




Noise-Robust Gaussian Distribution Based Imbalanced Oversampling

Xuetao Shao and Yuanting Yan (✉) 

Artificial Intelligence Institute, School of Computer Science and Technology,
Anhui University, Hefei 230601, Anhui, People's Republic of China
ytyan@ahu.edu.cn

Abstract. Imbalanced data classification has become one of the hot topics in the field of data mining and machine learning. Oversampling is one of the mainstream methods to solve the imbalance problem by synthesizing new samples to balance the data distribution. However, due to the limited sample local information, the data synthetic process is risky in deteriorating the class overlap phenomenon, showing a vulnerable robustness with respect to data noise. In this paper, we propose a noise robust gaussian distribution based imbalanced oversampling (NGOS). NGOS first determines the neighborhood radius based on the global information, and then assigns sampling weights to minority class samples based on the density and the distance information within each of the neighborhoods. Finally, NGOS generates new samples with a Gaussian distribution model. We validate the effectiveness of our proposed method on the 38 KEEL datasets, DT classifier and eleven comparison methods. Experimental results show that our method outperforms the other compared methods in terms of Fmeasure, AUC, Gmean. The codes of NGOS are released in <https://github.com/ytyanncp/NGOS>.

Keywords: Imbalanced data classification · Oversampling · Noise · Gaussian distribution

1 Introduction

Imbalanced data classification has become a hot topic in the fields of data mining and machine learning. Data imbalance poses a great challenge to the robustness of traditional classification algorithms. And they are widespread in fields such as fraud detection [12], network intrusion monitoring [17], software defect prediction [6]. Researchers have proposed a variety of methods for learning imbalanced data, which can be roughly divided into two categories: data-level methods, algorithm-level methods [8]. Algorithm-level methods mainly adapt classifiers specifically designed for imbalanced data or improve traditional classifiers to

This work was supported in part by the National Natural Science Foundation of China under Grant 62376002.

make them suitable for imbalanced data. Data-level methods mainly resample the imbalanced dataset by adding minority class samples (oversampling) or removing majority class samples (undersampling) to balance the dataset.

Data-level methods have become the mainstream method for solving imbalanced data classification problems due to their simplicity, efficiency, and independence of subsequent classifiers [16]. Data-level methods are mainly divided into undersampling, oversampling, and hybrid sampling [8]. Undersampling achieves balance by removing some of the majority class samples. Oversampling synthesizes minority class samples to balance the dataset. Hybrid sampling combines the above two strategies to achieve better learning results. Recent studies have shown that oversampling methods are significantly superior to undersampling methods on traditional classifiers because they provide a higher proportion of safe samples while reducing the proportion of non-safe samples [7].

SMOTE [4] is the most classic oversampling method, but its mechanism of randomly selecting the nearest neighbors of minority class samples for linear interpolation to generate new samples ignores the distribution information of the samples. To address the shortcomings of SMOTE, researchers have proposed many oversampling methods in recent years. These include Borderline-SMOTE [9], which emphasizes synthesizing samples in the boundary region, Safe-Level-SMOTE [3], which emphasize synthesizing samples in the safe region. Unlike the above-mentioned method that only utilizes minority class information, GDO [20] utilizes the density and distance information of both the majority and minority classes to weight the minority class samples, simultaneously. However, these methods either overemphasize synthesizing samples in specific areas, leading to overfitting, or overemphasize preserving the original data distribution and ignore the adverse effects of noisy samples on the classification model.

To address these problems, this paper proposes a noise-robust gaussian distribution based imbalanced oversampling (NGOS). NGOS uses an adaptive neighborhood determination method to mine sample neighborhood information and introduces information entropy to measure the uncertainty of different sample distributions within the neighborhood to reduce the sampling rate of highly overlapping samples (even noise samples) and reduce the risk of introducing additional class overlap and noise samples. To avoid oversampling of the minority class being too concentrated in the boundary region, the method combines the distance information between the minority and majority classes in the neighborhood to expand the potential space for synthesizing minority class samples.

The main contributions of this paper are summarized as follows:

- A noise-robust oversampling method (NGOS) based on Gaussian distribution for imbalanced data is proposed.
- NGOS enhances the robustness of the minority oversampling model to noise by introducing a fixed neighborhood information mining method and information entropy, and reduces the risk of introducing additional class overlap and noise samples by reducing the sampling rate of highly overlapping samples (even noise samples).

- NGOS expands the potential space for synthesizing minority class samples by combining the distance information between the minority and majority classes in the neighborhood, and properly synthesizes new samples in the safe region to avoid overfitting problems in the boundary region.
- We evaluate the performance of NGOS on 38 KEEL datasets by comparing it with 11 data-level methods. The experimental results show that we achieved the best performance in terms of Fmeasure, AUC, and Gmean.

