Expert Systems With Applications
+ OA Mirror
Manuscript Draft

Manuscript Number: ESWA-D-19-01163

Title: Differential evolution-guided resampling combined with Ada-Boost ensemble learning for imbalanced data classification

Article Type: Full length article

Keywords: Imbalanced data classification; Ensembles; Differential evolution; Resample; Ada-Boost.

Dear editor,

Please find enclosed our manuscript entitled “Differential evolution-guided resampling combined with Ada-Boost ensemble learning for imbalanced data classification”, which we would like to submit for publication as a research paper in Expert Systems with Applications. The work described has not been submitted elsewhere for publication, in whole or in part, and all the authors listed have approved the manuscript that is enclosed.

The problem of classifying imbalanced data is widespread in many fields, while skewed class distribution leads to undesirable effects on training and classification processes. Ensemble learning, especially the combination of boosting with certain preprocessing techniques emerges as an effective solution to imbalanced data classification.

We develop a novel ensemble algorithm with the guidance of differential evolution in this research. Firstly, we under-samples the majority instances whose cluster stability measured by differential evolution clustering is high. Then, the differential evolutionary strategy is embedded in the oversampling process for training base classifiers with Ada-Boost. Differential evolutionary guidance promotes the diversity of various subsets of both majority and minority class instances to train base classifiers. With experiments, the proposed method outperforms several commonly used methods focusing on imbalanced binary class problems.

Thank you for your consideration of our work.

With best regards,

Associate Professor Junguang Gao
Beijing Technology and Business University
No. 33, Fucheng Road, Haidian District, Beijing, China.
Tel: +86-18510029255
E-mail: girl_lilac@126.com

Differential evolution-guided resampling combined with Ada-Boost ensemble learning for imbalanced data classification

Ping Gong, Junguang Gao[*], Li Wang[1*], Fangkai Zhang
School of Economics and Management, Beihang University, Beijing, China, gongping_sd@163.com
Beijing Technology and Business University, Beijing, China, girl_lilac@126.com
School of Economics and Management, Beihang University, Beijing, China, wlbh@vip.163.com
School of Economics and Management, Beihang University, Beijing, China, fangkai_zhang@163.com

Abstract

The classification of imbalanced datasets has become one of the most challenging problems in data mining research. Skewed class distribution leads to undesirable effects on both the training and classification processes, and more accurate methods are needed. Ensemble learning has emerged as an effective solution. The combination of boosting with certain preprocessing techniques can enhance the classification of imbalanced data. In this study, we develop a novel ensemble algorithm with the guidance of differential evolution. Our methodology first under-samples the majority instances whose cluster stability measured by differential evolution clustering is high. Then, the differential evolutionary strategy is embedded in the oversampling process for training base classifiers with Ada-Boost. Differential evolutionary guidance promotes the diversity of various subsets of both majority and minority class instances to train base classifiers. Focusing on highly imbalanced binary class problems, we prove that the proposed method outperforms commonly used methods based on ensembles with G_{means} and AUC metrics.

Key words:

Imbalanced data classification, Ensembles, Differential evolution, Resample, Ada-Boost.

1. Introduction

Skewed class distributions hinder classification tasks (Yang & Wu, 2006). The problem of classifying imbalanced data is widespread in the fields of mechanical engineering, financial loan management, fraud detection, text classification, medical diagnosis and others (Khreich, Granger, Miri, & Sabourin, 2012) (Zhang, Islam, & Lu, 2012), and it thus has high research value and application prospects. The imbalanced data distribution phenomenon in these applications occurs when the number of samples of a certain category in the datasets is much larger than that of another category (focusing on

Supported by the National Key R&D Program of China (No. 2018YFC0807500), the Beijing Natural Science Foundation (Grant No.9162003) and the National Key Research and Development Plan (Grant No.2016YFF0202604).

the binary category problem). That is, the data presence is not balanced as the classifier expects. The rare distribution information and inner structure of the minority class may be ignored or even treated as noise. Then, the criterion, which is used to guide the training process and evaluate the classification effect, may lead to misunderstandings, resulting in low prediction accuracy by classification models. Such models thus favor the majority class, and important samples in real applications are often wrongly classified (Guo et al., 2017).

