A Novel Oversampling and Feature Selection Hybrid Algorithm for Imbalanced Data Classification

Fang Feng · Kuan-Ching Li · Erfu Yang · Qingguo Zhou · Lihong Han · Amir Hussain · Mingjiang Cai

Received: date / Accepted: date

Abstract Traditional approaches tend to cause classier bias in the imbalanced data set, resulting in poor classification performance for minority classes. In particular, there are many imbalanced data in financial fraud, network intrusion, and fault detection, where recognition rate of minority classes is pertinent than the classification performance of majority classes. Therefore, there is pressure on developing efficient algorithms to solve the class imbalance problem. To this end, this article presents a novel hybrid algorithm Negative Binary General (NBG), to improve the performance of imbalanced classifications by combining oversampling and a feature selection algorithm. A novel oversampling algorithm, Negative-positive Synthetic Minority Oversampling

Fang Feng
School of Information,Guizhou University of Finance and Economics, Guiyang, Guizhou, China
School of Information Science and Engineering, Lanzhou University, Lanzhou, Gansu, China
Corresponding author: Fang Feng
E-mail: fengf15@lzu.edu.cn

Kuan-Ching Li
Department of Computer Science and Information Engineering, Providence University, Taichung, Taiwan

Erfu Yan
Department of Design, Manufacture and Engineering Management, University of Strathclyde, Glasgow G1 1XJ, UK

Qingguo Zhou
School of Information Science and Engineering, Lanzhou University, Lanzhou, Gansu, China

Lihong Han
School of Information Science and Engineering, Lanzhou University, Lanzhou, Gansu, China

Amir Hussain
School of Computing, Edinburgh Napier University, Merchiston Campus, Edinburgh EH10 5DT, Scotland, U.K.

Mingjiang Cai
Guizhou University of Finance and Economics, Guiyang, Guizhou, China

Technique (NPSMOTE), improves sample generation's practicability while the Binary Ant Lion Optimizer (BALO) algorithm extracts the most significant features to improve the classification performance. Simulation experiments carried out using seven benchmark imbalanced data sets demonstrate that, the proposed NBG algorithm significantly outperforms the classification of imbalanced small-sample data sets compared to nine other existing and six recently published algorithms.

Keywords Imbalanced data · Oversampling · Feature selection · General Vector Machine

1 Introduction

The class imbalance problem has emerged in the fields of medical diagnosis [79], financial fraud detection, network intrusion detection [42,7,37,75,84], IoT security [92], spam filtering, biological engineering [52,71], customer retention [6], among several other segments of the society [62,53]. The number of samples in different categories varies significantly on an imbalanced data set [54]. That is, the class with far more samples belongs to the majority class, while the class with a relatively small number of samples belongs to the minority class. In this article, the majority class is called the negative class, while the minority class is called the positive class.

Different quantities of samples signify providing different quantities of information to the classifier. Therefore, traditional machine learning-based classification algorithm's performance is commonly inferior in the minority class of the class imbalance problem, while the positive class often more important than the negative class in the class imbalance problem. For example, in cancer diagnosis, the entire data set is supposed to contain 100 samples. Among that 99 samples belong to the negative class (class 1), and 1 sample belongs to the positive class (class 2). If 1% of classification errors is that the cancer patient is classified as normal, this incident can cause a cancer patient's death. Furthermore, it is not easy to achieve 99% accuracy in practice, so thus, the misclassification rate of the positive class is considerably excessive.

Traditional machine learning based classification algorithms are unsuited to the class imbalance problem [87], mainly due to the following aspects: 1) the positive class samples are very few, traditional machine learning based classification algorithms are challenging to learn the characteristics of the data, 2) the data sets contain many noises. The influence of noises on both positive class and negative class is unsymmetrical. It makes imbalanced learning more difficult, and 3) The data sets from two classes are always overlapped. The importance of class imbalance problem and the limitation of traditional machine learning based classification algorithms motivate researchers to develop efficient algorithms to solve the class as mentioned above imbalance problem. Therefore, this article mainly improves the classification performance of minority class samples.

In recent decades, many algorithms have been developed to deal with the class imbalance problem, primarily based on resampling algorithm, feature selection, and exaction algorithm, cost-sensitive learning algorithm, and ensemble algorithm. Among them, the resampling algorithm is composed of oversampling algorithm, undersampling algorithm. The oversampling algorithm balances the class distribution by generating new minority class samples. In contrast, the undersampling algorithm balances the class distribution by removing the intrinsic samples in the majority class, although these algorithms may lead to the loss of important information [48, 64]. The feature selection algorithm is to select the significant features. In the class imbalance problem, minority class samples are easily treated as noises, which can be reduced by feature selection [85]. Also, feature selection can make the classifier achieve optimal performance [30]. The feature extraction algorithm is the transformation of features from high-dimensional to low-dimensional, generally creating new features [30]. A cost-sensitive learning algorithm means that negative class samples' misclassification is assigned a higher costs for positive class samples. However, it is challenging to set cost matrices [43]. Ensemble algorithms can improve a single classifier's performance by combining a variety of base classifiers [49].

In this article, a novel algorithm called NBG is proposed, from the combination of NPSMOTE, BALO, and GVM algorithms. It can not only produce effective positive class sample by using an improved SMOTE algorithm NPSMOTE, but also the classification performance of imbalanced classification is improved by exploiting the state-of-the-art BALO as a feature selection of more significant features and the recently proposed GVM in classification. The main contributions of this paper are as follows:

(1)A hybrid algorithm NBG is proposed to solve the imbalanced Small-sample data classification problems. It compromises the merits of over-sampling and feature selection,

(2)A new over-sampling algorithm, namely NPSMOTE, is proposed to generate new positive class samples. Unlike the SMOTE algorithm,the NPSMOTE generates a positive class sample between a positive class sample and negative class sample. In this algorithm, the purpose of generating synthetic samples is to improve the performance of the classifier. The algorithm is to find the samples with learning difficulties and synthesize them. In the MWMOTE algorithm, the importance of minority class sample is measured by the Euclidean distance between minority class sample, and its the nearest neighbor majority class sample. The importance of minority class sample represents the difficulty of learning minority class sample [10]. Therefore, we think that the direction of learning the difficulties of minority class sample should be the direction of connection with the nearest neighbor majority class sample,

(3)The state-of-the-art BALO algorithm is explored to be used as a feature selection algorithm. The BALO is a relatively new heuristic algorithm. It contains adaptive boundary shrinking mechanism and elitism, and has the advantage of high development and fast convergence speed. Furthermore, the feature selection algorithm can improve the classification performance of the algorithm,

(4)A recently proposed GVM is explored as the machine learning classification algorithm. Note that there is no any available recent research that applies the GVM in the class imbalance problem. The GVM has the advantage of strong generalization ability.

Among references in imbalanced classification [51,17,68,72,39], hybrid algorithms are usually better than single ones. This study's primary motivation is that resampling algorithm, feature selection algorithm, and the novel classification algorithm have different advantages, and there have certain limitations. Once identified and isolated their characteristics, the proposed hybrid algorithm merged their advantages to obtain a better classification performance than each of them isolated. Besides, BALO and GVM are rarely used in imbalanced classification, while NPSMOTE, BALO and GVM have their own advantages, so thus, NBG is proposed under the combination of SMOTE, BALO and GVM.

The rest of this article is organized as follows. Section 2 reviews the relevant researches related to the class imbalance problem, the proposed algorithm described in Section 3, the experimental results for verifying the proposed algorithm are given in Section 4, and finally, concluding remarks and future work directions are given in Section 5.

2 Related work

There are four primary algorithms to solve the class imbalance problem: resampling algorithm, feature selection and exaction algorithm, cost-sensitive learning algorithm and ensemble algorithm. In addition, there are also some hybrid algorithms. In this section we mainly describe the related work of these algorithms. He et al. give the research development about class-imbalance learning, which include the foundation of imbalanced learning, the unique technologies, and the main application fields in their book Imbalanced Learning Foundations, Algorithms, and Applications [35]. Alberto et al. provide a formal description of the imbalanced learning problem and focuses on its main features and the related solution in their book Learning from Imbalanced Data Sets [26].

2.1 Resampling

The oversampling algorithm balances the class discribution by generating new minority class samples, while the undersampling algorithm balances the class distribution by removing the intrinsic samples in the majority class. However, the undersampling algorithm is easy to lose useful information [48,64]. Furthermore, the most frequently used oversampling algorithm is the SMOTE algorithm [16] in which new minority class samples are created by interpolating the randomly selected two minority class samples. The disadvantages

Table 1 Summary of articles employing resampling algorithms

Category	Strategy	Articles
oversampling	Distance based	[16], [31] , [36], [10], [33], [10],[2]
	Cluster based	[20]
	Density based	[15]
	Smooth bootstrap form	[55]
	security levels	[14]
	Smooth bootstrap form	[55]
undersamping	random based	[11]
	Tmek link pair based	[77]
	Condensed Nearest Neighbor rule	[32]
	KNN based	[44]
	Neighborhood cleaning rule	[45]

of SMOTE include over-generalization, which is likely due to enlarging the overlapped areas between the majority and minority class [83]. Han et al. proposed Borderline-Smote [31] that creates new minority class samples by the boundary minority class samples, as the performance of this method depends on the choice of the number of the nearest neighbors. If the number of nearest neighbors is too small, a part of minority class samples was mistaken as noises and cannot synthesize the new sample.

Cohen et al. proposed AHC [20] that balance the dataset through cluster that undersamples the majority examples by the K-means algorithm and oversamples the minority examples by the agglomerative hierarchical clustering. Hu et al. proposed the MSMOTE [36], which considers the distribution of the minority class and removes noises. However, when the class imbalance and the class overlap problem happen simultaneously, it may lead to the problem that the production instances may be distributed in the majority class regions [10]. Bunkhumpornpat et al. proposed DBSMOTE [15] based on the concept of density cluster that produces the minority samples by the boundary minority class samples. He et al. proposed ADASYN [33] that produces the number of minority samples dependent on the number of the adjacent majority class samples, where the noise also increases in multiples [18]. Menardi et al. presented the ROSE [55] that oversampling the minority examples based on a smoothed bootstrap form.

Barua et al. proposed the MWMOTE based on the proposed two-stage process, where the weight is assigned [10] to the minority class samples based on the Euclidean distance between the candidate minority class and the candidate majority class first, then the minority class with large weights have higher oversampling opportunities. This algorithm cannot detect the relevant minority class samples with far away from the majority class samples [59]. Bunkhumpornpat et al. proposed the safeLevel-SMOTE [14] based on security levels that produce the new minority samples by selecting the two minority class samples with a high-security level. Abdi et al. presented the MDO [2] to balance the data that integrates the Mahalanobis distance-based over-sampling and boosting technologies. The above algorithms are the most well-known extended algorithms of SMOTE. Alberto et al. reviewed a state-of-the-art of SMOTE algorithm in its 15th year anniversary [25].

Random Under-Sampling (RUS) removes a part of the majority samples by the random sample algorithm [11]. Since the characteristics of randomness and contingency, this algorithm might lose some vital information on the majority samples [58]. Tomek proposed the TomekLinks [77] based on the idea of Tmek link pair that remove the noise and boundary majority sample. P E Hart proposed the CNN [32]– an algorithm that removes the majority sample far away from the boundary. Kubat et al. proposed the OSS [44] based on the KNN that deletes the majority sample, which is different from the nearest neighbor classes. Finally, Jorma et al. proposed the Neighborhood Cleaning Rule (NCL) [45] based on the ENN [82]. These resampling algorithms have been summarized in Table 1.