The rest of this paper is organized as follows: Sect. 2 introduces related work and the GDO algorithm. Section 3 proposes the NGOS algorithm. Section 4 presents experimental comparisons and analyses. Section 5 concludes the paper.

2 Related Work

2.1 Resampling Methods

Undersampling methods balance the dataset by removing some majority class samples. SDUS [22] uses a supervised constructive process to learn majority-class local patterns in terms of sphere neighborhoods (SPN) to maintain the distribution pattern of original data in selecting majority-class sample subsets from different perspectives. RUS [2] randomly removes majority class samples to balance the dataset. It may discard important information. Tomek [11] links identify Tomek pairs where a minority class sample and a majority class sample are mutual nearest neighbors, and remove the majority class samples. ENN [19] removes majority class samples that have mostly minority class samples among their k nearest neighbors. However, they do not explicitly specify the number of samples for removing which may lead to undesired level of data imbalance. CC [15] first clusters the minority class samples and then selects either the centroid or the majority class sample closest to the centroid of each cluster. In addition, RBU [13] performs undersampling by calculating inter-class potentials, which reflect the amount of information contained in the majority class. However, it requires iterative steps, making it slower.

Oversampling methods balance the dataset by synthesizing minority class samples. SMOTE [4] synthesizes minority class samples by randomly selecting seed samples and applying linear interpolation. It may generate a large number of new noisy samples. To address this, researchers started to restrict the selection of seed samples in SMOTE. Borderline-SMOTE [9] confines the seed sample selection to the boundary region, considering samples at the classification boundary more difficult to classify. In contrast, Safe-Level-SMOTE [3] argues that minority class samples located in safe regions are better suited as seed samples, because synthesizing samples in the boundary region is more likely to introduce noisy and overlapping samples. ADASYN [10] adaptively assigns weights for seed sample selection based on the density of sample distributions. MWMOTE [1] combines location and density factors and integrates data clustering to assign weights for

minority class samples. These methods are all derived from SMOTE [4], and their synthesis methods use linear interpolation. Therefore, overgeneralization issues may arise during sample synthesis. GDO [20] samples and proposes a new sample synthesis method based on Gaussian models. As mentioned above, it overly emphasizes majority class samples in the weighted selection of seed samples, which may result in the synthesis of noisy samples.

Hybrid sampling methods balance the dataset by combining oversampling methods and undersampling methods, which combines the advantages of both. Most of these methods use SMOTE [4] as the main oversampling process and then combine it with different undersampling methods to balance the dataset. SMOTE+TL [2] and SMOTE+ENN [2] combine SMOTE with Tomek links and ENN, respectively. They first use SMOTE to oversample, and then use Tomek links and ENN for undersampling. However, using SMOTE for synthesis can lead to overgeneralization issues. LDAS [21], which is different from the traditional oversampling-then-undersampling process mentioned above, first cleans the overlap area using undersampling methods, and then synthesizes minority class samples using oversampling methods.

2.2 Gaussian Distribution Based Oversampling (GDO)

GDO [20] believes that different minority classes carry different information, so it considers both density and distance information to assign different weights for selecting seed samples to different minority classes.

The sample selection weight factor of GDO is shown in Eq. (1). Where $C(x_i)$ represents the proportion of majority class samples in the K nearest neighbors, and $D(x_i)$ represents the proportion of the distance between majority class samples and the total distance in the K nearest neighbors.

$$I(x_i) = C(x_i) + D(x_i) \quad (1)$$

Then, the weights are normalized as shown in Eq. (2):

$$\widehat{I}(x_i) = \frac{I(x_i)}{\sum_{i=1}^{|N^{min}|} I(x_i)} \quad (2)$$

Where $|N^{min}|$ represents the number of minority class samples.

Let o be the origin of the coordinates, and for any seed sample x_i , a random vector $\vec{o\hat{v}}$ is generated. Then, $\vec{x_i\hat{v}}$ is the direction vector, and the newly synthesized sample point is on this direction.

$$\vec{x_i\hat{v}} = \vec{o\hat{v}} - \vec{o\hat{x}_i} \quad (3)$$

Next, the length of vector $\vec{x_i\hat{x}'}$ is determined, which follows the Gaussian distribution:

$$|\vec{x_i\hat{x}'}| = d_i \sim N(\mu_i, \alpha\sigma_i) \quad (4)$$

Where $\mu_i = 0$ and σ_i is the Euclidean distance between the seed sample x_i and its nearest same-class sample.

Therefore, the vector form of the newly synthesized sample is:

$$\vec{ox'} = \vec{ox_i} + \frac{|\vec{x_i x'}|}{|\vec{x_i v}|} \cdot \vec{x_i v} \quad (5)$$

3 Proposed Method

3.1 Analysis of the GDO Algorithm

The GDO algorithm relies on K-nearest neighbor (KNN) calculation to obtain local distribution information. However, in imbalanced data, the majority class is the dominant one in the sample space. Therefore, the decision process based on K is prone to bias towards the majority class, and the method has poor robustness to noise. In addition, to achieve better performance for different data distributions, it is usually necessary to find suitable parameter K, which makes the algorithm less adaptable.

