with ENaN versions.

In each experiment, a stratified five-fold cross-validation was performed. This procedure was carried out five times, and the average results were re-

ported. Two classification methods were used to evaluate performance. One was the unpruned CART decision tree, which is commonly used in imbalanced classification. The other method was the kNN classifier ($k \in \{1, 3, 5, \ldots, 21\}$), which was used because it is a classifier associated with local neighbors. Note that there are three parameters that influence the performance of kNN: the number of neighbors k, the distance metric, and the weights of the attributes. When kNN was used as a neighborhood method in SMOTE, the issue about the number of neighbors is improved. Therefore, the SMOTE methods with kNN, NaN, and ENaN were compared with respect to this issue and the other two parameters were not considered in the experiment. The Euclidean distance and equal weights of the attributes were used.

The classification performance was assessed using the area under the receiver operating characteristic (ROC) curve (AUC) and G-mean (Guan et al. (2019)). AUC and G-mean are comprehensive metrics that are insensitive to class imbalance. AUC was calculated using the arithmetic mean of sensitivity and specificity (Xie et al. (2022); López et al. (2013)). G-mean is the geometric mean of sensitivity and specificity. Sensitivity and specificity respectively denote the percentages of correctly recognized examples in the minority and majority classes. All experiments were performed using MATLAB R2017a [2].

4.3. Results and analysis

4.3.1. Comparison of neighborhoods

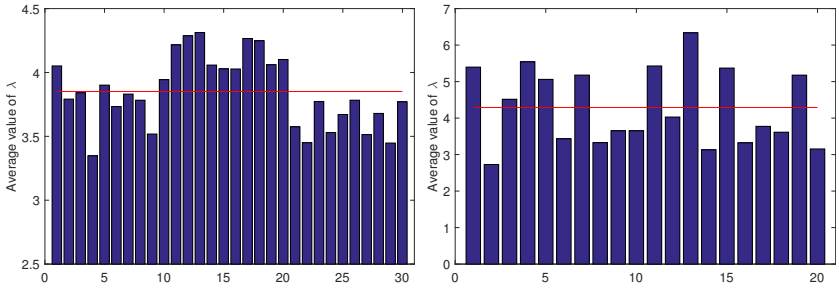
In this section, we compared three neighborhoods in SMOTE: kNN, NaN, and ENaN. kNN was used in the conventional SMOTE. The number of neighbors k was artificially set; we used $k = 3$ and $k = 5$ in this experiment. In addition, the value of NaNE ($k = \lambda$) was used. NaNSMOTE and ENaNSMOTE used NaN and ENaN to determine neighbors, respectively, so they did not need to set parameter k.

[2]The code will be available at https://github.com/Hacker-Andy/MATLAB-Source-Code-ImbalancedClassification

Table 3 Results obtained by decision tree on synthetic and real datasets.

Datasets	Metric	3NN	4NN	5NN	λNN	NaN	ENaN
Syn.	**AUC**	0.7891	0.7946	0.7976	0.7914	0.7910	**0.8038**
	Avg. Rank	4.5167	3.3667	2.8667	4.3833	3.9667	**1.9000**
	p_{Hochberg}	<0.001*	0.005*	0.045*	<0.001*	<0.001*	-
	p_{Wilcoxon}	<0.001*	<0.001*	0.004*	<0.001*	<0.001*	-
	G-mean	0.7780	0.7855	0.7895	0.7812	0.7806	**0.7968**
	Avg. Rank	4.5333	3.3333	2.9000	4.3667	4.0000	**1.8667**
	p_{Hochberg}	<0.001*	0.005*	0.032*	<0.001*	<0.001*	-
	p_{Wilcoxon}	<0.001*	<0.001*	0.005*	<0.001*	<0.001*	-
Real	**AUC**	0.6915	0.6905	0.6948	0.6906	0.6909	**0.6969**
	Avg. Rank	3.75	4.15	3.6	3.6000	3.2500	**2.6500**
	p_{Hochberg}	0.217	0.056*	0.217	0.217	0.311	-
	p_{Wilcoxon}	0.179	0.079*	0.135	0.067*	0.156	-
	G-mean	0.6329	0.6359	0.6364	0.6294	0.6249	**0.6441**
	Avg. Rank	3.65	3.75	3.6500	3.7000	3.8	**2.45**
	p_{Hochberg}	0.043*	0.043*	0.043*	0.043*	0.043*	-
	p_{Wilcoxon}	0.067*	0.048*	0.191	0.023*	0.044*	-