2.2 Feature selection and exaction

Table 2 Summary of articles employing feature selection and exaction algorithms

Category	Strategy	Articles
Filter	relies on the general statistical features of the training data without using any learning algorithm	[81], [47], [8] ,[5]
Wrapper	Evolutionary based	[12], [3]
Embedded	the process of learning classifier and feature selection are carried out simultaneously	[3]
Feature extraction	transforms data into a low-dimensional space	[73], [57]

Feature selection is to select some of the most representative features from the original features. There are three main types of feature selection: Filter, Wrapper and Embedded. The filter feature selection mainly relies on the general statistical features of the training data without using any learning algorithm. Wei et al. [81] proposed a new model based on the feature selection that exacts the crucial attributes related to the Total Hip Arthroplasty (THA) information to help make the decision. Lima and Pereira et al. [47] built a fraud detection model that use the undersampling strategy in the feature selection step, in which results showed that this model could efficiently improve the company's financial situation. Bae and Yoon et al. [8] proposed an ensemble framework for detecting polyp through the data sampling-based boosting algorithm and Partial Least Square (PLS) feature learning algorithm [70]. PLS is a wide class of methods for modeling relations between sets of observed variables by means of latent variables [70]. Finally, Alibeigi et al. [5] proposed an algorithm namely the DBFS to cope with high dimensional imbalanced data using the feature ranking algorithms. The Wrapper feature selection algorithm uses an evolutionary strategy to guide search. Beyan and Fisher et al. [12] proposed a hierarchical decomposition algorithm to deal with the class imbalance problem that integrates the clustering and outlier detection technologies. Al-Ghraibah et al. [3] proposed a new feature exaction method based on a sub-sampled classification framework to predict flare activity. The embedded feature selection algorithm means that the process of learning classifier and feature selection are carried out simultaneously. Dubey et al. [21] studied an ensemble system using the feature selection and data sampling to analyze various sampling algorithms, which showed that the proposed ensemble model could achieve stable and promising results.

Feature extraction transforms data into a low-dimensional space [80]. It is noted that the selected features and the original features are separated in different space. Song et al. designed a new predictor based on a suitable feature exaction algorithm called the nDNA-Prot that improves the identification of the DNA-binding proteins [73]. Moepya et al. [57] demonstrated the effectiveness of a cost-sensitive classifier for financial fraud detection problem. Also, this paper removed the redundant features through PCA and FA. These feature selection and extraction algorithms have been summarized in Table 2.

2.3 Cost-sensitive learning

Table 3 Summary of articles employing cost-sensitive learning

Category	Strategy	Articles
Cost-sensitive learning	Modifying the decision thresholds	[93]
	compensation strategy	[86]
	the cost-sensitive margin mean and the cost-sensitive penalty	[19]
	the fuzzy rough set theory	[78]
	using a weighting strategy	[61],[4]

The main idea of the cost-sensitive learning is as follows. When training the classification model, the least squares error of the samples is no longer minimized as a training algorithm, but the pursuit of the overall misclassification cost is minimized [22]. Compared to the resampling algorithm, the cost-sensitive learning algorithm is relatively rare. The reasons are as follows: 1) it is difficult to determine the optical value of the cost matrices, 2) the classification algorithm often needs to be adjusted in the cost-sensitive learning algorithm. The adjustment of the method requires a certain amount of expert knowledge and practical experience [29]. Zou et al. [93] proposed a novel framework that improves the classification performance in the class imbalance problem by finding the best classification threshold. Yu et al. [86] proposed a novel algorithm to deal with the imbalance class problem based on a compensation strategy and ELM. Cheng et al. proposed a new algorithm namely the CS-LDM that introduces the cost-sensitive margin mean and the cost-sensitive penalty [19]. A new ensemble algorithm based on the cost-sensitive decision trees and the evolutionary algorithm is proposed by Krawczyk et al. [43]. Vluymans et al. [78] proposed a new type of classifier to deal with the multi-instance class imbalance problem based on the fuzzy rough set theory. Oh et al. [61] proposed a new error function that enlarges the update weights of the minority class and reduces the update weights of the majority class in which the results showed that such method could be efficiently applied to the class imbalance problem. Finally, Ali et al. proposed the Can-CSC-GBE [4] that integrates the CSL, GentleBoost, AdaBoostM1 and Bagging, which the results showed that such a system improves the classification performance on the breast cancer data set. These cost-sensitive learning algorithms have been summarized in Table 3.

2.4 Ensemble

Table 4 Summary of articles employing ensemble algorithms

Category	Strategy	Articles
Ensemble	individual penalty parameters and the weighted exponential error function	[91]
	combining the principles of boosting and the construction of supervised projections	[27]
	novel undersampling	[9]
	novel oversampling	[69]
	active learning	[76]
	Stochastic-Ensemble	[13]

As known, the ensemble algorithm has better classification performance than a single classifier. Therefore, some researchers have solved the class imbalance problem by using the ensemble algorithm in recent years. Zieba et al. proposed the boosted SVM to predict the post-operative life expectancy in lung cancer patients, where individual penalty parameters are introduced and the weighted exponential error function is minimized [91]. García-Pedrajas and García-Osorio proposed a new ensemble algorithm, where a supervised projection of the original data into a new space is obtained first, then used to train the classier [27]. Bao et al. proposed a new undersampling ensemble algorithm, namely the BNU-SVM based on the SVM and boosted Near-miss under-sampling algorithm [9], from which results showed that such a algorithm is a effective solution to concept detection with large-scale imbalanced data sets. Ren et al [69]proposed different ensemble classifiers to detect the Microaneurysms (MAs) based the adaptive over-sampling algorithm. The experiments have shown that these algorithms effectively improve the classification performance of the minority class. Zieba and Tomczak [76] proposed a novel training algorithm to predict the short-term loans repayment based on an active learning strategy and boosting SVM. Lusa et al. proposed the Stochastic-Ensemble to deal with the high-dimensional data with rare events [13], in which the results showed that this method could achieve excellent performance on high-dimensional data with rare events. These ensemble algorithms have been summarized in Table 4.

2.5 Hybrid algorithm

Table 5 Summary of articles employing hybrid algorithms

Category	Strategy	Articles
Hybrid	combined resampling with feature selection	[51]
	combined resampling with ensemble algorithm	[17], [68], [72], [39]

In addition to the above four algorithms, some algorithms combine with the above algorithms to reach better performance. Taghi et al [51] combined resampling with feature selection algorithm to alleviate the class imbalance problem, which results showed that the combined algorithm can obtain better classification performance than single algorithm in software defect prediction. Chawla et al [17] proposed SMOTEBoost based on resampling and ensemble algorithm, where indirectly changing the update weights and modifying for skewed distributions. Experiments shown that SMOTEBoost could effectively improve the rare class classification performance if compared to SMOTE and Boosting. Rayhan et al [68] proposed CUSBoost that combined undersampling with boosting algorithm, which results showed that CUSBoost could efficiently be applied to classify highly imbalanced datasets. Chris et al [72] proposed RUSBoost that introduce RUS into the AdaBoost algorithm, which results showed that RUSBoost runs faster and has better classification performance than SMOTEBoost. Zhao et al [39] proposed WHMBoost based on two resampling algorithms and two base classifiers. They considered that resampling algorithm and single base classifier might have specific limitations. The hybrid algorithm based on resampling and base classifer can complement each other. Experiments have shown that WHMBoost achieves better AUC, F-Measure, and Geometric Mean than other algorithms as AdaBoost, RUSBoost, RBBoost, RHSBoost, SMPTEBoost, CUSBoost, MeBoost on 40 imbalanced datasets.These hybrid algorithms have been summarized in Table 5.

3 Proposed algorithm

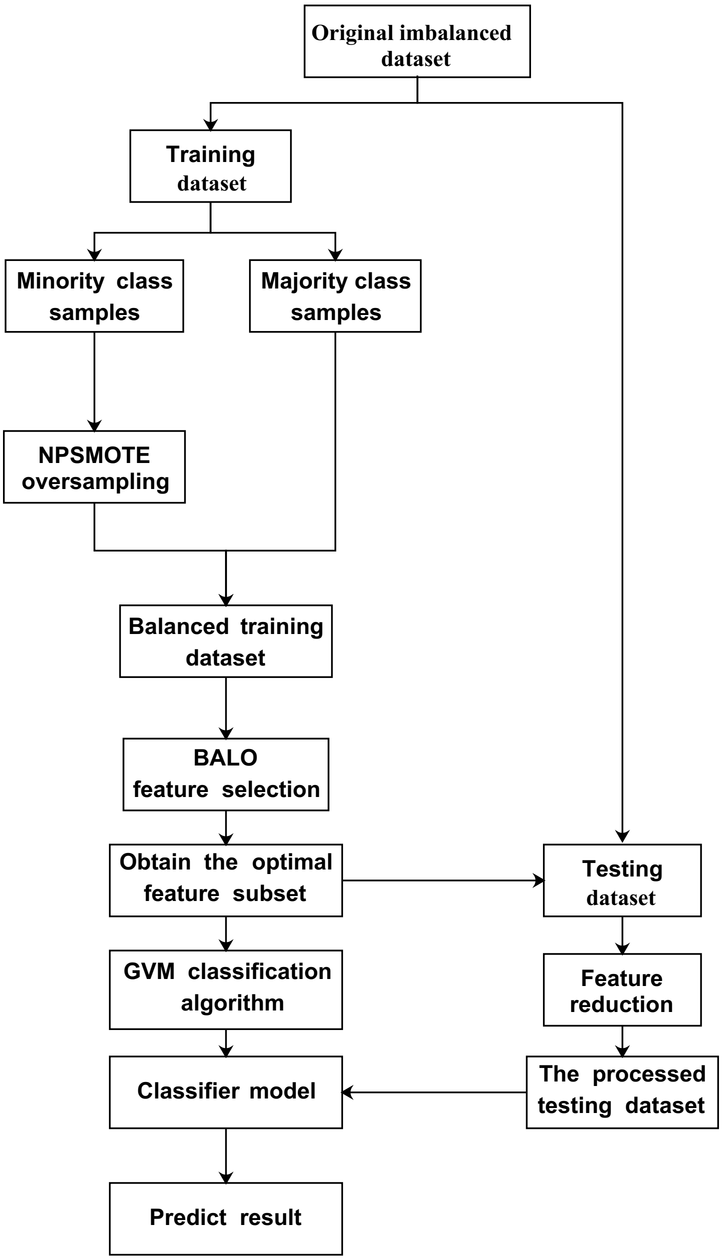

Fig. 1 The complete workflow of the proposed NBG

In this section, the complete workflow of the proposed NBG is given, as shown in Fig. 1. First, the complete imbalance data set (two-class) is divided into a training data set and testing data set by using the stratified random sampling method. A stratified random sample method is a random sample in which members of the population are first divided into strata, then are randomly selected to be a part of the sample. Next, to adopt the oversampling algorithm, the training data set is divided into the minority class subset and the majority class subset. The synthetic minority class samples produce the third step by applying the proposed oversampling algorithm NPSMOTE. From this, a balanced training set is obtained. To enhance the class performance, the optimal feature subset in the balanced training set by the state-of-the-art BALO method is selected in the fourth step, so then the processed training set is used to train the machine learning classifier. A recently proposed GVM algorithm is applied as the machine learning classification algorithm in this research. The testing data set's dimension is reduced by the optimal feature subset before the processed testing data set is presented to the classifier model. Lastly, the predicted result after the testing is obtained. In the following subsection, the NPSMOTE, BALO, and GVM are described.