At present, research on how to improve recognition accuracy for minority samples has become popular in the field of machine learning and data mining. The related techniques commonly include three groups plus their ensemble learning approaches. The first group focuses on making improvements at the algorithm level (Di Martino, Fernández, Iturralde, & Lecumberry, 2013). The second group considers preprocessing techniques at the data level prior to the training task, such as the feature selection method to reduce the impacts of unrelated features and the resampling method to reduce the impacts on the learning process (Chawla, Bowyer, Hall, & Kegelmeyer, 2002) (Batista, Prati, & Monard, 2004). The third group considers incorporating cost-sensitive approaches to misclassification into the learning process. Ensemble learning combines the above approaches with some integration strategies to construct strong and accurate classifiers whose solutions are superior to those of non-ensemble methods (Sun et al., 2015) (Jian, Gao, & Ao, 2016). At present, research on how to improve recognition accuracy for minority samples has become popular in the field of machine learning and data mining. The related techniques commonly include three groups plus their ensemble learning approaches. The first group focuses on making improvements at the algorithm level (Di Martino, Fernández, Iturralde, & Lecumberry, 2013). The second group considers preprocessing techniques at the data level prior to the training task, such as the feature selection method to reduce the impacts of unrelated features and the resampling method to reduce the impacts on the learning process (Batista, Prati, & Monard, 2004) (Chawla, Bowyer, Hall, & Kegelmeyer, 2002). The third group considers incorporating cost-sensitive approaches to misclassification into the learning process. Ensemble learning combines the above approaches with some integration strategies to construct strong and accurate classifiers whose solutions are superior to those of non-ensemble methods (Sun et al., 2015) (Jian, Gao, & Ao, 2016).

Ensemble learning attempts to increase the accuracy of individual classifiers through combination and promotion. It has been proposed that there are ensemble approaches performing exceeding non-ensemble ones. (Polikar, 2006). Combinations of minor variants of the same classifier largely overcome the class imbalance, and their strategies are built to change the class distribution of the instances. The key factor is to ensemble and promote, to obtain higher accuracy than the individual base ones (Kuncheva, 2004). The most commonly used methods for constructing ensembles with data variation include Ada-Boost (Freund & Schapire, 1997) and Bagging (Breiman, 1996). Ensemble techniques are designed to maximize learning accuracy. To apply them to skewed class distribution problems, they are combined with preprocessing algorithms in both simple and complex ways. Their combinations with some random resampling methods, such as RUS-Boost (Seiffert, Khoshgoftaar, Van Hulse, & Napolitano, 2010), SMOTE-Boost (Chawla, Lazarevic, Hall, & Bowyer, 2003), Under-Bagging (Barandela, Valdovinos, & Sánchez, 2003) and SMOTE-Bagging (Wang & Yao, 2009), vastly outperform more elaborate techniques, and they have less implementation complexity.

To take advantage of the randomness in resampling to improve the accuracy of base classifiers, we propose combining the concept of differential evolution (DE) with the resampling process, in which our concentration is on the diversity of base classifiers, favoring the possibility of belonging to one class. This diversity mechanism is an innovation of our model, as we seek to improve the performance of the base classifiers without a trade-off with diversity, as occurs with other resampling-Boost methods. In experiments, we verified that our method outperforms several common ensemble methods on imbalanced datasets.

This study makes the following contributions. We under-sample majority class instances, considering their clustering consistency (using a DE-based clustering algorithm) before training the base classifiers, which lends more importance to the structure information of the initial sample set.

During the Boost process, we innovatively add SMOTE guided by the DE fitness function to obtain ensembles that deal with imbalanced datasets more accurately.

2. The problem of imbalanced data classification

In this section, we explore the difficulties that may occur because of the skewed distribution of data instances. We list the evaluation criteria commonly used in imbalanced distribution datasets and then discuss the common ensemble approaches (Galar et al., 2012) for later comparison.

2.1 Class imbalance problem in binary classification

For the imbalanced binary classification scenario, the majority class (negative class) is the category with a larger number of samples, while the minority class is the category with a smaller number of samples (positive class). Because of the sample size imbalance, the information contained in the majority class dominates, and the other class is underrepresented in the dataset. Ordinary classification learning algorithms usually do not perform well on imbalanced data because they are accuracy-oriented and biased toward the majority. Classification of negative instances favors the accuracy metric, while for positive instances the correct prediction more important, hindering the classification learning in real application problems.