Table 3 presents the AUC and G-mean results obtained using the decision tree classifier on the synthetic and real datasets. The best cases are highlighted in bold type. For each metric, four rows of statistics are presented. The first row lists the average values of the five-fold cross-validation experiment. The second row shows the average rank of using each neighborhood. The ranks were obtained using the Friedman test (Friedman (1937)), which evaluates the performance of methods across multiple datasets. The better the method, the lower the rank. The pos-hoc test after the Friedman test was performed using the Hochberg test (García et al. (2009)). The p-values of the Hochberg test are listed in the third row. Besides, we also executed the Wilcoxon signed-rank test (Rosner et al. (2006)) to compare the proposed method with the rest of methods, and the p-values are shown in the fourth row. If the significance level p-value is less than 0.1, the difference in the compared methods is considered



(a) Synthetic datasets (b) Real datasets

Figure 5 Distribution of the values of NaNE λ for all synthetic and real datasets.

statistically significant, which is marked with an asterisk.

Table 3 shows that using ENaN in SMOTE performed best compared with using kNN and NaN. SMOTE with $k = 5$ outperformed SMOTE with $k = 3$ and $k = \lambda$. We computed the values of λ for each dataset as shown in Fig. 5. The averaged NaNE values (denoted by red lines) of the synthetic and real datasets are 3.85 and 4.29, respectively. Therefore, we added the results of using kNN ($k = 4$) as the neighborhood. As explained, the number of neighbors in NaN is not larger than the value of NaNE λ whereas the number of neighbors in ENaN is not smaller than λ. The fact that 5NN performed better than 3NN proves the superiority of ENaN over NaN.

Figure 6 shows the changes in AUC and G-mean obtained by kNN classifiers with the increase in the number of nearest neighbors k. Overall, ENaN performed the best in terms of AUC and G-mean. As k increased, ENaN maintained the best results. This indicates the stability of ENaNSMOTE regardless of the number of neighbors of the kNN classifier. Table 4 and Fig. 6 confirm that ENaN performed best, followed by 5NN, and NaN produced worse results. Hence, the outstanding performance obtained using ENaNSMOTE to process imbalanced datasets is consistent with respect to different classifiers (decision tree and kNN).

Because of the page limit, Fig. 7 shows scatter plots of only two synthetic

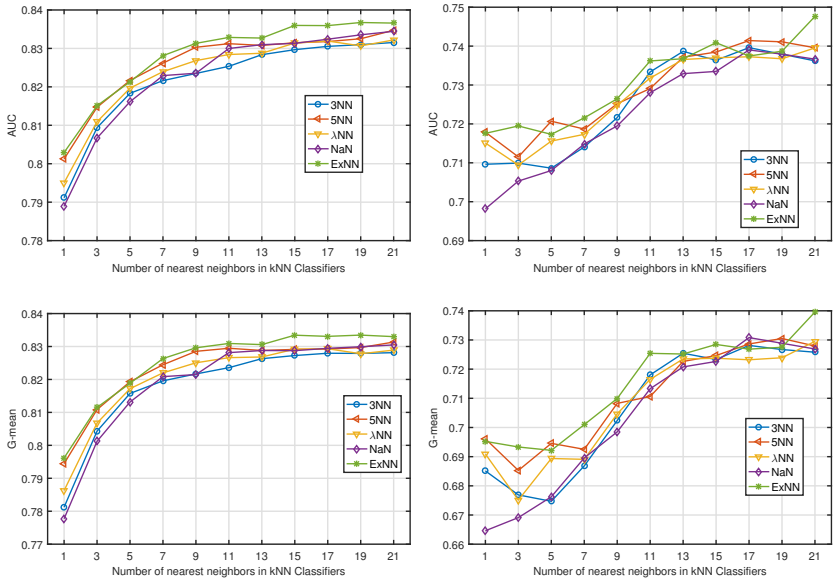
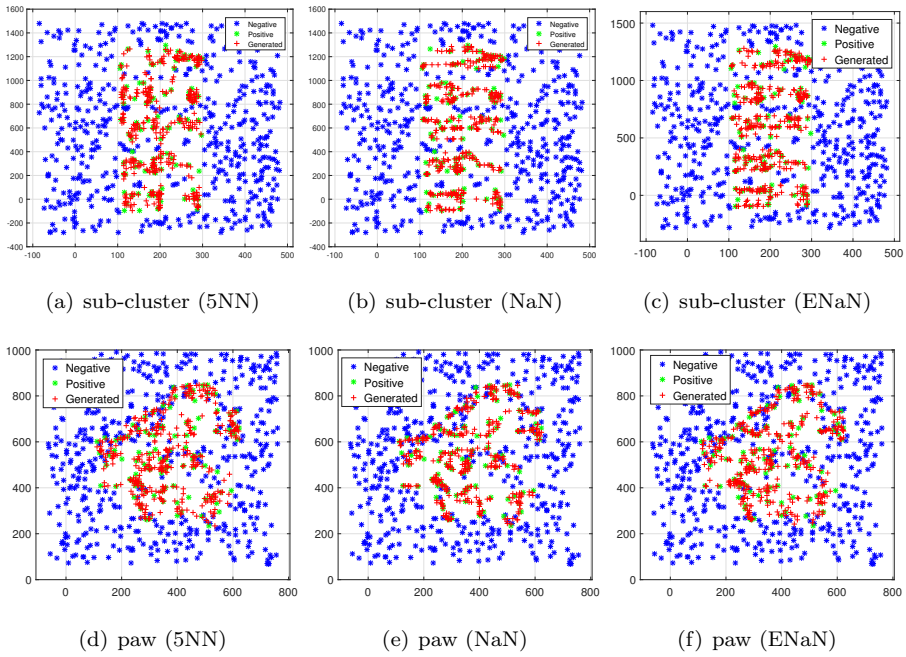