As shown in Fig. 1, when the parameter K in the K-nearest neighbor calculation is set to 5, the weight of sample A calculated by Eq. (1) is 2 (each sample can obtain the maximum weight value). Therefore, sample A has the highest probability of being selected as the seed sample, but synthesizing minority samples based on sample A as the seed sample will further increase the difficulty of classification. In addition, GDO believes that samples in the safe region are easier to identify and ignores these samples. This causes the sampling process to concentrate too much on the boundary area, which may cause overfitting.

As shown in Fig. 1, both B and C have a weight of 0 (they will not be selected as seed samples). However, compared with C, it can be seen that sample B is clearly closer to the decision boundary. Synthesizing samples based on B can strengthen the classification boundary to a certain extent and expand the potential generation space of synthesized samples, which can avoid the potential overfitting problem caused by synthesizing too many samples in the boundary area and improve the subsequent learning performance.

To address the above issues, this paper proposes an improved Gaussian sampling method, NGOS, which uses a fixed-radius neighborhood partition method and an information entropy-based neighborhood information measurement method to enhance the performance of imbalance learning.

3.2 Local Information of Samples

The KNN method measures the local distribution information of samples by finding their K-nearest neighbors. As shown in Fig. 1(a), this method cannot effectively characterize the differences in sample distributions, as it only considers the relationship between the target sample and its nearest neighbors, ignoring the local distribution information of its neighbors. To address this issue, this

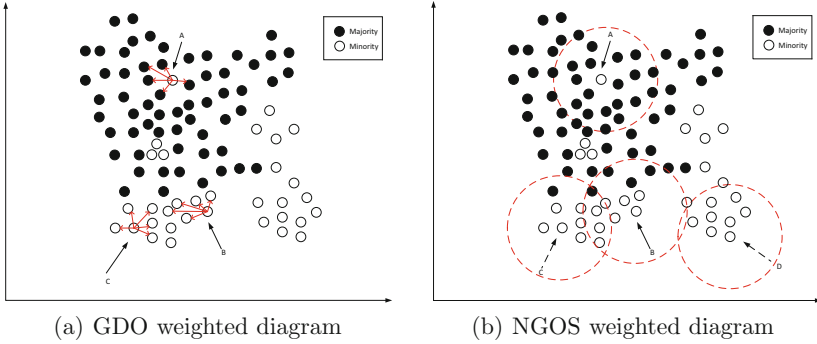


Fig. 1. Comparison of the weighting schemes between GDO and NGOS

paper utilizes the global information of sample distributions to achieve adaptive determination of neighborhoods.

$$R = 2 \sum_{x_i, x_j \in X_{train}} \frac{dist(x_i, x_j)}{(|N_{train}|(|N_{train}| - 1))} \tag{6}$$

Where $|N_{train}|$ is the number of samples in the training set, and $dist(x_i, x_j)$ is the Euclidean distance between sample x_i and x_j . It can be seen that this radius R considers the global distribution information of the samples. With R as the radius and the target sample x_{min} as the center, we obtain a subset X_{candi} of all samples whose distance to x_{min} is less than R .

$$X_{candi} = \{x_p \in X_{train}, dist(x_{min}, x_p) < R\} \tag{7}$$

Based on X_{candi} , it is easy to evaluate the data distribution within the sample neighborhood.

3.3 Estimation of Weight for Minority Class Sample Selection

In classification tasks, samples located in different regions have different impacts on the classification model [18]. To characterize the impact of different samples on the classification model, GDO uses the KNN method to characterize the importance of minority class samples. Although this method assigns high weights to boundary samples, it also overly emphasizes minority class samples located in dense majority class areas, which leads to the synthesis of a large number of potential noise samples. In other words, the GDO is not robust to noise samples. To address this issue, this paper uses information entropy [5] and distribution density to measure the samples and enhance the robustness of the model to noise.

Specifically, we first measure the distribution differences within the local neighborhood of a sample using the Eq. (8):

$$p_i = \frac{N_{i_{candi}}^{maj}}{|N_{i_{candi}}^{maj}| + |N_{i_{candi}}^{min}|} \tag{8}$$

Where $|N_{i_{candi}}^{min}|$ represents the number of minority class samples in the candidate set of sample x_i , and $|N_{i_{candi}}^{maj}|$ represents the number of majority class samples in the candidate set of sample x_i .

$$E(x_i) = -p_i \log_2 p_i - (1 - p_i) \log_2 (1 - p_i) \quad (9)$$

From Eq. (9), it can be seen that when $p_i = 1/2$, Eq. (9) obtains the maximum value of 1, and gradually decreases as p_i decreases or increases. When $p_i = 0$ or 1, we set the value to 0. In other words, the closer a sample is to the decision boundary, the greater its weight, and the further it goes into the majority class area, the smaller its weight. As shown in Fig. 1(b), all the neighboring samples of sample A are of different classes (sample A is a noise sample), and synthesizing samples at this position will increase the difficulty of training the classifier. Therefore, we set its weight to 0 according to Eq. (9). Similarly, sample D located in the safe area has neighboring samples that are all of the same class and can be easily identified by the classifier. Therefore, we also set its weight to 0.