2.2 Evaluation metric for imbalanced data distribution

In the traditional supervised learning method, the overall classification accuracy of samples is generally used as the evaluation index. However, for imbalanced datasets, the overall accuracy cannot effectively measure the quality of classifiers. For skewed two-class distributions, we evaluate the classifiers by the results for both correctly and incorrectly recognized samples in the two classes, as shown in the confusion matrix (Table 1).

Table 1. Confusion matrix

In Table 1, TP and FN denote the number of correct categorizations and the number of misclassifications in a minority class, respectively, while FP and TN denote the number of correctly classified and misclassified samples in the majority class, respectively. These four elements are used for further evaluation indexes. True positive rate: $TP_{rate} = TP/(TP+FN)$; true negative rate: $TN_{rate} = TN(FP+TN)$; false positive rate: $FP_{rate} = FP/(FP+TN)$; false negative rate: $FN_{rate} = FN/(TP+FN)$. The index G_{mean} can maximize the balance of the accuracy of the majority and minority classes. Only when the classification accuracy of both classes is high can a high value of G_{mean} be obtained.

$$G_{mean} = \sqrt{\frac{TP}{TP+FN} + \frac{TN}{TN+FP}} \tag{1}$$

The receiver operating characteristic (ROC) graphic combines the four rates to make a more comprehensive criterion that visualizes the trade-off between TP_{rate} and FP_{rate}. It is plotted with FP as the abscissa and TP as the ordinate. The area under the ROC curve (AUC) represents the probability that a classifier will rank some positive instance higher than some negative instance. The larger the AUC value, the better the current classifier performs. It is computed as the area of the ROC curve:

$$AUC = \frac{1 + TP_{rate} - FP_{rate}}{2} \tag{2}$$

2.3 Approaches to the imbalanced data classification problem

Of the three levels mentioned to solve the imbalanced data classification problem, data-level methods combined with ensemble learning strategies have shown particularly good performance. The most commonly used ensemble strategies are the Bagging, boosting and hybrid strategies.

The boosting strategy uses the whole dataset to train all classifiers, and in each iteration, data instances are given different degrees of importance. The weights of instances that are incorrectly classified in the current classification round will be updated to guide the next learning process. Ada-Boost is relatively mature and is widely used in imbalanced data classification. In this work, we adopt the Ada-Boost technique with the embedding of the resampling process guided by differential evolution measurements to optimize classification performance.

Oversampling techniques synthesize minority samples trough replication or interpolation to achieve a quantity balance of two types of samples. The synthetic oversampling of minority samples (SMOTE) method interpolates new samples, relying on the connection between the minority sample and their closest k neighbors, which to some extent makes up for the overfitting defect. Borderline-SMOTE (Agrawal, Viktor, & Paquet, 2015) determines the boundary samples of the minority classes by looking for the k nearest neighbors. It only interpolates to synthesize on the border points, which is more conducive to classification learning. A proposed instructive oversampling method found the boundary set of minority samples by introducing a clustering consistency coefficient. This method can remove noise points by using the nearest neighbor density value of the boundary samples. The SMOTE-Boost technique introduces newly generated samples into the next round of weight assignment, normalizing the weights of the new distribution.

Under-sampling methods balance the classes by reducing the number of samples in the majority class. Common methods include random under-sampling (RUS) and single-side sampling (Kang, Chen, Li, & Zhou, 2017). Using the region information, they under-sample the majority, making the samples more representative, reducing the calculation scale and improving the classification performance. RUS-Boost randomly removes examples in each iteration of Ada-Boost and renormalizes the weight distribution of the updated sample set.

Empirical research on ensemble learning values both accurate base classifiers and ensemble diversity. In Ada-Boost, the resampling is based on the weighted distribution of data, and Ada-Boost adds diversity to an ensemble learning process. The weights of the samples are modified according to the correctness of the prediction in each iteration process (Díez-Pastor, Rodríguez, García-Osorio, &

Kuncheva, 2015).