Figure 6 AUC and G-mean results using kNN classifiers on synthetic (left) and real (right) datasets.

datasets after using SMOTE (k=5), NaNSMOTE, and ENaNSMOTE for oversampling. Figures 7 (a)–(c) show the scatter plots of the *sub-cluster* dataset with $DR = 0\%$; Figs. 7 (d)–(f) present the scatter plots of the *paw* dataset with $DR = 70\%$. Comparing the two groups of figures, we can see that the samples generated using ENaNSMOTE fit the true distribution better than those generated using SMOTE and NaNSMOTE. NaNSMOTE generated fewer borderline examples because NaN is the intersection of λ-nearest neighbors and reverse λ-nearest neighbors. In contrast, ENaNSMOTE yielded a larger number of effective border samples because ENaN is the union of λ-nearest neighbors and reverse λ-nearest neighbors. This facilitated a more expanded local structure and helped to generate high-quality examples.

In addition, we explored the distribution of four types of examples (safe, borderline, rare, and outlying) in real datasets according to the method proposed in (Napierala and Stefanowski (2016)). Owing to the page limit, Table 4 lists the distributions of four types of examples in the minority class of the



(a) sub-cluster (5NN) (b) sub-cluster (NaN) (c) sub-cluster (ENaN)

(d) paw (5NN) (e) paw (NaN) (f) paw (ENaN)

Figure 7 Sample distributions of two synthetic datasets (*sub-cluster* with DR=0% and *paw* with DR=70%) after using SMOTE (k=5), NaNSMOTE, ENaNSMOTE.

seven original (None) and processed (using NaNSMOTE and ENaNSMOTE) real datasets. ENaNSMOTE tended to generate more borderline examples, whereas NaNSMOTE yielded more safety examples. Interestingly, the number of minority examples obtained by NaNSMOTE was smaller than that obtained by ENaNSMOTE in some datasets, such as *satimage* and *yeast*. This is because in these datasets, some instances had no NaNs and, therefore, generated no new instances. This violates our assumption that minority class examples carry valuable information, given their scarcity. Therefore, it is necessary to oversample as many positive instances as possible, just as ENaNSMOTE does.