As overall, a novel algorithm called NBG is proposed in which the SMOTE, BALO and GVM are combined. It can produce an effective positive class sample by using an improved SMOTE algorithm NPSMOTE. However, the classification performance of imbalanced classification is improved by exploiting the state-of-the-art BALO as a feature selection of more significant features and the recently proposed GVM in classification. As a note, the NPSMOTE generates a positive class sample between a positive class sample and negative class sample. In this algorithm, the purpose of generating synthetic samples is to improve the performance of the classifier. The algorithm is to find the samples with learning difficulties and synthesize them. The importance of minority class sample represents the difficulty of learning minority class sample [10]. Therefore, we believe that learning the difficulties of learning minority class sample should be the direction of connection with the nearest neighbor majority class sample. Therefore, the NPSMOTE algorithm can generate more valuable samples for classification. In the class imbalance problem, minority class samples are quickly treated as noises, which can be reduced by feature selection [85]. Furthermore, the BALO algorithm can adaptively search the space of features optimally and be converging to an optimal optimal solution better than well-known feature selection methods Particle Swarm Optimizer (PSO), Genetic Algorithms (GAs), Ant Lion Optimizer (ALO), and Binary Bat Algorithm (BBA) on 21 data sets [23]. Therefore, the state-of-the-art BALO algorithm is explored to be used as a feature selection algorithm. Lastly, the GVM has the advantage of solid generalization ability, so then the proposed GVM is explored as the machine learning classification algorithm.

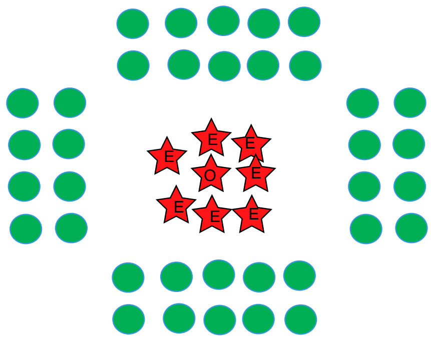

Fig. 2 Minority class sample O is surrounded by outer minority class sample E

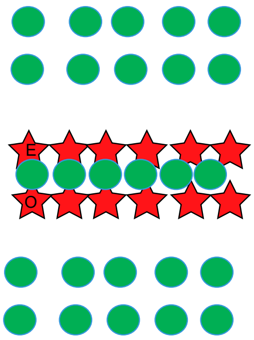

Fig. 3 Minority class sample E and O are clamped by majority class sample

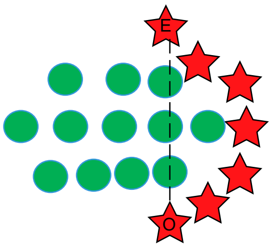

Fig. 4 Minority class samples are enclosed in an arc, E and O are sandwiched with majority class samples

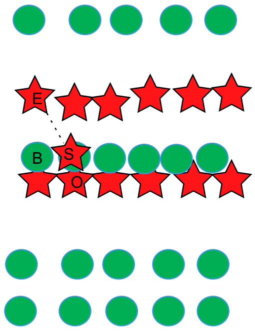

Fig. 5 The synthetic sample S generated by the MWMOTE falls in majority class

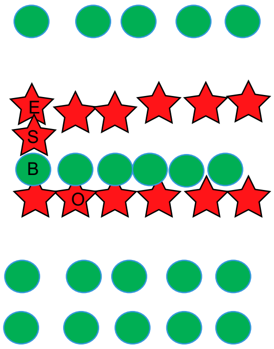

Fig. 6 The synthetic sample S produced by the NPSMOTE falls between E and B

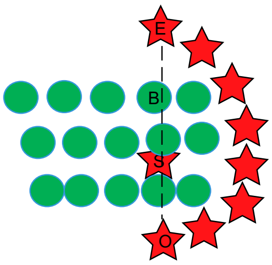

Fig. 7 The synthetic sample S generated by the MWMOTE falls in majority class

3.1 NPSMOTE

In this subsection, the shortcomings of the existing oversampling algorithms are introduced, and the motivation of the NPSMOTE algorithm is given, fol-

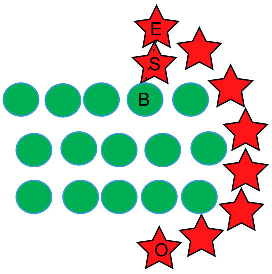

Fig. 8 The synthetic sample S produced by the NPSMOTE falls between E and B

lowed by a detailed presentation of the NPSMOTE algorithm. The Borderline-SMOTE mainly performs the SMOTE operations on the border-line samples. However, this algorithm may miss the relevant border-line samples as given in Fig. 2, where the stars represent the minority class samples, and the circulars represent the majority class samples. Suppose that the whole data set is S, the majority class is NS and the minority class is PS. $mins$ and $majs$ represents the number of minority and majority class samples. For every $x_i(i = 1, 2, .., mins)$ in the minority class PS, we calculate its K nearest neighbors from the whole data set S, The number of majority class samples among the K nearest neighbors is denoted by M. In the Borderline-SMOTE algorithm, if $K > M \geq K/2$, namely the number of x_i's majority nearest neigbors is larger than the number of its minority ones, x_i is defined as a border-line sample. In this case, if K=5, the M values of the samples E and O are both 0, which is not satisfied $K > M \geq K/2$, the samples E and O are not defined as the border-line samples. In fact, there are the border-line samples. The similarity between the ADASYN algorithm and the Borderline-SMOTE algorithm lies in that it also pays attention to the minority class samples on the boundary, but it uses the density distribution of samples to assign the different weights to each minority class sample and determines the probability of each minority sample as the primary sample according to the weight. However, the ADASYN algorithm may have an inappropriate weight assignment, as shown in Fig. 2 this phenomenon. In the ADASYN algorithm, the weight w_i of the sample is defined as: $w_i = \frac{M/K}{Z}$, where Z is a normalization factor. In this case, if K=5, the M values of the samples E and O are both 0, then the weight w_i of the samples E and O are both 0. In fact, the samples E and O are the key samples.

The MWMOTE algorithm does not consider the differences between the minority and majority class samples and so the synthetic samples generated by the MWMOTE algorithm are mostly to be misclassified by a classifier. Figs. 3 and 4 shows two examples. As shown in Fig. 3, there are six majority class samples are sandwiched between minority class samples E and O. They are relatively close to the minority class sample and are easyily misjudeged as a cluster. In this way, it is possible to obtain the majority class samples by using the interpolation method of the minority class samples E and O.

The synthetic sample may the bad sample, which will have a negative impact on our classification. In the MWMOTE algorithm, the generated sample is randomly selected from clusters. As shown in Fig. 4, if the selected generated samples are E and O, the generate samples may be in the majority class, which will have a negative impact on classification.

The Borderline-SMOTE and AYASYN algorithm generate synthetic samples based on the idea of KNN. The Borderline-SMOTE algorithm may miss the key bound-line samples, and the AYASYN algorithm may allocate the incorporate weights to the samples. Furthermore, to address how to choose the value of K is a challenging problem in these two algorithms. The MWMOTE algorithm generates synthetic samples based on the idea of clustering. It does not take into account the difference between the minority and majority class samples and also not use special conditions to select the two samples for interpolation.

Algorithm 1 NPSMOTE

```
Input: Data set $S$={$(x_i, y_i)$,i=1,2,...,N, $y_i \in \{maj, min\}$}, The number of majority sample $N_{maj}$,The number of minority sample $N_{min}$, $N_{maj}$+$N_{min}$=$N$, Nearest neighbor parameter $K$.
Output: The oversampled data set $S^{new2} = \{(x_i, y_i), i = 1, 2, ..., N+(N_{maj}-N_{min})\}, y_i \in \{maj, min\}\}$
Separate data sets $S$according to different categories, majority sample set $S^{maj}$and minority sample set $S^{min}$
Initialize $S^{new}$ as an empty set
Delete the noise data in the data set $S$, and get the new data set $S^D$, the new minority data set $PS^{'}$, and the new majority data set $NS^{'}$. The number of new minority class samples is $mins$and the number of new majority class samples is $majs$
Find the K-nearest majority class sample of $x_i \in PS^{'}$for each minority class, and unify the K-nearest majority class samples for each minority class, and place them in the set $SB_i$
Calculate the Euclidean distance of $r$ between $x_i$ and its nearest majority sample classes, With $x_i$ as the center of the circle and $r$as the radius of the circle, the number of samples of minority classes in the circle is calculated as $O_p$. With the nearest sample of majority classes $z_i$ as the center of the circle, $r$as the radius of the circle, the number of samples of majority classes in the circle is calculated as $O_n$
for $i = 1$ to $mins$ do
    The weight of minority class sample according to the formula: $w_i = \frac{O_n+1}{O_p+1}$, $x_i \in PS^{'}$
    The limited distance of minority class sample according to the formula: $limd_i = \frac{O_n+1}{O_p+O_n+2}$,$x_i \in PS^{'}$
end for
for $i = 1$ to $mins$ do
    According to the formula $Newn_i = fracw_i \sum_{i=1} minsw_i * (majs - mins)$, the number of synthetic samples needed to be generated for each minority class sample is calculated as $Newn_i$, and the new sample is $x_i^p = x_i + d * d_i * (z_i - x_rani)$, $rand$ is a random number between 0 and 1
    Add $x_i^p$ to $S^{new2}$
end for
Get the oversampled data set $S^{new2} = S \cup S^{new2}$
```

The above algorithms are all based on the SMOTE algorithm, though there have some problems. The NPSMOTE algorithm is used to solve the issues mentioned above and based on the following considerations: 1) To avoid the expansion of noises, it is necessary to remove the noisy sample at the start. For every $x_i(i = 1, 2, .., mins)$ in the minority class PS, we calculate its K nearest neighbors from the whole data set S, The number of majority class samples among the K nearest neighbors is denoted by M. If $K = M$, x_i is defined as the noise sample [31,67], 2) For different minority class samples, different cost weights need to be allocated according to their distribution density and the location relationship of the majority class samples. The weight represents the importance of the minority class samples, and decides how many samples are synthesized based on this minority class sample, 3) The purpose of generating the synthetic samples is to improve the classifier's performance. Therefore, we focus on the hard-to-learn minority class samples, 4) The generation of synthetic samples should be limited by distance d_i. Suppose that the direction of learning the difficulties of minority class sample should be the direction of connection with the nearest neighbor majority class sample. If the synthetic sample is generated by between each minority class sample and its nearest neighbor majority class sample, this sample can easily change the boundary of the classifier for minoirty class. The closer the location of the synthetic sample distribution to the samples of majority class, the more decision boundaries that may widen minority class.

The NPSMOTE algorithm is shown in Algorithm 1 and mainly divided into three stages. The first stage mainly removes the noise samples in the minority class set, then we can get the new minority data set $PS^{'}$, and the new majority data set $NS^{'}$. Steps 1 to 3 belong to the first stage. The second stage, each minority class example $x_i \in PS^{'}$ is assigned the weight w_i and the distinct limit $limitd_i$. The setting of weights w_i mainly considers the following points: (1) Minority class samples near the boundary should be assigned higher weights than minority class samples away from the boundary, because minority class samples near the boundary can provide more decision information; (2) The sparse minority class samples are more important than the dense class minority samples, because the sparse minority class samples are the important samples that affect the classifier's performance, and more synthetic samples are needed to balance different types of samples; (3)It is more important for minority class samples nearby the majority class samples with denser distribution than for minority class samples nearby the majority class samples with sparse distribution. Because the former has a greater impact on the classifier's performance, we should give it a higher weight. The setting of the distance $limitd_i$ mainly considers the following two points: (1) The distance between the minority samples with the sparse distribution and the synthetic samples generated by them should be farther than the distance between the minority samples with the dense distribution and the synthetic samples generated by them, because the former needs to generate more wider distance $limitd_i$ to improve the classifier's performance; (2) It is more important for minority class samples nearby the majority class samples with denser distribution than for minority class samples nearby the majority class samples with sparse distribution, the former should be assigned more wider distance $limitd_i$. Step 4 to 9 belong to the second stage. The third stage, the number of the synthetic samples generated by interpolating the minority class sample $x_i \in PS^{'}$ and the nearest neighbor majority class sample. Step 10 to 14 belong to the third stage.