3. DERS-Boost: differential evolution-guided under-sampling and oversampling embedded in boosting

In this section, we provide a detailed description of our model to overcome the skewed class problem using resampling and the ensemble strategy. Considering that random under-sampling may cause some loss of information from the original distribution, we use the cluster stability of the majority class samples to under-sample before training the classifiers, and we then use boosting with differential evolutionary updates of the minority samples. We under-sample the majority instances whose category property are relative definite. During the boosting iteration, we use SMOTE to generate minority instances with the measurement of fitness of differential evolution. We thus mitigate the information loss problems that occur in under-sampling, enhancing the accuracy and diversity of base classifiers for the ensemble process and decreasing the computing scale of classifiers' training.

The structural diagram of the proposed model is as follows.

Fig 1. Structural diagram of the DERS-Boost model

3.1 Differential evolutionary clustering-based under-sampling

Differential evolution (DE), a simple and powerful evolutionary algorithm (Wang, Liao, Zhou, & Cai, 2014), uses three operators—mutation, crossover and selection—to complete the optimization process. In our model, the first step is to divide the initial sample set into two clusters with unsupervised DE-based clustering. To realize the clustering optimization process, we set the cluster centers as an evolving population. The goal of this differential evolution process is to maximize the distance between two clusters and minimize the distance within clusters. The initial population is randomly chosen from the sample set and presented in vectors called chromosomes to form candidate solutions to the multidimensional optimization problem.

In the differential evolution process, the target vectors are three randomly selected parent vectors from the current population that evolve through differential mutation. Scaling the difference of any two of the three parent vectors with the scaler number *F* generates the donor vector, which exchanges some of its components with the target vector in the next crossover operation to generate the trial vector according to a given rate *CR*. The evolution keeps the population size constant over subsequent generations by selection operation. Selection mechanism determines whether the trial or the target vector survives to the next generation according to their fitness. By this mechanism, the population finally gets the best fitness value, with the best individuals as cluster centers.

DE-based clustering partitions the initial samples into two clusters. With relatively few iterations, some samples in a cluster are not always stable. We distinguish the non-boundary samples by their clustering stability with reference to their real category label. Here, we define the clustering stability value of the samples as the ratio of times to be assigned to a majority cluster to the total time of clustering:

$$C_I = \frac{1}{m} \max \left\{ \sum_{t=1}^{h} \delta(\pi_t(s_i), C) \right\} \tag{3}$$

where *m* is the clustering time, $\pi_t(s_i)$ is the class label of sample s_i assigned to the t-th clustering process and *C* is the set of all possible labels. The comparison function $\delta()$ takes a value of 1 when the two parameters are equal and 0 otherwise. If a sample element is always divided into one cluster, its consistency is highly stable. Generally, in the mapping space of a dataset, the closer the sample is to the center of the cluster, the higher its clustering stability. Samples in the majority class with the clustering stability value C_I higher than a given threshold α are non-boundary samples. Here, we take α as 80% of the total clustering times. We then randomly under-sample the non-boundary points of the majority class, giving more importance to the data distribution information in the under-sample process.

3.2 Differential evolution-guided oversampling

The differential evolution-guided SMOTE technique is inspired by the application of the evolutionary prototype selection algorithm to imbalanced classification domains, where some of the features can be better fitted. Prototype selection can reduce the reference set for the nearest neighbor classifier. Its sampling process contributes to accuracy improvement and the storage necessity reduction. Facing the imbalance classification problem, the objective changes.

To use the boosting process to obtain useful subsets of the original dataset, we first use SMOTE to generate subsets of the same size *n+* of the initial minority class. The differential evolution for guiding SMOTE is represented differently from the first under-sampling part. The chromosome codifying the solution is now a binary vector that represents the presence or absence of minority instances in the dataset, including synthetic ones. We limit the search space with only synthetic instances by setting the norm of the chromosome vectors. All the initial minority class instances are always introduced into the new dataset. A chromosome vector is now represented as

$$V = (v_{x_1}, v_{x_2}, ..., v_{x_{n+}}) \tag{4}$$

Element v_{x_i} takes a value of 1 when the synthetic instance x_i is included in the dataset, and 0 otherwise. The fitness function considers the expected performance of the selected data subset and the balance between the minority and majority classes. Because G_{mean} has fewer over-fitting problems and performs well, the fitness function is defined as

$$fitness = G_{mean} - |1 - \frac{2n+}{n-} \cdot p| \tag{5}$$

where *2n+* is the size of minority samples used in training, including the initial minority class and synthetic ones; *n-* is the number of majority samples in the train; and *p* is a penalization factor that represents how important the balance between two classes is. Penalization is recommended to be 0.2.