Figure 8 shows the scatter plots of two real datasets after NaNSMOTE and ENaNSMOTE were used. Because these real datasets have high-dimensional features, we used t-SNE (Van der Maaten and Hinton (2008)) to reduce them to two dimensions for visualization. The t-SNE technique can retain the local

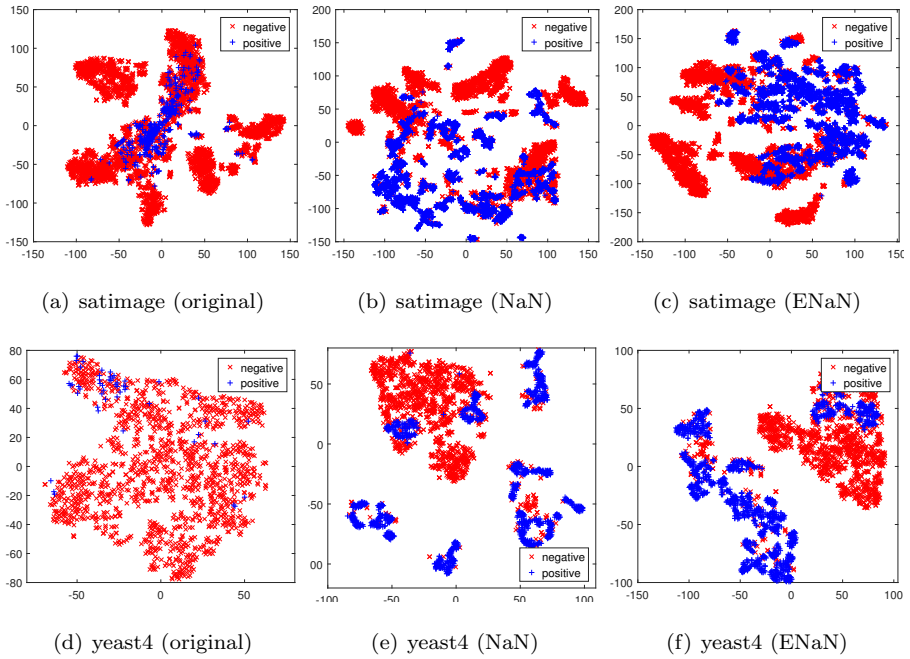
Table 4 Four types of examples in the minority class of seven real datasets.

		Safe		Borderline		Rare		Outlying	
Dataset	Neighbor	Num	Ratio	Num	Ratio	Num	Ratio	Num	Ratio
satimage	None	197	47.47	135	32.53	49	11.81	34	8.19
	NaN	3915	**99.80**	1	0.03	1	0.03	6	0.15
	ENaN	**3998**	99.45	**22**	**0.55**	0	0.00	0	0.00
car-good	None	33	47.83	27	39.13	6	8.70	3	4.35
	NaN	**1653**	**99.64**	6	0.36	0	0.00	0	0.00
	ENaN	1613	97.23	**46**	**2.77**	0	0.00	0	0.00
breastcancer	None	16	19.75	31	38.27	21	25.93	13	16.05
	NaN	**114**	**59.07**	59	30.57	14	7.25	6	3.11
	ENaN	95	48.47	**79**	**40.31**	18	9.18	4	2.04
cmc	None	46	13.81	142	42.64	83	24.92	62	18.62
	NaN	**785**	**69.41**	280	24.76	41	3.63	25	2.21
	ENaN	712	62.46	**358**	**31.40**	42	3.68	28	2.46
abalone	None	39	11.64	119	35.52	77	22.99	100	29.85
	NaN	**3430**	**90.29**	303	7.98	47	1.24	19	0.50
	ENaN	3249	84.57	**454**	**11.82**	93	2.42	46	1.20
yeast4	None	2	3.92	19	37.25	10	19.61	20	39.22
	NaN	1287	**99.23**	6	0.46	0	0.00	4	0.31
	ENaN	**1403**	97.91	**26**	**1.81**	4	0.28	0	0.00
balance	None	0	0.00	0	0.00	4	8.16	45	91.84
	NaN	**556**	**96.53**	20	3.47	0	0.00	0	0.00
	ENaN	508	88.19	**66**	**11.46**	1	0.17	1	0.17

structure of the data while revealing the global structure, such as clusters. To an extent, using ENaN obtained more expanded spaces of the minority class than using NaN.