As shown in Fig.2, if the K value of the nearest neighbor isn't appropriate, the Borderline-SMOTE and ADASYN algorithm have no majority class samples in the K-nearest neighborhood samples of the minority class sample. In comparison, the NPSMOTE algorithm is to find the nearest K majority class samples, rather than finding the majority class samples from the nearest neighborhood samples. The nearest K majority class samples are always be found, so the NPSMOTE algorithm does not appear the above problem. As shown in Figs .5, and 7, the generated synthetic sample S generated from E and O may fall into the majority class by the MWMOTE. In the NPSMOTE algorithm, as shown in Figs.6, and 8, the synthetic sample S is generated from E and the nearest majority class sample B. Notably, the generated sample meets the requirements.

3.2 BALO

Algorithm 2 BALO

```
Input: $m_1$ number of ants, $m_2$ number of antlions, $epochs$ number of epchos
Output: $p_{best}$ the best antlion position, $f_{best}$ the best antlion fitness value .
All of antlions' and ants' position are randomly initialize as 0 or 1.
Calculate the fitness value of all antlions and ants.
Sort all of the antlions' fitness value to obtain the best antlion position $p_{best}$.
while the end condition is not satisfied do
    Calculate the mutation rate $r$ by Eqs. (4)
    for each ant do
        Select an antlion using Roulette wheel
        Perform mutation operation on the selected antlion by Eqs. (3), called $CW_1$.
        Perform mutation operation on the best fitness antlion by Eqs. (3), called $CW_2$.
        Perform crossover operation between $CW_1$ and $CW_2$ by Eqs. (2), then we get the new position of ant.
    end for
    Calculate the fitness of all ants.
    Set an antlion as its corresponding ant it if the corresponding ant fitness becomes fitter than antlion fitness
    Adjust elite if an antlion becomes fitter than the elite
end while
Reture the best antlion position $p_{best}$ and its fitness value $f_{best}$.
```

The ALO algorithm is a swarm intelligence optimization algorithm. Proposed by Mirjalili in 2015, it mainly simulates the process of antlions catching ants

[56]. In the process of hunting, the antlion first digs a conical pit in the sand, moves the sand through a circular path and throws it out, which becomes a pit trap. Next, the larva ant lion hides at the bottom of the pit below and waits for the prey to fall into the trap. If the edge of the pit is sharp, the prey is more natural to fall into the trap. As the ant lion realizes that its prey has fallen into the trap, it tries to prevent the prey from escaping and trying to catch the prey. At the same time, after the prey falls into the trap, it also tries to escape the trap. The ant lion keeps throwing sand out so that the prey could fall further into the bottom of the pit. When the prey falls entirely into the bottom of the pit, the antlion will eat the prey. Then the antlion rebuilds the trap and continues to wait for the next prey.

The ALO algorithm mainly includes the following steps: defining random walk ants, establishing the trap by the antlion, ant falling into traps, ant being caught by antlion, rebuilding the trap by the antlion. The ALO algorithm includes the adaptive boundary shrinking mechanism and elitism, it has high development and fast convergence speed [56]. The adaptive boundary shrinking mechanism refers to the fact that when an ant is found to falling into a trap, the antlion quickly throws sand out to prevent the ants from escaping. In the ALO algorithm, it is represented by the adaptive reduction of the radius of the ant's random walk. Elitism means that after each iteration, the antlion with the best fitness is preserved. The location of ants is influenced by the elite antlion and the antlion selected by the roulette, so the location of the ant depends on the average value of the elite antlion and the antlion selected by the roulette.

The BALO algorithm is developed based on the ALO. It mainly limits the initial position of the ALO algorithm to discrete the value which is used to solve the constraint problem of the discrete space [23]. It is equivalent to the binary ALO algorithm. The pseudo code of the NPSMOTE algorithm is shown in Algorithm 2. In the ALO algorithm, each ant updates its position by an average of two positions, one of which is obtained by the elite antlion through the appropriate random walk step size, and one of which is obtained by the selected antlion through the appropriate random walk step size. In the BALO algorithm, the same search method is used, but the average operator is changed to cross operation. The cross formula is as shown in Eq. (1):

$$Ant_i^{t+1} = Crossover(RW1, RW2) \tag{1}$$

Among them, $Crossover(x; y)$ represents the crossover operation between x and y, $RW1$ and $RW2$ represent the carrier of the elite antlion and randomly selected antlion. The average run character in the ALO algorithm represents the operation of attracting ants into the antlion trap. A crossover operator replaces the BALO algorithm. The crossover used here is a simple random crossover, and the probability between its two input vectors is the same as given in Eq. (2) below.

$$x^d = \begin{cases} x_1^d & if(rand) \geq 0.5 \\ x_2^d & otherwise \end{cases} \tag{2}$$

$RW1$ represents the attraction of elite antlions to ants, and represented by the random walk of elite antlions with appropriate steps. It can be expressed by the random variation around the selected antlion. The Eq. (3) express the mutation rate of the binary elite antlions in the BALO algorithm. $RW2$ represents the attraction of the selected antlion to ants by the roulette using random mutations.

$$x_{out}^d = \begin{cases} x_{in}^d & if(rand1) \geq r \\ rand2 & otherwise \end{cases} \tag{3}$$

$$r = 0.9 + \frac{-0.9 * (i-1)}{IterMax - 1} \tag{4}$$

x_{out}^d represents the d-th dimension of the transformed output vector. x_{in}^d represents the input vectors that are crossed. $rand1$, $rand2$ are two random numbers between 0 and 1. r is the mutation rate. The value of r is linearly decreasing from 0.9 to 0 with the number of iterations. The formula for r is as shown in Eq. (4), where r represents the i-th iteration's mutation rate, and $IterMax$ represents the total number of iterations to retrieve the optimal value.

3.3 GVM

As proposed in 2016, the GVM algorithm is mainly composed of three essential layers of neural networks, namely input layer, hidden layer and output layer [88]. The difference between the GVM and the traditional neural networks lies in the introduction of the Monte Carlo training algorithm and the Design Risk Minimization principle. The core idea of the Monte Carlo training algorithm is the random variation and optimal selection. The Design Risk Minimization principle is as follows. Within the same parameter range, different parameters follow different paths, and the variance of output results is the smallest. That is, the same results can be achieved through different paths so that we can pay more attention to the essence of data. The GVM has a strong generalization ability and can obtain excellent performance in the small data sets, and successfully applied in Android malware detection, phishing detection and the groundwater status prediction [90,24,89].

Table 6 Data descriptions used in the experiment

Name	Minority	Majority	Features	Sample size	Minority size	Majority size	Imbalance rate(IR)	Source
breast_tissue	"car" and "fad"	All other	9	106	36	70	1.94	UCI
bupa	"1"	"2"	6	345	145	200	1.38	UCI
cleveland	positive	negative	12	303	35	268	7.66	Keel
ecoli01VS235	positive	negative	7	244	24	220	9.17	Keel
glass4	containers	ALL other	9	244	13	201	15.47	UCI
Wisconsin	Malignant	Benign	9	683	239	444	1.86	UCI
glass6	headlamps	ALL other	9	244	29	185	6.38	Keel

4 Experimental results and analysis

4.1 Data sets and experimental environment

As shown in Table 6, seven class-imbalanced data sets are used in the experiments performed, i.e., $Bupa$, $glass4$, $Breast\_tissue$ and $Wisconsin$ from the UCI machine learning repository [66] data sets, and $cleveland$, $glass6$ and $ecoli01VS235$ from Keel [1] data sets. Details of the data sets mainly include minority categories, majority categories, the number of the features, the total number of samples, the sample size of the minority and majority classes, and the source of the data sets. $Bupa$, $cleveland$, $ecoli01VS235$, $Wisconsin$ are two-class data sets, while $breast\_tissue$, $glass4$, $glass6$ are converted into two categories in the experimentation.

The experiments in this study are performed in a PC with Intel core i5 7500 3.4GHz CPU, 8MB SmartCache, 8GB memory, Windows 10 64-bit OS, running in MATLAB R2017a environment. The final results presented are average result of 20 runs.

4.2 Evaluation metric and function

As in [46,34,28,60], *Accuracy*, *True-positive rate* (TPR), *False-positive rate* (FPR), AUC, *F-measure*, *G-mean* are the most commonly used evaluation metrics. We also adopt these evaluation metrics to compare the performance of different algorithms, and corresponding formulas are as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5}$$

$$TPR = \frac{TP}{TP + FN} \tag{6}$$

$$FPR = \frac{FP}{TN + FP} \tag{7}$$

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = TPR = \frac{TP}{TP + FN} \tag{9}$$

$$F\text{-}measure = \frac{2 * Recall * Precision}{Recall + Precision} \tag{10}$$

$$G\text{-}mean = \sqrt{\frac{TP}{TP + TN} * \frac{TN}{TN + FP}} \tag{11}$$

where $True\ positive$ (TP) is the number of positive class correctly classified as positive class, $True\ negative$ (TN) the number of negative class correctly classified as negative class. $False\ positive$ (FP) the number of negative class mistakenly classified as positive class, $False\ negative$ (FN) the number of positive class mistakenly classified as negative class, *Accuracy* the ratio of the number of correctly predicted samples to the number of all the predicted samples, and TPR the ratio of the number of positive classes that are correctly predicted to be positive classes to the actual number of positive classes. The larger the index, the better. FPR represents the ratio of the number of negative classes that are mistakenly predicted to be positive classes to the number of actual negative classes: the smaller the indicator, the better. *F-measure* is a comprehensive indicator whose value is closer to 1, representing the better performance of the classifier, and *G-mean* is also a comprehensive indicator-the closer to 1, the better. When the data is imbalanced, this indicator is very valuable. *AUC* stands for the area under the ROC curve- the bigger, the better. In this article, the minority class is a positive class, while the majority class is a negative class.