However, using Eq. (9) alone will overly focus on samples with high uncertainty in the boundary area, in other words, it will assign higher weights to samples with higher uncertainty, which may lead to overemphasizing such samples and causing overfitting problems. Therefore, NGOS introduces distance information of the samples within the neighborhood, appropriately expands the selection range of seed samples, increases the synthesis space of potential synthesized samples, and enhances the robustness of the model. To achieve this, NGOS proposes the following distance measurement method:

$$D'(x_i) = \frac{\frac{\sum_{x_j \in X_{i_{candi}}^{maj}} dist(x_i, x_j)}{|N_{i_{candi}}^{maj}|}}{\frac{\sum_{x_j \in X_{i_{candi}}^{maj}} dist(x_i, x_j)}{|N_{i_{candi}}^{maj}|} + \frac{\sum_{x_j \in X_{i_{candi}}^{min}} dist(x_i, x_j)}{|N_{i_{candi}}^{min}|}} \quad (10)$$

When $|N_{i_{candi}}^{maj}| = 0$ or $|N_{i_{candi}}^{min}| = 0$, it means that the candidate set of the sample only contains majority class samples (such as sample A in Fig. 2) or minority class samples (such as sample D in Fig. 2). In these cases, the weight $D'(x_i)$ of the sample is set to 0. It can be seen that Eq. (10) uses distance information between samples to select seed samples for minority class synthesis that are farther away from the decision boundary for unstable samples (i.e., samples with different classes in their neighborhoods), thus avoiding overfitting.

By considering both density and distance factors, the following method is proposed to calculate the weight of each sample:

$$I(x_i) = D'(x_i) + E(x_i) \quad (11)$$

From Eq. (11), it can be seen that the weight of minority samples that are deep in the majority class area is 0, while the weight of samples located at or close to the decision boundary is relatively high.

3.4 Probabilistic Seed Sample Selection and Time Complexity

After calculating the weight of each minority sample using Eq. (11), we normalize the weights using Eq. (2) to convert them into probabilities. The selection of seed samples and generation of new samples follow the iterative process below: at each iteration, a seed sample is chosen based on its probability \hat{I} , and new minority samples are synthesized based on the chosen seed sample. This process continues until the number of minority samples is equal to the number of majority samples. The number of samples to be synthesized is determined by the Eq. (12).

$$G = |N^{maj}| - |N^{min}| \quad (12)$$

The process of NGOS is described in Algorithm 1. To calculate Eq. (9) and Eq. (10), we first need to compute the candidate set of samples x_i , which has a time complexity $O(|N_{train}|)$. Then, each minority class sample needs to be calculated, resulting in a time complexity $O(|N_{train}| |N^{min}|)$ for the minority class weighting process (lines 2–8). The data generating process (lines 14–18), the minority class instances are resampled G times and the time complexity is $O(|G|)$. Because G is smaller than N_{train} , the time complexity of Algorithm 1 is $O(|N_{train}| |N^{min}|)$.

Algorithm 1. NGOS(α)

Input: the original dataset D , scaling factor α ;

Output: balanced dataset S ;

- 1: Divide into minority class D_{min} and majority class D_{maj} ;
 - 2: for x_i in D_{min} :
 - 3: Calculate the radius R ; Eq. (6)
 - 4: Calculate the candidate set X_{candi} of x_i ; //Eq. (7)
 - 5: Obtain the density factor weight $E(x_i)$; Eqs. (8) and (9)
 - 6: Obtain the distance factor weight $D'(x_i)$; //Eq. (10)
 - 7: Calculate the information weight $I(x_i)$; //Eq. (11)
 - 8: end for
 - 9: for x_i in D_{min} :
 - 10: Calculate the normalized weight $\hat{I}(x_i)$; //Eq. (2)
 - 11: end for
 - 12: Calculate the number of samples needed for balance G ; //Eq. (12)
 - 13: Initialize the number of minority class samples to be synthesized $n = 0$;
 - 14: while $n < G$:
 - 15: Synthesizing samples with using Eqs. (3)–(5);
 - 16: Add the synthesized sample to D'_{min} ;
 - 17: $n = n + 1$;
 - 18: end while
 - 19: $S = D \cup D'_{min}$;
-

4 Experiments and Analysis

To validate the effectiveness of our proposed NGOS algorithm, we designed a three-stage experimental study. First, we will briefly introduce the evaluation metrics and settings used in our experiments. Then, we analyzed the influence of algorithm parameters on its performance. Finally, we compare our proposed method with other state-of-the-art resampling methods on the KEEL dataset.