Differential evolution here begins with a randomly initiated candidate population of *n+* dimensional binary vectors. The subsequent generations in DE are denoted as $G = 0,1,...,G_{\max}$. The chromosome vector in each generation is represented as

$$V_{i,G} = [v_{1,i,G}, v_{2,i,G}, ..., v_{j,i,G}, ..., v_{n+,i,G}],\ v_{j,i,G} \in \{0,1\} \tag{6}$$

$V_{i,G}$ means that it is the i-th chromosome vector of the G-th generation. The initial population (G=0) initializes within the search space and is constrained by the preset minimum bound $V_{\min} = \vec{0}$ and maximum bound $V_{\max} = \vec{1}$.

In the mutation process, three parent vectors $V_{r_1}, V_{r_2}, V_{r_3}$ are randomly sampled from the current population with three random mutually exclusive integers as their indices. The donor vector is obtained as

$$V_{d,G} = v_{r_1,G} + F \times (V_{r_2,G} - V_{r_3,G}) \tag{7}$$

F scales the difference of any two vectors. Here, we adopt the self-adaptive operator

$$\lambda = \exp(1 - \frac{G_{\max}}{G_{\max} + 1 - G}), F = F_0 \bullet 2^{\lambda} \tag{8}$$

F_0 is the mutation rate, and *F* varies from F_0 to $2F_0$ (specially taking F_0 to be 0.5). The mutated donor $V_{d,G}$ is mapped to binary space by a sigmoid function with a displacement:

$$sig = \frac{1}{1 + \exp(-(x - v_{j,d,G}))} \tag{9}$$

The donor vector exchanges its components with the target vector in the crossover operation, forming the trial vector $U_{i,G}$ as

$$u_{j,i,G} = \begin{cases} v_{j,d,G} & if\ rand_{i,j}[0,1] \le Cr\ or\ j = j_{rand} \\ v_{j,i,G} & otherwise \end{cases} \tag{10}$$

where j_{rand} is a randomly chosen index ensuring that the trial vector obtains at least one element from the donor vector and *Cr* is the given crossover rate. As G grows, the selection process ensures that the size of population remains constant according to the fitness value.

$$X_{i,G+1} = \begin{cases} U_{i,G} & if\ f(U_{i,G}) \le f(V_{i,G}) \\ V_{i,G} & if\ f(U_{i,G}) > f(V_{i,G}) \end{cases} \tag{11}$$

$f()$ is the fitness function defined above. Hence, selection maintains the superiority of the population.

3.3 Ada-Boost combined with DE-guided resampling

Random techniques combining accurate base classifiers with high diversity perform well in ensemble learning. The proposed DE-guided resampling method maximally considers the distribution information of the initial dataset. It improves random resampling, which may discard potentially useful samples of the majority class or lead to overlapping samples. Here, we focus on the skewed property of datasets, taking advantage of DE-guided resampling and the boost technique to resolve the classification of imbalanced data.

The training of base classifiers in an ensemble requires different training samples to ensure diversity. We benefit from the instability of differential evolution, which results from the initial randomness.

The embedding of DE-guided resampling is simple and easy to implement in the boosting algorithm. We follow the idea of the SMOTE-Boost algorithm, introducing the DE-based SMOTE process inside the Ada-Boost loop. In the Boost induction process, only the weights of the instances in the set after the DE-guided SMOTE is updated. The whole DERS-Boost procedure is outlined below.

```
Algorithm: DERS embedded in Ada-Boost
Input: Train-set $S=\{(x_i,y_i)\}$ , $i=1,2,...,N$
        $(N=n^-+2n^+)$ , $y_i\in\{1,-1\}$ ;
T: Number of iterations;
I: Base classifier;
$H_t : x\to\{-1,1\}$ ;
1.  $w_i = 1/N$
2:  $D_1(i)=(w_1,...,w_N)=(\frac{1}{N},...,\frac{1}{N})$
3: for t = 1 to T do
        $[S',H_t]=DE-SMOTE(S)$  // $H_t$ is the trained classifier used in the evolution process.
        $e_t = P(H_t(x_i)\neq y_i)=\sum_{(x_i,y_i)\in S} w_{ti} I(H_t(x_i)\neq y_i)$
        // The classification error rate considers only the weights of misclassified samples in the initial set S.
        $\varepsilon_t=\frac{1}{2}\ln(\frac{1-e_t}{e_t})$
        $D_{t+1}=\frac{D_t(i)\exp(-\varepsilon_t y_i H_t(x_i))}{Z_t}, Z_t=2\sqrt{e_t(1-e_t)}$
        $H_{final}=sign(f(x))=sign\left(\sum_{t=1}^{T}\varepsilon_t H_t(x)\right)$
    end for
Output: Boost classifier $H_{final}$ .
```