The NBG algorithm's evaluation function seeks a balance between the error rate and the number of features. As shown in Eq.(12), *Accuracy* represents the overall recognition rate, S represents the number of features of the selected feature subset. T represents the total number of features, and $\alpha + \beta = 1$, where α and β are used to weigh the weight relationship between the error rate and the number of features.

$$fitness = \alpha(1 - Accuracy) + \beta\frac{S}{T} \tag{12}$$

Table 7 Description of Contrast Algorithms

Algorithm	Detailed description
GVM	GVM classification algorithm
BM	BALO is used as feature selection algorithm and GVM is used as classification algorithm
NG	NPSMOTE is used as oversampling algorithm and GVM is used as classification algorithm
ABM	AYASYN is used as oversampling algorithm, BALO is used as feature selection algorithm and GVM is used as classification algorithm
SBM	SMOTE is used as oversampling algorithm, BALO is used as feature selection algorithm and GVM is used as classification algorithm
BBM	Borderline_SMOTE is used as oversampling algorithm, BALO is used as feature selection algorithm and GVM is used as classification algorithm
MSBM	MSNOTE is used as oversampling algorithm, BALO is used as feature selection algorithm and GVM is used as classification algorithm
MWBM	MWNOTE is used as oversampling algorithm, BALO is used as feature selection algorithm and GVM is used as classification algorithm
NBM	NPSMOTE is used as oversampling algorithm, BPSO is used as feature selection algorithm and GVM is used as classification algorithm
NBG	NPSMOTE is used as oversampling algorithm, BALO is used as feature selection algorithm and GVM is used as classification algorithm

4.3 Comparison of the NBG algorithm and other algorithms

To verify the proposed algorithm's performance and effectiveness, comparisons to similar algorithms are shown in Table 7. In all oversampling algorithms, we stop sampling as the number of minority class samples is the same as the majority class samples by the oversampling algorithm. All the algorithms are tested by the random stratified sampling, where 80% of which are the training sets and 20% are the testing sets. The parameters of the BALO algorithm and the GVM algorithm are the same in the same data set. Among them, the population number of BALO algorithm in *breast_tissue*, $ecoli01VS235$, $glass6$, $glass4$ and $WBCD$ is set as 5, and the number of iterations is set as 4. The BALO algorithm's population number in *bupa* and *cleveland* is set as 5, and the number of iterations is set as 10. The number of nearest neighbors in the oversampling algorithms of NG, NBM, ABM, SBM, BBM, MSBM and NBG is set as 5, while the number of $K1$ in the oversampling algorithms of

Table 8 Experimental results of NBG and nine similar algorithms on the testing data set

Data set	Method	Accuracy	TPR	FPR	AUC	G-mean	F-measure
Breast_tissue	GVM	0.7273	0.375	0.0714	0.6518	0.59	0.5
	BM	0.7727	0.5	0.0714	0.7143	0.6814	0.6154
	NG	0.7727	0.75	0.2143	0.7679	0.7676	0.7059
	ABM	0.8182	0.625	0.0714	0.7768	0.7618	0.7143
	SBM	0.8636	1	0.2143	0.8929	0.8864	0.8421
	BBM	0.9091	0.875	0.0714	0.9018	0.9014	0.875
	MSBM	0.9091	1	0.1429	0.9286	0.9258	0.8889
	MWBM	0.8182	1	0.2857	0.8571	0.8452	0.8
	NBM	0.8636	1	0.2143	0.8929	0.8864	0.8421
	NBG	**0.9545**	**1**	**0.0714**	**0.9643**	**0.9636**	**0.9412**
bupa	GVM	0.6812	0.5862	0.25	0.6681	0.6631	0.6071
	BM	0.6957	0.5172	0.175	0.6711	0.6532	0.5882
	NG	0.6957	0.6897	0.3	0.6948	0.6948	0.6557
	ABM	0.7107	0.5172	0.15	0.6836	0.6631	0.6
	SBM	0.7101	0.6552	0.25	0.7026	0.701	0.6552
	BBM	0.7107	0.5517	0.175	0.6884	0.6747	0.6154
	MSBM	0.6812	0.6207	0.275	0.6728	0.6708	0.6707
	MWBM	0.6812	0.6897	0.325	0.6823	0.6823	0.6452
	NBM	0.6232	0.7586	0.475	0.6418	0.6311	0.6286
	NBG	**0.6957**	**0.7586**	**0.35**	**0.7043**	**0.7022**	**0.6769**
cleveland	GVM	0.9143	0.3333	0.0313	0.651	0.5683	0.4
	BM	0.9714	0.6667	0	0.8333	0.8165	0.8
	NG	0.8517	0.6667	0.125	0.7708	0.7638	0.4444
	ABM	0.9143	1	0.0938	0.9531	0.952	0.6667
	SBM	0.8571	0.6667	0.125	0.7708	0.7638	0.444
	BBM	0.9714	0.6667	0	0.8333	0.8165	0.8
	MSBM	0.8571	1	0.1563	0.9219	0.9186	0.5455
	MWBM	0.9143	1	0.0938	0.9531	0.952	0.6667
	NBM	0.8	1	0.2188	0.8906	0.8839	0.4615
	NBG	**0.9429**	**1**	**0.0625**	**0.9688**	**0.9682**	**0.7**
ecoli01VS235	GVM	0.9714	0.72	0	0.86	0.8465	0.8333
	BM	0.9755	0.8	0.0045	0.8977	0.8924	0.8711
	NG	0.9796	0.8	0	0.9	0.8944	0.8889
	ABM	0.9470	0.92	0.05	0.935	0.9333	0.7847
	SBM	0.9470	0.88	0.0455	0.9173	0.9155	0.7788
	BBM	0.9837	0.88	0.0045	0.9377	0.9346	0.9156
	MSBM	0.9592	0.84	0.0273	0.9063	0.903	0.8129
	MWBM	0.9755	0.84	0.0091	0.9155	0.9109	0.8778
	NBM	0.9796	0.8	0	0.9	0.8944	0.8889
	NBG	**0.9918**	**0.96**	**0.0045**	**0.9777**	**0.9766**	**0.9596**
glass4	GVM	0.9545	0.6667	0.0244	0.8211	0.8065	0.6667
	BM	0.9773	0.6667	0	0.8333	0.8165	0.8
	NG	0.7727	1	0.2439	0.878	0.8695	0.375
	ABM	0.9318	1	0.0732	0.9634	0.9627	0.6667
	SBM	0.9091	1	0.0976	0.9512	0.95	0.6
	BBM	0.9773	0.6667	0	0.8333	0.8165	0.8
	MSBM	0.9773	0.6667	0	0.8333	0.8165	0.8
	MWBM	0.8636	1	0.1463	0.9268	0.9239	0.5
	NBM	0.9318	1	0.0732	0.9634	0.9627	0.6667
	NBG	**0.9773**	**1**	**0.0244**	**0.9878**	**0.9877**	**0.8571**
Wisconsin	GVM	0.9562	0.9583	0.0449	0.9567	0.9567	0.9388
	BM	0.9635	0.9792	0.0449	0.9671	0.967	0.9495
	NG	0.9562	0.9792	0.0562	0.9615	0.9613	0.94
	ABM	0.9708	0.9792	0.0337	0.9727	0.9727	0.9592
	SBM	0.9708	0.9375	0.0112	0.9631	0.9628	0.9574
	BBM	0.9562	0.9167	0.0225	0.9471	0.9466	0.9362
	MSBM	0.9635	0.9583	0.0337	0.9623	0.9623	0.9485
	MWBM	0.9635	0.9792	0.0449	0.9671	0.967	0.9495
	NBM	0.9708	0.9792	0.0337	0.9727	0.9727	0.9592
	NBG	**0.9781**	**1**	**0.0337**	**0.9831**	**0.983**	**0.9697**
glass6	GVM	0.9535	0.6667	0	0.8333	0.8165	0.8
	BM	0.9535	0.6667	0	0.8333	0.8165	0.8
	NG	0.9767	1	0.027	0.9865	0.9864	0.9231
	ABM	0.9767	0.8333	0	0.9167	0.9129	0.9091
	SBM	1	1	0	1	1	1
	BBM	1	1	0	1	1	1
	MSBM	1	1	0	1	1	1
	MWBM	1	1	0	1	1	1
	NBM	0.9767	1	0.027	0.9865	0.9864	0.9231
	NBG	**1**	**1**	**0**	**1**	**1**	**1**

the MWBM is set as 5, $K2$ and $K3$ are set as 5 in the oversampling algorithm MWBM. The particle swarm optimization (PSO) algorithm is a kind of simulation optimization algorithm to simulate the birds' predation, which is proposed by Dr. Kennedy [40], and the binary PSO algorithm, BPSO algorithm, is proposed two years later [41]. The $C1$ and $C2$ of the BPSO algorithm are set as 1.49445 [85]. The number of iterations in the BPSO is set to 10, the population size to 10, while the maximum and minimum speeds are set to 1.

As shown in Table 8, the experimental results of the comparison among the proposed NBG and other similar algorithms can be concluded that the proposed NBG algorithm has notable advantages over other comparison algorithms on all the data sets. The specific analysis is as follows:

(1)The proposed algorithm NBG outperforms the single classification algorithm GVM, this result indicates that the NBG can improve the recognition rate of the minority classes and alleviate the negative impact of the class imbalance compared with the traditional classification algorithm. In contrast to the GVM, the NBG can obtain has obvious advantages in terms of *Accuracy*, TPR, *AUC*, *G-mean*, *F-measure* in all seven data sets. Especially, the GVM fall behind the NBG by 62.5%, 17.24%, 66.67%, 24%, 33.33%, 4.17%, 33.33% using TPR metrics in data set *breast_tissue*, *bupa*, *cleveland*, $ecoli01VS235$, $glass4$, $Wisconsin$ and $glass6$, respectively.

(2)Compare with NBG and BM, it indicates that the resampling algorithm NPASMOTE can further improve the classification performance based on the feature selection algorithm BM. We observe that the NBG outperforms the BM on TPR, *AUC*, *G-mean* metrics in all seven data sets. In respect of TPR, the NBG outperforms BM 50%, 24.14%, 33.33%, 33.33%, 16%, 2.08%, 33.33% in data set *Breast_tissue*, *bupa*, *cleveland*, $glass4$, $ecoli01VS235$, $Wisconsin$ and $glass6$, respectively. In particular, the NBG achieved the AUC of 0.9643, which improves 25% compared with the BM in data set *Breast_tissue*.

(3)By comparing NBG and NG, it can be found that the feature selection algorithm BALO further improves the classification performance of the imbalanced classification based on the oversampling algorithm. Compared with the NG, the NBG shows the better results with respect to *Accuracy*, TPR, *AUC*, *G-mean*, *F-measure* for all data sets. In particular, the NBG has a significant improvement over the NG in terms of *AUC*, *G-mean*, *F-measure*.

(4)Comparing the results of the ABM, SBM, BBM, MSBM,MWBM and NBG, It indicates that the oversampling algorithm NPSMOTE is more effective than the AYASYN, SMOTE, Borderline_SMOTE, MSNOTE and MWNOTE in generating minority class samples. One type is the oversampling hybrid group, which is a combination of the oversampling and feature selection algorithm BALO and includes ABM, SBM, BBM, MSBM, MWBM and NBG. The TPR, *AUC*, *G-mean*, *F-measure* of the NBG are the highest among these six hybrid algorithms in all seven data sets.

(5)Compare with NBM and NBG, it is evident that BALO is better than BPSO as the feature selection algorithm in the class imbalance problem. Comparing the results of the NBM and the NBG, the NBG obtains the better results in all seven data sets on *Accuracy*, *AUC*, *G-mean* and *F-mean* metrics.

The NBM fall behind the NBG by 16%, 2.08% using TPR metrics in data set $ecoli01VS235$, $Wisconsin$, respectively. The TPR of the NBM is the same as the NBG in other five data sets.

(6)By Comparing NG and GVM, it shows that the recognition rate of the minority classes is noticeable improved by the oversampling algorithm NPSMOTE. The NG is significantly better than the GVM in terms of TPR, FPR, AUC, *G-mean* in all seven data sets.The NG improves the GVM by 37.5%, 10.35%, 33.34%, 8%, 33.33%, 20.9%, 33.33% in terms of TPR in data set *Breast_tissue*, *bupa*, *cleveland*, $ecoli01VS235$, $glass4$, $Wisconsin$ and $glass6$, respectively. In respect of *F-measure*, the GVM fall behind the NG by 20.59%, 4.86%, 4.44%, 5.56%, 0.12%, 12.31% in data set *Breast_tissue*, *bupa*, *cleveland*, $ecoli01VS235$, $Wisconsin$ and $glass6$, respectively.