4.1 Experimental Settings

Evaluation Metrics. We use *Fmeasure*, *Gmean*, *AUC* (the area under the ROC curve) [16] which are the most frequently used metrics in imbalance learning were applied in this study.

Datasets. Table 1 provides detailed information about the datasets, including the dataset name, the abbreviation of the dataset (Abbr), the number of attributes (Atts), the size of the dataset (Size), the number of samples in the minority class (Min), and the imbalance ratio (IR).

Classifiers. In our experiments, we use Decision Tree (DT) classifiers provided by the scikit-learn library in Python with default parameters. To ensure the correctness of the experimental results, we used 5-fold cross-validation with 10 repetitions for the training and test set split.

Comparison Methods. In our experiments, we compared our proposed NGOS algorithm with 11 other resampling methods, including SMOTE(SMO), Borderline-SMOTE (BSM), ADASYN(ADA), MWMOTE (MWO), SMOTE Tomek links (STL), SMOTE ENN (SENN), GDO, CC, ROS, RUS, RBU.

4.2 Experimental Results and Analysis

Parameter Analysis. In NGOS, when performing oversampling, the length of the synthetic minority class mode d is derived from $N(\mu_i, \alpha\sigma_i)$, where α is a scaling factor to control the sampling density of the seed sample. To investigate the influence of the parameter α on NGOS under different data distributions, we selected 10 datasets. The best value is highlighted in bold.

Table 2 shows the AUC values and their average values for 10 datasets under different parameter values for the DT classifier. The average values indicate that NGOS performs best when α is set to 1.5, with D02, D03, D05, and D12 datasets achieving the best performance at $\alpha = 1.5$. The D14, D24 and D33 datasets achieve the best performance at $\alpha = 1.4$, 6 out of 10 datasets perform best around these values. Therefore, we recommend setting the α to 1.5.

Table 1. Description of KEEL Datasets

Dataset	Abbr	Size	Atts	Min	IR	Dataset	Abbr	Size	Atts	Min	IR
abalone19	D01	4173	9	32	129.41	newthyroid1	D20	214	6	35	5.11
abalone918	D02	730	9	41	16.8	newthyroid2	D21	214	6	35	5.11
car-good	D03	1727	7	69	24.03	pb134	D22	471	11	28	15.82
car-vgood	D04	1727	7	65	25.57	page-blocks0	D23	5471	11	559	8.79
cleveland04	D05	176	14	13	12.54	pima	D24	767	9	267	1.87
dermatology6	D06	357	35	20	16.85	p86	D25	1476	11	17	85.82
e013726	D07	280	8	7	39	p97	D26	243	11	8	29.38
e01	D08	219	8	77	1.84	segment0	D27	2307	20	329	6.01
flare-F	D09	1065	12	43	23.77	s25	D28	3315	10	49	66.65
glass1	D10	213	10	76	1.8	scvc	D29	1828	10	123	13.86
glass5	D11	213	10	9	22.67	vehicle0	D30	845	19	198	3.27
haberman	D12	305	4	81	2.77	vehicle2	D31	845	19	218	2.88
iris0	D13	149	5	49	2.04	vowel0	D32	987	14	89	10.09
kgpvs	D14	1641	42	52	30.56	wr35	D33	690	12	10	68
krivb	D15	2224	42	22	100.09	wr4	D34	1598	12	53	29.15
krvkzovd	D16	2900	7	104	26.88	wisconsin	D35	682	10	239	1.85
kvkzvf	D17	2192	7	27	80.19	yeast1	D36	1483	9	429	2.46
l024567891	D18	442	8	37	10.95	yeast6	D37	1483	9	35	41.37
lnf	D19	147	19	6	23.5	zoo-3	D38	100	17	5	19

Comparison with Other Resampling Methods. This section compares NGOS with 11 resampling methods in Sect. 4.1, which include 6 oversampling methods, 3 undersampling methods, and 2 hybrid methods.