4.3.2. Validation of ENaN in SMOTE's variants

In this section, we used five popular or state-of-the-art SMOTE variants to validate the improvement of ENaN compared to kNN. The five SMOTE variants and their primary settings are shown in Table 5. The first two are informed oversampling methods and the last three are hybrid sampling methods. Their hyperparameters were obtained empirically or following the settings in the original papers. According to Table 3, SMOTE with 5NN exhibited comparable performance; therefore, $k\_SMOTE = 5$ was set in these SMOTE's variants. We



(a) satimage (original) (b) satimage (NaN) (c) satimage (ENaN)

(d) yeast4 (original) (e) yeast4 (NaN) (f) yeast4 (ENaN)

Figure 8 T-SNE visualization of two real datasets before (original) and after using NaNSMOTE and ENaNSMOTE.

Table 5 Primary settings of SMOTE's variants.

Method	**Primary Settings**
BL-SMOTE (BL-SM)	$k\_SMOTE = 5, k\_border = 9$
ADASYN (ADS)	$k\_SMOTE = 5, k\_density = 5$
SMOTE-ENN (SM-ENN)	$k\_SMOTE = 5, k\_ENN = 3$
SMOTE-WENN (SM-WENN)	$k\_SMOTE = 5, k\_WENN = 3$
SMOTE-RkNN (SM-RkNN)	$k\_SMOTE = 5, \lambda = 2, k\_RkNN = \sqrt{n}$

replaced kNN with ENaN in SMOTE and compared the conventional SMOTE variants and their ENaN versions.

Table 6 shows the comparison results obtained using the decision tree on synthetic and real datasets. The Wilcoxon signed-rank test was used to compare each conventional SMOTE variant with its ENaN-replaced version. An asterisk indicates a significant difference ($p-$value < 0.1) between the compared methods. In most cases, the ENaN versions of the SMOTE variants achieved

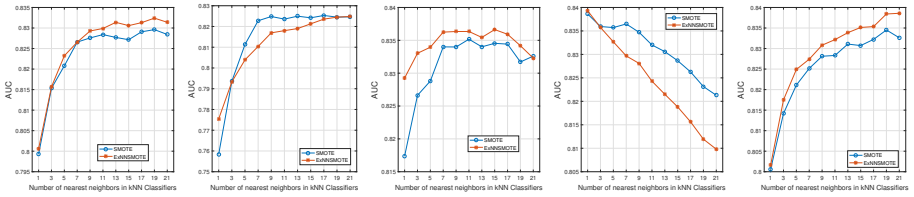
Table 6 Results of comparing kNN and ENaN in SMOTE variants obtained by decision tree on synthetic and real datasets.

Data	Matric	Neigh.	BL-SM	ADS	SM-ENN	SM-WENN	SM-RkNN
Syn.	AUC	kNN	0.7967	0.7663	0.8164	0.8363	0.7955
		ENaN	**0.8018***	**0.7798***	**0.8283***	**0.8372**	**0.8048***
	G-mean	kNN	0.7879	0.7440	0.8133	**0.8327**	0.7872
		ENaN	**0.7943**	**0.7635***	**0.8261***	0.8307	**0.7982***
Real	AUC	kNN	0.6875	0.6823	0.7182	0.7412	0.6946
		ENaN	**0.6941**	**0.6867**	**0.7511***	**0.7430**	**0.6974**
	G-mean	kNN	0.6230	**0.6098**	0.6779	0.7262	**0.6384**
		ENaN	**0.6412***	0.5977	**0.7299***	**0.7275**	0.6376

better results than the conventional kNN versions. A significant difference was observed on the synthetic datasets.

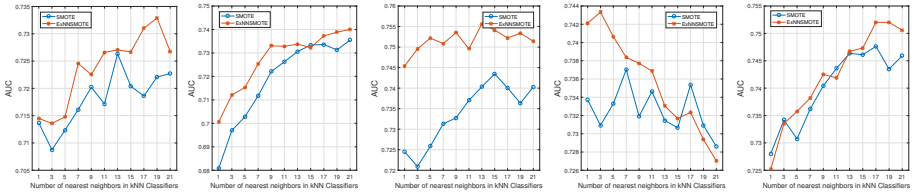
Figures 9 and 10 show the AUC results obtained using kNN classifiers with $k \in \{1, 2, \ldots, 21\}$ on synthetic and real datasets, respectively. The ENaNS-MOTE variants achieved a continuous improvement over the SMOTE variants as the number of neighbors in the kNN classifiers increased, with the exception of SMOTE-WENN. We try to explain the reason. SMOTE-WENN is our previously proposed method, which uses weighted ENN rule to detect and delete noisy examples after SMOTE. On the one hand, SMOTE-WENN aims to preserve as many minority class examples as possible in the borderline areas. This facilitates the improvement of the performance of the minority class. On the other hand, ENaN obtains the nearest neighbors based on the sample distribution of the minority class without considering the majority class. Hence, in the borderline areas, the samples generated by ENaNSMOTE may cause noise. According to the noise filtering method WENN, more majority class examples are removed (shown in Fig. 11), leading to reduced performance of the majority class. This results in imbalanced recognition rates of both classes (i.e., sensitivity and specificity), so AUC and G-mean reduce.