4. Experiments and Analysis

Here, we present the experiment set-up and the empirical comparison. First, we show the ensemble methods used as the base for the comparison part, and we present the details of the experimental imbalanced dataset used in the comparison. We then compare our proposal with some good performers on the class imbalance problem.

4.1 Experimental framework

The base classifier used by the ensembles is SVM, which is relatively accurate but worth improving for skewed data. Regarding the ensemble learning algorithms, we compare the proposed model with methods proved to be with good performents (shown in Table 2).

Table 2. Ensembles used in comparison

In ensemble learning, the number of base classifiers usually affects the assessed performance. We constrain all the methods tested to their fine tuning based on the dataset, to ensure that they have the equal opportunities to achieve best performance. According to empirical research, the boosting-based ensembles work optimally with 10 base classifiers. We set 10 SVM classifiers for the three boosting-based algorithms and our model. The configuration parameters used in related algorithms are shown in Table 3. All the experiments were developed using Matlab.

Table 3. Parameters set in the related algorithms

We considered six imbalanced binary datasets from the UCI machine learning repository. The properties of these datasets are summarized in Table 4. For each dataset, the number of instances (Size), the number of attributes (Atts), the size of each class, the percentage of samples in each class and the imbalance ratio (IR) are listed.

Table 4. Basic information of the experimental dataset

4.2 Experimental results and analysis

We compare our proposal with the previously presented methods and show the G_{means} test results of all the algorithms for each dataset in Table 5.

Table 5. The G_{means} values of four methods

As shown in Table 5, our proposed method performs better than all the other methods for four of the six datasets. Although it is inferior to SBO and RUSB on the Wisconsin-cancer dataset and the abalone dataset, we can see that as the IR grows on these sets, DERS-Boost is superior. It is thus important for practical application.

Fig 2. Comparison of the AUC-value of four methods

To graphically show the advantage of DERS-Boost over the other methods, we present a scatter plot in Fig 2, in which each point compares our proposal with one of the other algorithms in the dataset. The y-axis and x-axis positions are the AUC values that the four methods obtain for the six datasets. Usually, when the IR is higher, the AUC is relatively low. Our proposal outperforms four datasets in terms of the AUC value.

5. Conclusion

We present a new approach to enhancing classification results by boosting classifiers for imbalanced data classification problems. With differential evolution optimization for under-sampling and oversampling, we improve the AUC results compared to random resampling. We adopt a measure and prove that our proposal outperforms listed common methods, especially for more imbalanced datasets. In addition, with differential evolution optimization for resampling guidance, we can obtain more accurate base classifiers with more diversity, which should play an important role in ensemble learning. In future research, we will focus on the design of the fitness function to further promote the diversity and quality of the base classifiers.

Acknowledgments

This work is supported by National Key R&D Program of China (No. 2018YFC0807500), the Beijing Natural Science Foundation under Grant No.9162003, and the National Key Research and Development Plan under Grant No.2016YFF0202604.

References

Agrawal, A., Viktor, H. L., & Paquet, E. (2015, November). SCUT: Multi-class imbalanced data classification using SMOTE and cluster-based undersampling. In *2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K)*(Vol. 1, pp. 226-234). IEEE.

Barandela, R., Valdovinos, R. M., & Sánchez, J. S. (2003). New applications of ensembles of classifiers. *Pattern Analysis & Applications*, *6*(3), 245-256. https://doi.org/10.1007/s10044-003-0192-z.

Batista, G. E., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, *6*(1), 20-29. https://doi.org/10.1145/1007730.1007735.