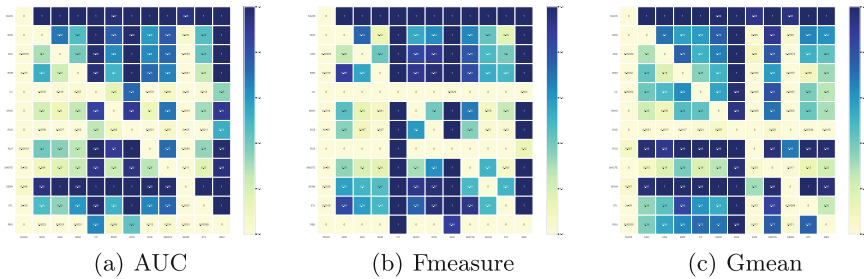
Due to space limited, we only provide the AUC for each dataset. From Table 3, it can be seen that NGOS performs the best overall compared to other comparison methods, achieving the best average values for AUC. For easily classified datasets such as D13, its evaluation metrics also reach 1, like other comparison methods. Additionally, NGOS achieves the best performance on 12 datasets for AUC. SENN achieves the best performance on 13 datasets for AUC. It can be seen that SENN is the biggest competitor of NGOS, although it achieves the best performance on one more dataset than NGOS for AUC, its overall average performance is not as good as NGOS.

Therefore, we use Bayesian analysis [14] to further compare the performance of NGOS and other comparison methods (especially SENN). Unlike other testing methods, Bayesian analysis does not fall into the pitfalls of black and white thinking and could estimate the probability that the performance of two classifiers is different(or equal). Figure 2 shows the corresponding results of Bayesian testing.

As shown in Fig. 2(a) and (c), on the DT classifier, the probability that NGOS outperforms all other comparison methods except SENN is close to 100%, and

Table 2. Influence of parameter α on DT in terms of the AUC metric

Dataset	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
D02	0.6501	0.6533	0.6788	0.6576	0.6567	0.6856	0.6460	0.6465	0.6591	0.6794	0.6623
D03	0.9286	0.9227	0.9216	0.9134	0.9137	0.9390	0.9240	0.9298	0.9347	0.9273	0.9313
D05	0.7453	0.7472	0.7686	0.7226	0.7498	0.7853	0.7313	0.7808	0.7659	0.7738	0.7746
D10	0.7305	0.7496	0.7326	0.7418	0.7448	0.7483	0.7071	0.7361	0.7379	0.7251	0.7172
D12	0.5747	0.5763	0.5971	0.5754	0.5858	0.5974	0.5842	0.5675	0.5725	0.5886	0.5876
D14	0.9882	0.9861	0.9899	0.9919	0.9957	0.9862	0.9919	0.9922	0.9942	0.9884	0.9859
D24	0.6761	0.6712	0.6746	0.6789	0.6832	0.6747	0.6729	0.6765	0.6736	0.6727	0.6709
D33	0.6475	0.6848	0.6689	0.6640	0.7177	0.6909	0.6586	0.6283	0.7072	0.6587	0.7094
D35	0.9427	0.9434	0.9376	0.9430	0.9370	0.9384	0.9370	0.9431	0.9443	0.9383	0.9422
D38	0.6593	0.6432	0.7201	0.6379	0.6622	0.6982	0.6591	0.6835	0.7043	0.6863	0.6633
Avg	0.7285	0.7341	0.7432	0.7276	0.7396	0.7506	0.7227	0.7328	0.7438	0.7375	0.7376

**Fig. 2.** The value in the i -th row and j -th column represents the probability that the i -th method performs better than the j -th method.

the probability of outperforming SENN is as high as 95%. Although NGOS has one less best dataset than SENN on AUC, its performance on all 38 datasets far exceeds SENN. From Fig. 2(b), it can be seen that the performance of NGOS on Fmeasure is very outstanding, significantly better than other comparison methods, with a probability of almost 100% of outperforming other comparison methods, including SENN, even though, SENN is a hybrid resampling method.

5 Conclusion

This paper proposed the NGOS to addresses several issues of the GDO: 1) GDO emphasizes the majority class in local regions, resulting in the generation of too many synthetic samples around minority class samples deep in the majority class region, introducing more difficult-to-learn samples that hinder the training of the learning model. 2) GDO regards that samples in safe regions are easier to recognize, thus ignoring these samples, but this can lead to oversampling being too concentrated on the boundary region, which increases the risk of overfitting. 3) Both density and distance information in the GDO method rely on the KNN