In summary, ENaN obtained significantly better results than kNN and NaN in most cases. Even if ENaN did not perform significantly better than kNN,



(a) BL-SM (b) ADS (c) SM-ENN (d) SM-WENN (e) SM-RkNN

Figure 9 AUC results of comparing kNN and ENaN in SMOTE variants on synthetic datasets.



(a) BL-SM (b) ADS (c) SM-ENN (d) SM-WENN (e) SM-RkNN

Figure 10 AUC results of comparing kNN and ENaN in SMOTE variants on real datasets.

ENaN was comparable and has an advantage of without parameter k over kNN.

5. Conclusions and future work

In this study, we propose a new neighborhood concept ENaN for SMOTE and its variants to deal with imbalanced data. The experimental results demonstrated that ENaN performs better than kNN and NaN in SMOTE and its variants. The main advantages of ENaN include three folds. First, ENaN determines nearest neighbors adaptively according to the sample distribution, which can capture the data structure precisely. Second, ENaN explores broad local neighborhoods for minority class examples, particularly for rare cases and outliers. This is beneficial for generating effective and high-quality examples and for improving the sample distribution. Third, ENaN does not require to set the number of neighbors, which makes it easy to use and hence enhances its applicability.

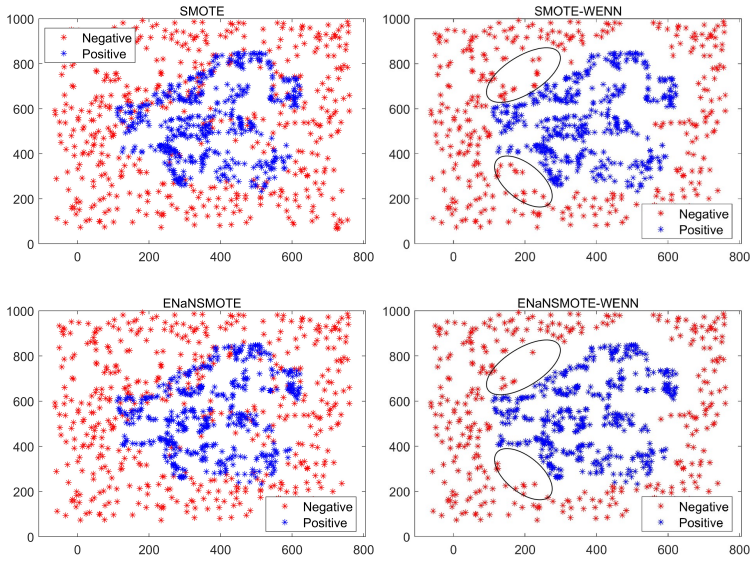


Figure 11 Take synthetic dataset *paw* with $DR = 70\%$ for example to show the sample distributions after using SMOTE, SMOTE-WENN, NaNSMOTE, and NaNSMOTE-WENN.

The proposed ENaN has a limitation of determining nearest neighbors only in the minority class without considering the distribution of the majority class. This limitation may result in reduced performance of the majority class when there is a large degree of overlap between classes. This is illustrated by the worse results of ENaN in SMOTE-WENN compared to original SMOTE-WENN. In the future, we expect to improve ENaN by considering the distributions of both classes.

Acknowledgments

This work was supported by the Natural Science Foundation of Shandong Province (ZR2021QF059); the Science, Education and Industry Integration Pilot Project of Qilu University of Technology (Shandong Academy of Sciences) (2022PX097); the National Natural Science Foundation of China (62076143 and 61906104); the Natural Science Foundation of Shandong Province (ZR2021MF090 and ZR2019BF018); and the Teaching and Research Project of Qilu University

of Technology (Shandong Academy of Sciences) (2021yb62). We would like to thank Editage (www.editage.cn) for English language editing.