(7)Compare with GVM and BM, this reflected the fact that the feature selection algorithm BALO can significantly improve the classification performance of the imbalanced classification based on GVM algorithm. In comparison with the GVM, the BM obtains significantly better performance on TPR, AUC, *G-mean*, *F-measure* metrics in data set *Breast_tissue*, *cleveland*, $ecoli01VS235$, $glass4$, $Wisconsin$ and $glass6$.The BM outperforms the GVM by 4.54%, 1.45%, 5.71%, 0.41%, 2.28%, 0.73% using *Accuracy* metrics in data set *Breast_tissue*, *bupa*, *cleveland*, $ecoli01VS235$, $glass4$ and $Wisconsin$, respectively.

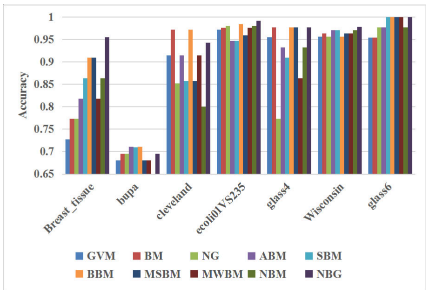

Fig. 9 Display of *Accuracy* value for the different algorithm on seven data sets

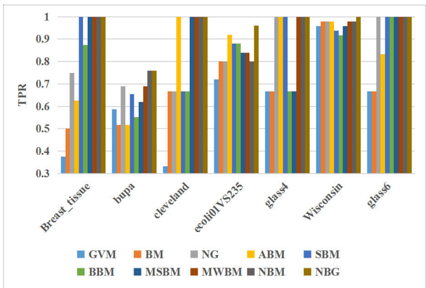

Fig. 10 Display of TPR value for the different algorithm on seven data sets

As it can be seen from Fig. 10, the TPR of the NBG is the highest among the ten algorithms in all seven data sets. It is depicted in Fig. 12 the AUC for ten algorithms in all seven data sets, and could observe that the AUC of the NBG is the highest. Additionally, Fig. 13 shows the *G-mean* for ten algorithms over the seven testing data sets, and identified that the *G-mean* of the NBG is the highest among the ten algorithms. From Fig. 14, we can observe that the NBG achieves the highest *F-measure* value among the ten algorithms in six out of the seven data sets. Fig. 9 shows the *Accuracy* of the NBG and the other nine algorithms on the seven data sets. We can see that the NBG achieves the highest *Accuracy* in five out of the seven testing data sets. Next, it is depicted in Fig. 11 the FPR of ten algorithms, and the NBG is

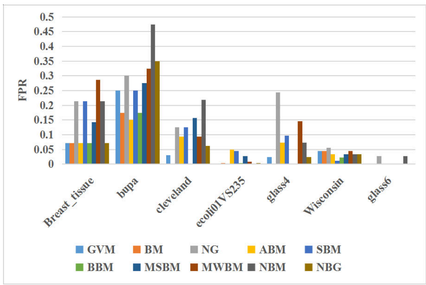

Fig. 11 Display of FPR value for the different algorithm on seven data sets

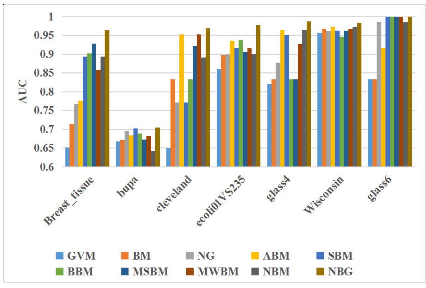

Fig. 12 Display of AUC value for the different algorithm on seven data sets

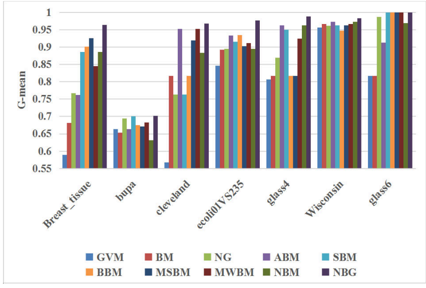

Fig. 13 Display of $G - mean$ value for the different algorithm on seven data sets

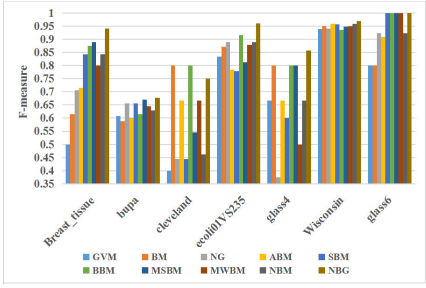

Fig. 14 Display of $F - measure$ value for the different algorithm on seven data sets

the lowest among the ten algorithms in two out of seven testing data sets. The representative value of column type that does not appear is 0. In summary, the proposed algorithm NBG significantly improves the recognition rate of the minority class in all data sets with respect to other nine similar algorithms.

Compared with the six advanced algorithms recently published, it verifies the NBG's effectiveness in the imbalanced classification. The experimental results are shown in Table 9 and 10. The best results for each indicator are highlighted. The data partitioning algorithm is the same as the algorithm in the comparative paper. In the data set $Breast\_tissue$, we applied 4-fold cross validation. In other six data sets, we applied 5-fold cross validation. Table 9 and Table 10 shows the comparisons between the NBG and previous algorithms, mainly including A-SUWO [59], SMOTE-IPF [74], CHC [83], SMOTE-ENN+LCMine+CAEP [50], NCL+LCMine+CAEP [50] and Incremetal-SVD [65]. "-" means that information is not available. They can be shown that the NBG outperforms all the other algorithms in all data sets. Therefore, the NBG can be better applied in the imbalanced data classification problem.

Besides, the proposed algorithm's time complexity is evaluated with the training running time averaged from 20 runs. The training running time using the comparison methods in Table 7 are shown in Table 11. We can see that the running time of the proposed method is acceptable. To further optimize

Table 9 Experimental results of NBG and other rencently published algorithms on the *Breast_tissue*, *bupa*, *cleveland*, *glass*4, *ecoli*01*VS*235 data set

Data set	Method	AUC	G-mean	F-measure
Breast_tissue	A-SUWO+SVM	0.86	0.748	0.685
	A-SUWO+KNN	0.845	0.779	0.71
	A-SUWO+LR	0.89	0.824	0.764
	A-SUWO+LDA	0.897	0.773	0.719
	NBG	**0.9362**	**0.9353**	**0.9165**
bupa	C4.5	0.644	-	-
	SMOTE+C4.5	0.6688	-	-
	SMOTE-ENN+C4.5	0.6146	-	-
	SMOTE-TL+C4.5	0.6018	-	-
	SL-SMOTE+C4.5	0.6684	-	-
	B1-SMOTE+C4.5	0.686	-	-
	B2-SMOTE+C4.5	0.6361	-	-
	SMOTE-IPF+C4.5	0.6753	-	-
	NBG	**0.7821**	**0.7762**	**0.7496**
cleveland	C4.5	0.5258	-	-
	SMOTE+C4.5	0.5485	-	-
	SMOTEENN+C4.5	0.5722	-	-
	SMOTE_TL+C4.5	0.6433	-	-
	SLSMOTE+C4.5	0.6007	-	-
	B1-SMOTE+C4.5	0.5475	-	-
	B2SMOTE+C4.5	0.5666	-	-
	SMOTEIPF+C4.5	0.6282	-	-
	NBG	**0.9688**	**0.9628**	**0.75**
glass4	SVM	0.9092	-	0.856
	SMOTE+SVM	0.9148	-	0.6633
	AYASYN+SVM	0.9176	-	0.6565
	sTL+SVM	0.9113	-	0.659
	sSafe+SVM	0.9143	-	0.6613
	sRST+SVM	0.9163	-	0.6463
	sCHC+SVM	0.9333	-	0.819
	EUSCHC+SVM	0.9251	-	0.7164
	AHC+SVM	0.935	-	0.8471
	FRB+CHC+SVM	0.923	-	0.7273
	FRB+SVM	0.8942	-	0.7197
	NBG	**0.9878**	**0.9877**	**0.8571**
ecoli01VS235	SVM	0.4955	-	0
	SMOTE+SVM	0.6606	-	0.4325
	AYASYN+SVM	0.5377	-	0.1648
	sTL+SVM	0.6628	-	0.4396
	sSafe+SVM	0.6598	-	0.4352
	sRST+SVM	0.6616	-	0.4264
	sCHC+SVM	0.6758	-	0.4844
	EUSCHC+SVM	0.7423	-	0.5691
	AHC+SVM	0.5405	-	0.1385
	FRB+CHC+SVM	0.7866	-	0.4224
	FRB+SVM	0.8659	-	0.5536
	Base+LCMine+CAEP	0.8377	-	-
	SMOTE+LCMine+CAEP	0.8023	-	-
	SMOTE-ENN+LCMine+CAEP	0.8632	-	-
	SMOTE-TL+LCMine+CAEP	0.8605	-	-
	AYASYN+LCMine+CAEP	0.8223	-	-
	Borderline SMOTE+LCMine+CAEP	0.7514	-	-
	SafeLevel SMOTE+LCMine+CAEP	0.8477	-	-
	ROS+LCMine+CAEP	0.8091	-	-
	ADOMS+LCMine+CAEP	0.8473	-	-
	SPIDER+LCMine+CAEP	0.8268	-	-
	AHC+LCMine+CAEP	0.8155	-	-
	SPIDER2+LCMine+CAEP	0.8132	-	-
	SMOTE-RSB+LCMine+CAEP	0.8227	-	-
	TL+LCMine+CAEP	0.7777	-	-
	CNN+LCMine+CAEP	0.7709	-	-
	RUS+LCMine+CAEP	0.8623	-	-
	OSS+LCMine+CAEP	0.7623	-	-
	CNNTL+LCMine+CAEP	0.8577	-	-
	NCL+LCMine+CAEP	0.8	-	-
	SBC+LCMine+CAEP	0.6682	-	-
	CPM+LCMine+CAEP	0.7641	-	-
	NBG	**0.9409**	**0.9384**	**0.8667**

Table 10 Experimental results of the NBG and other rencently published algorithms on the *Wisconsin*, *glass*6 data set