Table 3. AUC results on KEEL datasets obtained by DT

	NGOS	GDO	ADA	BSM	CC	MWO	ROS	RUS	SMO	SENN	STL	RBU
D01	0.5226	0.5220	0.5460	0.5426	0.7073	0.4944	0.5188	0.6448	0.5402	0.5816	0.5739	0.5237
D02	0.6856	0.6643	0.6879	0.6968	0.6674	0.6940	0.6111	0.6870	0.6495	0.6920	0.6616	0.6624
D03	0.9390	0.9108	0.7764	0.7994	0.8492	0.8419	0.9549	0.9567	0.8192	0.8214	0.7943	0.9660
D04	0.9858	0.9870	0.9920	0.9905	0.8629	0.9890	0.9792	0.9780	0.9935	0.9788	0.9766	0.9774
D05	0.7853	0.7533	0.7779	0.7899	0.7088	0.6471	0.6971	0.7800	0.8051	0.7746	0.7638	0.7460
D06	0.9985	0.9832	0.9835	0.9635	0.9546	0.9885	0.9835	0.9619	0.9885	0.9885	0.9885	0.9544
D07	0.8501	0.8590	0.8294	0.7387	0.8380	0.6087	0.5905	0.7059	0.8879	0.8372	0.8198	0.7205
D08	0.9758	0.9663	0.9708	0.9789	0.9732	0.9724	0.9647	0.9695	0.9668	0.9846	0.9815	0.9702
D09	0.6227	0.5887	0.5916	0.6100	0.6239	0.6556	0.6214	0.7526	0.5818	0.7564	0.6467	0.6167
D10	0.7483	0.7456	0.7186	0.7361	0.7258	0.7510	0.7385	0.7345	0.7364	0.7052	0.7365	0.7330
D11	0.9020	0.8551	0.9376	0.8476	0.9293	0.8451	0.9076	0.8507	0.8476	0.9756	0.8576	0.8388
D12	0.5971	0.5781	0.5420	0.5633	0.5281	0.5791	0.5726	0.5828	0.5297	0.6177	0.5941	0.5437
D13	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
D14	0.9976	0.9832	0.9979	0.9980	0.9978	0.9978	0.9979	0.9927	0.9978	0.9977	0.9999	0.9979
D15	1.0000	0.9736	1.0000	1.0000	0.9787	1.0000	1.0000	0.9886	1.0000	1.0000	1.0000	0.9992
D16	0.9635	0.9538	0.9446	0.9525	0.9238	0.9579	0.9463	0.9646	0.9526	0.9883	0.9586	0.9470
D17	1.0000	0.9997	0.9927	1.0000	0.9651	1.0000	1.0000	0.9877	1.0000	0.9960	0.9853	0.9982
D18	0.9043	0.9038	0.8956	0.8758	0.8619	0.8908	0.8436	0.8485	0.8929	0.8788	0.9021	0.8620
D19	0.8860	0.8173	0.7909	0.7900	0.7354	0.5379	0.6539	0.7103	0.8322	0.8250	0.8003	0.5779
D20	0.9530	0.9425	0.9710	0.9607	0.9767	0.9574	0.9355	0.9410	0.9356	0.9464	0.9516	0.9210
D21	0.9582	0.9442	0.9688	0.9682	0.9617	0.9648	0.9139	0.9506	0.9482	0.9556	0.9453	0.9413
D22	0.9977	0.9901	0.9955	0.9837	0.9114	0.9977	0.9784	0.9562	0.9977	0.9644	0.9898	0.9636
D23	0.9252	0.9138	0.9296	0.9194	0.8959	0.9236	0.9065	0.9371	0.9238	0.9378	0.9325	0.8888
D24	0.6832	0.6702	0.6756	0.6724	0.6388	0.6782	0.6739	0.6769	0.6621	0.7057	0.6690	0.6624
D25	0.9380	0.9761	0.7578	0.6379	0.5253	0.5436	0.5071	0.5999	0.7409	0.6634	0.6898	0.6021
D26	0.8188	0.6484	0.5426	0.6932	0.6020	0.4907	0.5370	0.6099	0.6041	0.5956	0.6719	0.7427
D27	0.9908	0.9883	0.9863	0.9896	0.9692	0.9887	0.9905	0.9796	0.9917	0.9885	0.9882	0.9864
D28	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9976	1.0000	1.0000	1.0000	1.0000
D29	1.0000	0.9996	1.0000	1.0000	1.0000	1.0000	1.0000	0.9998	1.0000	1.0000	1.0000	1.0000
D30	0.9316	0.9133	0.9145	0.9079	0.9329	0.9127	0.9076	0.9264	0.9064	0.9164	0.8998	0.9024
D31	0.9527	0.9434	0.9533	0.9552	0.9358	0.9472	0.9499	0.9403	0.9394	0.9428	0.9490	0.9526
D32	0.9725	0.9813	0.9564	0.9522	0.9547	0.9674	0.9223	0.9359	0.9650	0.9668	0.9618	0.9557
D33	0.7177	0.6514	0.5362	0.5216	0.6014	0.5129	0.5453	0.6339	0.4905	0.6614	0.5097	0.5103
D34	0.6500	0.5919	0.5599	0.5752	0.5887	0.5698	0.5376	0.6303	0.5808	0.6255	0.5531	0.5653
D35	0.9443	0.9392	0.9366	0.9330	0.9293	0.9363	0.9362	0.9458	0.9325	0.9480	0.9335	0.9404
D36	0.6616	0.6312	0.6654	0.6433	0.6479	0.6663	0.6493	0.6473	0.6524	0.6835	0.6616	0.6295
D37	0.8012	0.7248	0.7577	0.7502	0.7403	0.7620	0.7110	0.8075	0.7300	0.7816	0.7539	0.6260
D38	0.7201	0.6960	0.6676	0.7800	0.5032	0.7150	0.7147	0.6358	0.6045	0.6905	0.6963	0.6846
Avg	0.8679	0.8471	0.8355	0.8347	0.8196	0.8154	0.8131	0.8381	0.8323	0.8519	0.8368	0.8187

algorithm, which requires setting an appropriate K value for different datasets, reducing the adaptability of the algorithm. Experimental results on 38 KEEL datasets demonstrate that our method outperforms GDO in terms of the average rank of all evaluation metrics. Moreover, compared to other state-of-the-art resampling methods, our method also achieves the best performance.