References

Batista, G.E.A.P.A., Prati, R.C., Monard, M.C., 2004. A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter 6, 20–29.

Bian, Z., Vong, C.M., Wong, P.K., Wang, S., 2022. Fuzzy knn method with adaptive nearest neighbors. IEEE Transactions on Cybernetics 52, 5380–5393.

Błaszczyński, J., Stefanowski, J., 2015. Neighbourhood sampling in bagging for imbalanced data. Neurocomputing 150, 529–542.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16, 321–357.

Cheng, D., Zhu, Q., Huang, J., Wu, Q., Yang, L., 2019. A novel cluster validity index based on local cores. IEEE Transactions on Neural Networks and Learning Systems 30, 985–999.

Cheng, D., Zhu, Q., Huang, J., Yang, L., Wu, Q., 2017. Natural neighbor-based clustering algorithm with local representatives. Knowledge-Based Systems 123, 238–253.

Du, G., Zhang, J., Luo, Z., Ma, F., Ma, L., Li, S., 2020. Joint imbalanced classification and feature selection for hospital readmissions. Knowledge-Based Systems 200, 106020.

Faisal, S., Tutz, G., 2022. Nearest neighbor imputation for categorical data by weighting of attributes. Information Sciences 592, 306–319.

Fernandes, E.R.Q., de Carvalho, A.C.P.L.F., Yao, X., 2020. Ensemble of classifiers based on multiobjective genetic sampling for imbalanced data. IEEE Transactions on Knowledge and Data Engineering 32, 1104–1115.

Friedman, M., 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the American Statistical Association 32, 675–701.

García, S., Fernández, A., Luengo, J., Herrera, F., 2009. A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. Soft Computing 13, 959–977.

Guan, H., Zhang, Y., Cheng, H., Xian, M., Tang, X., 2019. BA2Cs: Bounded abstaining with two constraints of reject rates in binary classification. Neurocomputing 357, 125–134.

Guan, H., Zhang, Y., Ma, B., Li, J., Wang, C., 2021a. A generalized optimization embedded framework of undersampling ensembles for imbalanced classification, in: 2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA), pp. 1–10.

Guan, H., Zhang, Y., Xian, M., Cheng, H., Xianglong, T., 2021b. SMOTE-WENN: Solving class imbalance and small sample problems by oversampling and distance scaling. Applied Intelligence 51, 1394–1409.

Han, H., Wang, W., Mao, B., 2005. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning, in: International Conference on Intelligent Computing, Springer, Berlin, Heidelberg. pp. 878–887.

He, H., Bai, Y., Garcia, E.A., Li, S., 2008. ADASYN: Adaptive synthetic sampling approach for imbalanced learning, in: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), IEEE. pp. 1322–1328.

Huang, J., Zhu, Q., Yang, L., Cheng, D., Wu, Q., 2017. A novel outlier cluster detection algorithm without top-n parameter. Knowledge-Based Systems 121, 32–40.

Jiang, L., Li, C., Wang, S., 2014. Cost-sensitive bayesian network classifiers. Pattern Recognition Letters 45, 211–216.

Jiang, L., Qiu, C., Li, C., 2015. A novel minority cloning technique for cost-sensitive learning. International Journal of Pattern Recognition and Artificial Intelligence 29, 1551004.

Kahraman, H.T., 2016. A novel and powerful hybrid classifier method: Development and testing of heuristic k-nn algorithm with fuzzy distance metric. Data & Knowledge Engineering 103, 44–59.

Li, J., Zhu, Q., Wu, Q., 2019. A self-training method based on density peaks and an extended parameter-free local noise filter for k nearest neighbor. Knowledge-Based Systems 184, 104895.

Li, J., Zhu, Q., Wu, Q., Fan, Z., 2021. A novel oversampling technique for class-imbalanced learning based on SMOTE and natural neighbors. Information Sciences 565, 438–455.

Liu, X.Y., Wu, J., Zhou, Z.H., 2009. Exploratory undersampling for class-imbalance learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 39, 539–550.

Liu, Z., Cao, W., Gao, Z., Bian, J., Chen, H., Chang, Y., Liu, T.Y., 2020. Self-paced ensemble for highly imbalanced massive data classification, in: 2020 IEEE 36th International Conference on Data Engineering (ICDE), IEEE. pp. 841–852.