Data set	Method	AUC	G-mean	F-measure
Wisconsin	Base+LCMine+CAEP	0.9594	-	-
	SMOTE+LCMine+CAEP	0.9666	-	-
	SMOTE-ENN+LCMine+CAEP	0.9708	-	-
	SMOTE-TL+LCMine+CAEP	0.9791	-	-
	AYASYN+LCMine+CAEP	0.9737	-	-
	Borderline SMOTE+LCMine+CAEP	0.9666	-	-
	SafeLevel SMOTE+LCMine+CAEP	0.9696	-	-
	ROS+LCMine+CAEP	0.9605	-	-
	ADOMS+LCMine+CAEP	0.9622	-	-
	SPIDER+LCMine+CAEP	0.9666	-	-
	AHC+LCMine+CAEP	0.9593	-	-
	SPIDER2+LCMine+CAEP	0.9729	-	-
	SMOTE-RSB+LCMine+CAEP	0.9645	-	-
	TL+LCMine+CAEP	0.9739	-	-
	CNN+LCMine+CAEP	0.9706	-	-
	RUS+LCMine+CAEP	0.9656	-	-
	OSS+LCMine+CAEP	0.9671	-	-
	CNNTL+LCMine+CAEP	0.9688	-	-
	NCL+LCMine+CAEP	0.9791	-	-
	SBC+LCMine+CAEP	0.5	-	-
	CPM+LCMine+CAEP	0.9384	-	-
	NBG	**0.9842**	**0.984**	**0.9733**
glass6	Base+LCMine+CAEP	0.9052	-	-
	SMOTE+LCMine+CAEP	0.9365	-	-
	SMOTE-ENN+LCMine+CAEP	0.9338	-	-
	SMOTE-TL+LCMine+CAEP	0.9284	-	-
	ADASYN+LCMine+CAEP	0.9234	-	-
	Borderline-SMOTE+LCMine+CAEP	0.9032	-	-
	SafeLevel SMOTE+LCMine+CAEP	0.9203	-	-
	ROS+LCMine+CAEP	0.9252	-	-
	ADOMS+LCMine+CAEP	0.9311	-	-
	SPIDER+LCMine+CAEP	0.9279	-	-
	AHC+LCMine+CAEP	0.9365	-	-
	SPIDER2+LCMine+CAEP	0.9198	-	-
	SMOTE-RSB+LCMine+CAEP	0.9284	-	-
	TL+LCMine+CAEP	0.9252	-	-
	CNN+LCMine+CAEP	0.8658	-	-
	RUS+LCMine+CAEP	0.9423	-	-
	OSS+LCMine+CAEP	0.9068	-	-
	CNNTL+LCMine+CAEP	0.791	-	-
	NCL+LCMine+CAEP	0.9225	-	-
	SBC+LCMine+CAEP	0.7063	-	-
	CPM+LCMine+CAEP	0.7333	-	-
	Clustering-LMS	-	0.8773	-
	Clustering-SVD	-	0.8888	-
	CO^2RBFN-LMS	-	0.8593	-
	CO^2RBFN-SVD	-	0.8638	-
	Genetic-LMS	-	0.8877	-
	Genetic-SVD	-	0.895	-
	Incremetal-LMS	-	0.8743	-
	Incremetal-SVD	-	0.8913	-
	NBG	**0.9718**	**0.9706**	**0.9262**

Table 11 Comparison of training time(s) of different methods

Datasets	GVM	BM	NG	ABM	SBM	BBM	MSBM	MWBM	NBM	NBG
breast_tissue	7	1357	87	1437	1607	1437	156	159	727	147
bupa	28	1763	29	1687	2190	1301	2353	2359	2278	2148
cleveland	39	2849	88	3357	3025	2482	3599	3999	163	2861
ecoli01VS235	20	267	21	532	686	219	799	798	586	2182
glass4	7	95	9	184	152	112	142	196	706	277
Wisconsin	5	25	11	215	162	36	223	434	717	371
glass6	6	699	26	605	588	645	817	813	406	562

the algorithm, the proposed algorithm's time complexity should be improved in the future.

5 Conclusions and Future Work

Many existing algorithms are applied as one single algorithm for imbalanced classification problems, and the research on hybrid algorithms is seldom. The oversampling algorithm balances the class distribution by generating new minority class samples independent of the classification algorithm and relatively simple. Compared with the SMOTE, AYASYN, Borderline_SMOTE, MSNOTE, MWNOTE, the oversampling algorithm NPSMOTE can produce more effective positive class samples. The feature selection algorithm can effectively select significant features to improve the classification performance of imbalanced classification. The state-of-the-art BALO algorithm can adaptively search the space of features optimally, and better than the particle swarm optimizer and genetic algorithm [56]. In this article, a novel hybrid algorithm called the NBG is based on NPSMOTE and BALO for imbalanced binary data set classification. The NBG compromises the merits of oversampling and feature selection algorithms. The experiments are conducted on seven imbalanced data sets, and classification results show that our proposed NBG algorithm significantly outperforms nine similar algorithms and six algorithms recently published.

As future work, we would mainly conduct in the following areas: 1) the application of the BALO algorithm to improve the performance of the ensemble algorithm for solving the class imbalance problem. It is challenging to set parameters with respect to different algorithms in the ensemble algorithm, and we can apply the BALO algorithm to select the optimal parameter value, 2) to construct a hybrid algorithm based on the undersampling algorithm and feature selection algorithm. The undersampling algorithm may lead to the loss of important information [48,64], so the proposed undersampling algorithm must solve the problem of the loss of important information. The proposed hybrid algorithm combines the undersampling algorithm and feature selection algorithm's advantage and may achieve a better performance, 3) to establish an ensemble algorithm based on undersampling algorithm. The ensemble algorithm can obtain better performance than a single algorithm [63,38]. The undersampling algorithm outperforms the oversampling method in time efficiency. Therefore, we can use the undersampling algorithm as the base algorithm in the ensemble algorithm, 4) we will deliver an objective analysis of the ratio of the positive and negative samples in the used dataset and give some statistical hypotheses analysis about the future proposed algorithm, and 5) To further optimize the algorithm, the time complexity of the proposed algorithm should be improved in the future.

Acknowledgements

This work was supported by Guizhou University of Finance and Economics introduces talents scientific research start-up project in 2020 under Grant No. 2020YJ022.

References

1. Keel datasets. http://www.keel.es/.
2. Abdi, L., Hashemi, S.: To combat multi-class imbalanced problems by means of oversampling and boosting techniques. IEEE Transactions on Knowledge and Data Engineering **28**(1), 238–251 (2016)
3. Al-Ghraibah, A., Boucheron, L.E., Mcateer, R.T.J.: A study of feature selection of magnetogram complexity features in an imbalanced solar flare prediction data-set. In: IEEE International Conference on Data Mining Workshop, pp. 557–564 (2015)
4. Ali, S., Majid, A., Javed, S.G., Sattar, M.: Can-csc-gbe: Developing cost-sensitive classifier with gentleboost ensemble for breast cancer classification using protein amino acids and imbalanced data. Computers in Biology & Medicine **73**, 38–46 (2016)
5. Alibeigi, M., Hashemi, S., Hamzeh, A.: Dbfs: An effective density based feature selection scheme for small sample size and high dimensional imbalanced data sets. Data & Knowledge Engineering **81-82**(4), 67–103 (2012)
6. Amin, A., Anwar, S., Adnan, A., Nawaz, M., Howard, N., Qadir, J., Hawalah, A., Hussain, A.: Comparing oversampling techniques to handle the class imbalance problem: A customer churn prediction case study. IEEE Access **PP**(99), 1–1 (2016)
7. Anbar, M., Abdullah, R., Al-Tamimi, B.N., Hussain, A.: A machine learning approach to detect router advertisement flooding attacks in next-generation ipv6 networks. Cognitive Computation **10**(3-4), 1–14 (2018)
8. Bae, S.H., Yoon, K.J.: Polyp detection via imbalanced learning and discriminative feature learning. IEEE Transactions on Medical Imaging **34**(11), 2379 (2015)
9. Bao, L., Cao, J., Li, J., Zhang, Y.: Boosted near-miss under-sampling on svm ensembles for concept detection in large-scale imbalanced datasets. Neurocomputing **172**(C), 198–206 (2016)
10. Barua, S., Islam, M.M., Yao, X., Murase, K.: Mwmote–majority weighted minority oversampling technique for imbalanced data set learning. IEEE Transactions on Knowledge & Data Engineering **26**(2), 405–425 (2013)
11. Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. Acm Sigkdd Explorations Newsletter **6**(1), 20–29 (2004)
12. Beyan, C., Fisher, R.: Classifying imbalanced data sets using similarity based hierarchical decomposition. Pattern Recognition **48**(5), 1653–1672 (2015)
13. Blagus, R., Lusa, L.: Gradient boosting for high-dimensional prediction of rare events. Computational Statistics & Data Analysis **113** (2016)

14. Bunkhumpornpat, C., Sinapiromsaran, K., Lursinsap, C.: Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In: Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, pp. 475–482 (2009)
15. Bunkhumpornpat, C., Sinapiromsaran, K., Lursinsap, C.: Dbsmote: Density-based synthetic minority over-sampling technique. Applied Intelligence **36**(3), 664–684 (2012)
16. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research **16**(1), 321–357 (2002)
17. Chawla, N.V., Lazarevic, A., Hall, L.O., Bowyer, K.W.: Smoteboost: Improving prediction of the minority class in boosting. In: European Conference on Principles of Data Mining and Knowledge Discovery, pp. 107–1219 (2003)
18. Chen, S., He, H., Garcia, E.A.: Ramoboost: Ranked minority oversampling in boosting. IEEE Transactions on Neural Networks **21**(10), 1624–1642 (2010)
19. Cheng, F., Zhang, J., Wen, C.: Cost-sensitive large margin distribution machine for classification of imbalanced data. Pattern Recognition Letters **80**, 107 – 112 (2016). DOI https://doi.org/10.1016/j.patrec.2016.06.009. URL http://www.sciencedirect.com/science/article/pii/S0167865516301337
20. Cohen, G., Hilario, M., Sax, H., Hugonnet, S., Geissbuhler, A.: Learning from imbalanced data in surveillance of nosocomial infection. Artificial Intelligence in Medicine **37**(1), 7–18 (2006)
21. Dubey, R., Zhou, J., Wang, Y., Thompson, P.M., Ye, J.: Analysis of sampling techniques for imbalanced data: An n = 648 adni study. Neuroimage **87**(3), 220–241 (2014)
22. Elkan, C.: The foundations of cost-sensitive learning. In: International joint conference on artificial intelligence, vol. 17, pp. 973–978. Lawrence Erlbaum Associates Ltd (2001)
23. Emary, E., Zawbaa, H.M., Hassanien, A.E.: Binary ant lion approaches for feature selection. Neurocomputing **213**, 54–65 (2016)
24. Fang, F., Zhou, Q., Shen, Z., Yang, X., Han, L., Wang, J.Q.: The application of a novel neural network in the detection of phishing websites. Journal of Ambient Intelligence & Humanized Computing (13), 1–15 (2018)
25. Fernandez, A., Garcia, S., Chawla, N.V., Herrera, F.: Smote for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. Journal of Artificial Intelligence Research **61**, 863–905 (2018)
26. Fernández, A., García, S., Galar, M., Prati, R.C., Krawczyk, B., Herrera, F.: Learning from imbalanced data sets. Springer (2018)
27. García-Pedrajas, N., García-Osorio, C.: Boosting for class-imbalanced datasets using genetically evolved supervised non-linear projections. Progress in Artificial Intelligence **2**(1), 29–44 (2013)
28. Ghazikhani, A., Yazdi, H.S., Monsefi, R.: Class imbalance handling using wrapper-based random oversampling. In: 20th Iranian Conference on Electrical Engineering (ICEE2012), pp. 611–616. IEEE (2012)
29. Guo, H., Li, Y., Shang, J., Gu, M., Huang, Y., Gong, B.: Learning from class-imbalanced data: Review of methods and applications. Expert Systems with Applications **73**, 220–239 (2016)
30. Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., Bing, G.: Learning from class-imbalanced data: Review of methods and applications. Expert Systems with Applications **73**, 220–239 (2017)
31. Han, H., Wang, W.Y., Mao, B.H.: Borderline-smote: A new over-sampling method in imbalanced data sets learning. Lecture Notes in Computer Science **3644**(5), 878–887 (2005)
32. Hart, B.P.E.: ^{a}the condensed nearest neighbor rule,^{o}. In: IEEE Trans. Information Theory (1968)
33. He, H., Bai, Y., Garcia, E.A., Li, S.: Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In: IEEE International Joint Conference on Neural Networks, pp. 1322–1328 (2008)
34. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Transactions on Knowledge & Data Engineering **21**(9), 1263–1284 (2009)
35. He, H., Ma, Y.: Imbalanced learning: foundations, algorithms, and applications. John Wiley & Sons (2013)