References

1. Barua, S., Islam, M.M., Yao, X., Murase, K.: MWMOTE-majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Trans. Knowl. Data Eng.* **26**(2), 405–425 (2012)
2. Batista, G.E., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explor. Newsl.* **6**(1), 20–29 (2004)
3. Bunkhumpornpat, C., Sinapiromsaran, K., Lursinsap, C.: Safe-Level-SMOTE: safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In: Theeramunkong, T., Kijssirikul, B., Cercone, N., Ho, T.-B. (eds.) *PAKDD 2009. LNCS (LNAI)*, vol. 5476, pp. 475–482. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01307-2_43
4. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **16**, 321–357 (2002)
5. Chen, Y., Wu, K., Chen, X., Tang, C., Zhu, Q.: An entropy-based uncertainty measurement approach in neighborhood systems. *Inf. Sci.* **279**, 239–250 (2014)
6. Folino, G., Pisani, F.S., Sabatino, P.: An incremental ensemble evolved by using genetic programming to efficiently detect drifts in cyber security datasets. In: *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, pp. 1103–1110 (2016)
7. Garca, V., Sanchez, J.S., Marques, A., Florencia, R., Rivera, G.: Understanding the apparent superiority of over-sampling through an analysis of local information for class-imbalanced data. *Exp. Syst. Appl.* **158**, 113026 (2020)
8. Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., Bing, G.: Learning from class-imbalanced data: review of methods and applications. *Exp. Syst. Appl.* **73**, 220–239 (2017)
9. Han, H., Wang, W.-Y., Mao, B.-H.: Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) *ICIC 2005, Part I 1. LNCS*, vol. 3644, pp. 878–887. Springer, Heidelberg (2005). https://doi.org/10.1007/11538059_91
10. He, H., Bai, Y., Garcia, E.A., Li, S.: ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp. 1322–1328. IEEE (2008)
11. Ivan, T.: Two modifications of CNN. *IEEE Trans. Syst. Man Commun. (SMC)* **6**, 769–772 (1976)
12. Jurgovsky, J., et al.: Sequence classification for credit-card fraud detection. *Exp. Syst. Appl.* **100**, 234–245 (2018)
13. Koziarski, M.: Radial-based undersampling for imbalanced data classification. *Pattern Recogn.* **102**, 107262 (2020)
14. Krawczyk, B., Koziarski, M., Woźniak, M.: Radial-based oversampling for multi-class imbalanced data classification. *IEEE Trans. Neural Netw. Learn. Syst.* **31**(8), 2818–2831 (2019)
15. Lin, W.C., Tsai, C.F., Hu, Y.H., Jhang, J.S.: Clustering-based undersampling in class-imbalanced data. *Inf. Sci.* **409**, 17–26 (2017)
16. Lopez, V., Fernandez, A., Garca, S., Palade, V., Herrera, F.: An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics. *Inf. Sci.* **250**, 113–141 (2013)

17. Rodriguez, D., Herraiz, I., Harrison, R., Dolado, J., Riquelme, J.C.: Preliminary comparison of techniques for dealing with imbalance in software defect prediction. In: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, pp. 1–10 (2014)
18. Vuttipittayamongkol, P., Elyan, E., Petrovski, A.: On the class overlap problem in imbalanced data classification. *Knowl. Based Syst.* **212**, 106631 (2021)
19. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. *IEEE Trans. Syst. Man Cybern.* **3**, 408–421 (1972)
20. Xie, Y., Qiu, M., Zhang, H., Peng, L., Chen, Z.: Gaussian distribution based over-sampling for imbalanced data classification. *IEEE Trans. Knowl. Data Eng.* **34**(2), 667–679 (2022)
21. Yan, Y., Jiang, Y., Zheng, Z., Yu, C., Zhang, Y., Zhang, Y.: LDAS: local density-based adaptive sampling for imbalanced data classification. *Exp. Syst. Appl.* **191**, 116213 (2022)
22. Yan, Y., Zhu, Y., Liu, R., Zhang, Y., Zhang, Y., Zhang, L.: Spatial distribution-based imbalanced undersampling. *IEEE Trans. Knowl. Data Eng.* **35**, 6376–6391 (2023)