López, V., Fernández, A., García, S., Palade, V., Herrera, F., 2013. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Information Sciences 250, 113–141.

Luo, J., Qiao, H., Zhang, B., 2021. A minimax probability machine for nondecomposable performance measures. IEEE Transactions on Neural Networks and Learning Systems , 1–13.

Van der Maaten, L., Hinton, G., 2008. Visualizing data using t-sne. Journal of Machine Learning Research 9.

Mostafaei, S., Ahmadi, A., Shahrabi, J., 2022. Dealing with data intrinsic difficulties by learning an interpretable ensemble rule learning (PERL) model. Information Sciences 595, 294–312.

Napierala, K., Stefanowski, J., 2016. Types of minority class examples and their influence on learning classifiers from imbalanced data. Journal of Intelligent Information Systems 46, 563–597.

Napierała, K., Stefanowski, J., Wilk, S., 2010. Learning from imbalanced data in presence of noisy and borderline examples, in: International Conference on Rough Sets and Current Trends in Computing, Springer, Berlin, Heidelberg. pp. 158–167.

Radovanović, M., Nanopoulos, A., Ivanović, M., 2015. Reverse nearest neighbors in unsupervised distance-based outlier detection. IEEE Transactions on Knowledge and Data Engineering 27, 1369–1382.

Rosner, B., Glynn, R.J., Lee, M.L.T., 2006. The wilcoxon signed rank test for paired comparisons of clustered data. Biometrics 62, 185–192.

Sadhukhan, P., Palit, S., 2019. Reverse-nearest neighborhood based oversampling for imbalanced, multi-label datasets. Pattern Recognition Letters 125, 813–820.

Sáez, J.A., Luengo, J., Stefanowski, J., Herrera, F., 2015. SMOTE-IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. Information Sciences 291, 184–203.

Seng, Z., Kareem, S.A., Varathan, K.D., 2021. A neighborhood undersampling stacked ensemble (NUS-SE) in imbalanced classification. Expert Systems with Applications 168, 114246.

Stefanowski, J., 2013. Overlapping, rare examples and class decomposition in learning classifiers from imbalanced data. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 277–306.

Tian, Y., Bian, B., Tang, X., Zhou, J., 2021. A new non-kernel quadratic surface approach for imbalanced data classification in online credit scoring. Information Sciences 563, 150–165.

Wang, C., Xin, C., Xu, Z., 2021a. A novel deep metric learning model for imbalanced fault diagnosis and toward open-set classification. Knowledge-Based Systems 220, 106925.

Wang, G., Wong, K.W., Lu, J., 2021b. AUC-based extreme learning machines for supervised and semi-supervised imbalanced classification. IEEE Transactions on Systems, Man, and Cybernetics: Systems 51, 7919–7930.

Xie, Y., Qiu, M., Zhang, H., Peng, L., Chen, Z., 2022. Gaussian distribution based oversampling for imbalanced data classification. IEEE Transactions on Knowledge and Data Engineering 34, 667–679.

Xu, Y., Yu, Z., Chen, C.L.P., Liu, Z., 2021. Adaptive subspace optimization ensemble method for high-dimensional imbalanced data classification. IEEE Transactions on Neural Networks and Learning Systems , 1–14.

Yang, K., Yu, Z., Wen, X., Cao, W., Chen, C.L.P., Wong, H.S., You, J., 2020. Hybrid classifier ensemble for imbalanced data. IEEE Transactions on Neural Networks and Learning Systems 31, 1387–1400.

Yang, L., Zhu, Q., Huang, J., Cheng, D., Wu, Q., Hong, X., 2018. Natural neighborhood graph-based instance reduction algorithm without parameters. Applied Soft Computing 70, 279–287.

Zhang, A., Yu, H., Huan, Z., Yang, X., Zheng, S., Gao, S., 2022. SMOTE-RkNN: A hybrid re-sampling method based on smote and reverse k-nearest neighbors. Information Sciences 595, 70–88.

Zhao, S., Li, J., 2020. ELS: A fast parameter-free edition algorithm with natural neighbors-based local sets for k nearest neighbor. IEEE Access 8, 123773–123782.

Zhu, Q., Feng, J., Huang, J., 2016. Natural neighbor: A self-adaptive neighborhood method without parameter k. Pattern Recognition Letters 80, 30–36.