36. Hu, S., Liang, Y., Ma, L., He, Y.: Msmote: Improving classification performance when training data is imbalanced. In: Second International Workshop on Computer Science and Engineering, pp. 13–17 (2010)
37. Ieracitano, C., Adeel, A., Gogate, M., Dashtipour, K., Morabito, F.C., Larijani, H., Raza, A., Hussain, A.: Statistical analysis driven optimized deep learning system for intrusion detection. In: International Conference on Brain Inspired Cognitive Systems, pp. 759–769. Springer (2018)
38. Jin, X.B., Xie, G.S., Huang, K., Hussain, A.: Accelerating infinite ensemble of clustering by pivot features. Cognitive Computation **10**(6), 1042–1050 (2018)
39. Jz, A., Ju, J.A., Si, C.A., Rz, A., By, B., Ql, C.: A weighted hybrid ensemble method for classifying imbalanced data. Knowledge-Based Systems **203** (2020)
40. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95-International Conference on Neural Networks, vol. 4, pp. 1942–1948. IEEE (1995)
41. Kennedy, J., Eberhart, R.C.: A discrete binary version of the particle swarm algorithm. In: 1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation, vol. 5, pp. 4104–4108. IEEE (1997)
42. Khan, F.A., Gumaei, A., Derhab, A., Hussain, A.: Tsdl: A twostage deep learning model for efficient network intrusion detection. IEEE Access (2019)
43. Krawczyk, B., Woźniak, M., Schaefer, G.: Cost-sensitive decision tree ensembles for effective imbalanced classification. Applied Soft Computing Journal **14**(1), 554–562 (2014)
44. Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: One-sided selection. Proc. Int'l Conf. Machine Learning,1997 pp. 179–186 (1997)
45. Laurikkala, J.: Improving identification of difficult small classes by balancing class distribution. In: Conference on Ai in Medicine in Europe: Artificial Intelligence Medicine, pp. 63–66 (2001)
46. Lim, P., Goh, C.K., Tan, K.C.: Evolutionary cluster-based synthetic oversampling ensemble (eco-ensemble) for imbalance learning. IEEE Transactions on Cybernetics **PP**(99), 1–12 (2016)
47. Lima, R.F., Pereira, A.C.M.: A fraud detection model based on feature selection and undersampling applied to web payment systems. In: IEEE / Wic / ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp. 219–222 (2016)
48. Lin, Z.Y., Hao, Z.F., Yang, X.W., Liu, X.L.: Several svm ensemble methods integrated with under-sampling for imbalanced data learning. In: International Conference on Advanced Data Mining and Applications, pp. 536–544 (2009)
49. López, V., Fernández, A., García, S., Palade, V., Herrera, F.: An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Information sciences **250**, 113–141 (2013)
50. Loyola-González, O., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A., García-Borroto, M.: Study of the impact of resampling methods for contrast pattern based classifiers in imbalanced databases. Neurocomputing **175**(PB), 935–947 (2016)
51. M, K.T., K., G., A, B.L.: A comparative study of filter-based and wrapper-based feature ranking techniques for software quality modeling. International Journal of Reliability, Quality and Safety Engineering **18**(4), 341–364 (2011)
52. Mahmud, M., Kaiser, M.S., Hussain, A., Vassanelli, S.: Applications of deep learning and reinforcement learning to biological data. IEEE Transactions on Neural Networks & Learning Systems **29**(6), 2063–2079 (2017)
53. Malik, Z.K., Hussain, A., Wu, J.: An online generalized eigenvalue version of laplacian eigenmaps for visual big data. Neurocomputing **173**, 127–136 (2016)
54. Mao, W., Jiang, M., Wang, J., Li, Y.: Online extreme learning machine with hybrid sampling strategy for sequential imbalanced data. Cognitive Computation **9**(6), 780–800 (2017)
55. Menardi, G., Torelli, N.: Training and assessing classification rules with imbalanced data. Data Mining & Knowledge Discovery **28**(1), 92–122 (2014)
56. Mirjalili, S.: The ant lion optimizer. Advances in Engineering Software **83**(C), 80–98 (2015)
57. Moepya, S.O., Akhoury, S.S., Nelwamondo, F.V.: Applying cost-sensitive classification for financial fraud detection under high class-imbalance. In: IEEE International Conference on Data Mining Workshop, pp. 183–192 (2015)

58. Mollineda, R.A.: On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. Knowledge-Based Systems **25**(1), 13–21 (2012)
59. Nekooeimehr, I., Lai-Yuen, S.K.: Adaptive semi-unsupervised weighted oversampling (a-suwo) for imbalanced datasets. Expert Systems with Applications **46**, 405–416 (2016)
60. Nguyen, H.M., Cooper, E.W., Kamei, K.: Borderline over-sampling for imbalanced data classification. In: Proceedings: Fifth International Workshop on Computational Intelligence & Applications, vol. 2009, pp. 24–29. IEEE SMC Hiroshima Chapter (2009)
61. Oh, S.H.: Error back-propagation algorithm for classification of imbalanced data. Neurocomputing **74**(6), 1058–1061 (2011)
62. Poria, S., Cambria, E., Howard, N., Huang, G.B., Hussain, A.: Fusing audio, visual and textual clues for sentiment analysis from multimodal content. Neurocomputing **174**, 50–59 (2016)
63. Poria, S., Peng, H., Hussain, A., Howard, N., Cambria, E.: Ensemble application of convolutional neural networks and multiple kernel learning for multimodal sentiment analysis. Neurocomputing p. S0925231217302023 (2017)
64. Precision, R.: Data mining for imbalanced datasets: An overview (2015)
65. Pérez-Godoy, M., Rivera, A.J., Carmona, C.J., Jesus, M.J.D.: Training algorithms for radial basis function networks to tackle learning processes with imbalanced data-sets. Applied Soft Computing **25**(C), 26–39 (2014)
66. R. Mohammad F.A. Thabtah, T.M.: UCI machine learning repository. http://archive.ics.uci.edu/ml (2017). Accessed 12 December, 2017
67. Ramentol, E., Caballero, Y., Bello, R., Herrera, F.: Smote-rsb*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using smote and rough sets theory. Knowledge & Information Systems **33**(2), 245–265 (2012)
68. Rayhan, F., Ahmed, S., Mahbub, A., Jani, M.R., Shatabda, S., Farid, D.M.: Cusboost: Cluster-based under-sampling with boosting for imbalanced classification (2017)
69. Ren, F., Cao, P., Li, W., Zhao, D., Zaiane, O.: Ensemble based adaptive over-sampling method for imbalanced data learning in computer aided detection of microaneurysm. Computerized Medical Imaging & Graphics **55**, 54 (2017)
70. Rosipal, R., Krämer, N.: Overview and recent advances in partial least squares. In: International Statistical and Optimization Perspectives Workshop" Subspace, Latent Structure and Feature Selection", pp. 34–51. Springer (2005)
71. Satapathy, R., Cambria, E., Hussain, A.: Sentiment analysis in the bio-medical domain: techniques, tools, and applications, vol. 7. Springer (2018)
72. Seiffert, C., Khoshgoftaar, T.M., Van Hulse, J., Napolitano, A.: Rusboost: A hybrid approach to alleviating class imbalance. IEEE Transactions on Systems Man & Cybernetics Part A Systems & Humans **40**(1), 185–197 (2010)
73. Song, L., Li, D., Zeng, X., Wu, Y., Guo, L., Zou, Q.: ndna-prot: identification of dna-binding proteins based on unbalanced classification. BMC Bioinformatics,15,1(2014-09-08) **15**(1), 298 (2014)
74. Sáez, J.A., Luengo, J., Stefanowski, J., Herrera, F.: Smote–ipf: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. Information Sciences **291**(5), 184–203 (2015)
75. Tian, Q., Han, D., Li, K.C., Liu, X., Castiglione, A.: An intrusion detection approach based on improved deep belief network. Applied Intelligence (3) (2020)
76. Tomczak, J.M.: Boosted svm with active learning strategy for imbalanced data. Soft Computing **19**(12), 3357–3368 (2015)
77. Tomek, I.: Two modifications of cnn. IEEE Transactions on Systems Man & Cybernetics **SMC-6**(11), 769–772 (1976)
78. Vluymans, S., Saeys, Y., Cornelis, C., Herrera, F.: Fuzzy rough classifiers for class imbalanced multi-instance data. Pattern Recognition **53**(C), 36–45 (2016)
79. Wajid, S.K., Hussain, A.: Local energy-based shape histogram feature extraction technique for breast cancer diagnosis. Expert Systems with Applications **42**(20), 6990–6999 (2015)
80. Wajid, S.K., Hussain, A., Huang, K.: Three-dimensional local energy-based shape histogram (3d-lesh): A novel feature extraction technique. Expert Systems with Applications **112**, 388–400 (2018)

81. Wei, M.H., Cheng, C.H., Huang, C.S., Chiang, P.C.: Discovering medical quality of total hip arthroplasty by rough set classifier with imbalanced class. Quality & Quantity **47**(3), 1761–1779 (2013)
82. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems Man & Cybernetics **SMC-2**(3), 408–421 (2007)
83. Wong, G.Y., Leung, F.H.F., Ling, S.H.: A hybrid evolutionary preprocessing method for imbalanced datasets. Information Sciences (2018)
84. Xu, J., Han, D., Li, K.C., Jiang, H.: A k-means algorithm based on characteristics of density applied to network intrusion detection. Computer Science and Information Systems pp. 14–14 (2020)
85. Yijing, L., Haixiang, G., Xiao, L., Yanan, L., Jinling, L.: Adapted ensemble classification algorithm based on multiple classifier system and feature selection for classifying multi-class imbalanced data. Knowledge-Based Systems **94**, 88–104 (2016)
86. Yu, H., Sun, C., Yang, X., Yang, W., Shen, J., Qi, Y.: Odoc-elm: Optimal decision outputs compensation-based extreme learning machine for classifying imbalanced data. Knowledge-Based Systems **92**, 55–70 (2016)
87. Zayed, A.S., Hussain, A., Abdullah, R.A.: A novel multiple-controller incorporating a radial basis function neural network based generalized learning model. Neurocomputing **69**(16-18), 1868–1881 (2006)
88. Zhao, H.: General vector machine (2016)
89. Zhou, Q., Chen, H., Zhao, H., Zhang, G., Yong, J., Shen, J.: A local field correlated and monte carlo based shallow neural network model for non-linear time series prediction. Scalable Information Systems **3**(8), e5 (2016)
90. Zhou, Q., Feng, F., Shen, Z., Zhou, R., Hsieh, M.Y., Li, K.C.: A novel approach for mobile malware classification and detection in android systems. Multimedia Tools and Applications **78**(3), 3529–3552 (2019)
91. Ziba, M., Tomczak, J.M., Lubicz, M., Witek, J.: Boosted svm for extracting rules from imbalanced data in application to prediction of the post-operative life expectancy in the lung cancer patients. Applied Soft Computing Journal **14**(1), 99–108 (2014)
92. Zikria, Y.B., Afzal, M.K., Kim, S.W., Marin, A., Guizani, M.: Deep learning for intelligent iot: Opportunities, challenges and solutions. Computer Communications **164**(0140-3664), 50–53 (2020)
93. Zou, Q., Xie, S., Lin, Z., Wu, M., Ju, Y.: Finding the best classification threshold in imbalanced classification. Big Data Research **5**, 2–8 (2016)