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123-140. https://doi.org/10.1007/BF00058655.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321-357. https://doi.org/10.1613/jair.953.

Chawla, N. V., Lazarevic, A., Hall, L. O., & Bowyer, K. W. (2003, September). SMOTEBoost: Improving prediction of the minority class in boosting. In *European conference on principles of data mining and knowledge discovery* (pp. 107-119). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-39804-2_12.

Di Martino, M., Fernández, A., Iturralde, P., & Lecumberry, F. (2013). Novel classifier scheme for imbalanced problems. *Pattern Recognition Letters*, *34*(10), 1146-1151. https://doi.org/10.1016/j.patrec.2013.03.012.

Díez-Pastor, J. F., Rodríguez, J. J., García-Osorio, C., & Kuncheva, L. I. (2015). Random balance: ensembles of variable priors classifiers for imbalanced data. Knowledge-Based Systems, 85, 96-111. https://doi.org/10.1016/j.knosys.2015.04.022.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, *55*(1), 119-139. https://doi.org/

10.1006/jcss.1997.1504.

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., & Herrera, F. (2012). A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(4), 463-484. https://doi.org/10.1109/TSMCC.2011.2161285.

Guo, H., Li, Y., Shang, J., Gu, M., Huang Y., & Gong, B. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, *73*, 220-239. https://doi.org/10.1016/j.eswa.2016.12.035.

Jian, C., Gao, J., & Ao, Y. (2016). A new sampling method for classifying imbalanced data based on support vector machine ensemble. *Neurocomputing*, *193*, 115-122. https://doi.org/10.1016/j.neucom.2016.02.006.

Kang, Q., Chen, X., Li, S., & Zhou, M. (2017). A noise-filtered under-sampling scheme for imbalanced classification. *IEEE transactions on cybernetics*, *47*(12), 4263-4274. https://doi.org/10.1109/TCYB.2016.2606104.

Khreich, W., Granger, E., Miri, A., & Sabourin, R. (2012). Adaptive ROC-based ensembles of HMMs applied to anomaly detection. *Pattern Recognition*, *45*(1), 208-230. https://doi.org/10.1016/j.patcog.2011.06.014.

Kuncheva, L. I. (2004). *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 18(3), 964-964.

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and systems magazine*, *6*(3), 21-45. https://doi.org/10.1109/MCAS.2006.1688199.

Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., & Napolitano, A. (2010). RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *40*(1), 185-197. https://doi.org/10.1109/TSMCA.2009.2029559

Sun, Z., Song, Q., Zhu, X., Sun, H., Xu, B., & Zhou, Y. (2015). A novel ensemble method for classifying imbalanced data. *Pattern Recognition*, *48*(5), 1623-1637. https://doi.org/10.1016/j.patcog.2014.11.014.

Wang, J., Liao, J., Zhou, Y., & Cai, Y. (2014). Differential evolution enhanced with multiobjective sorting-based mutation operators. *IEEE transactions on cybernetics*, *44*(12), 2792-2805. https://doi.org/10.1109/TCYB.2014.2316552.

Wang, S., & Yao, X. (2009, March). Diversity analysis on imbalanced data sets by using ensemble models. In *2009 IEEE Symposium on Computational Intelligence and Data Mining* (pp. 324-331). IEEE. https://doi.org/10.1109/CIDM.2009.4938667.

Yang, Q., & Wu, X. (2006). 10 challenging problems in data mining research. International Journal of Information Technology & Decision Making, 5(04), 597-604. https://doi.org/10.1142/S0219622006002258.

Zhang, D., Islam, M. M., & Lu, G. (2012). A review on automatic image annotation techniques. *Pattern Recognition*, *45*(1), 346-362. https://doi.org/10.1016/j.patcog.2011.05.013.

***Highlights (for review)**

Research highlights

The first study to apply differential evolution in resampling for classification.

The differential evolution-based clustering keeps the distribution information.

The improved SMOTE ensures data diversity, enhancing classification performances.

Table(s)

Table 1. Confusion matrix

	Assigned to positive class	Assigned to negative class
Actual positive class	TP	FN
Actual negative class	FP	TN

Table(s)

Table 2. Ensembles used in comparison

Method	Brief description
Ada-Boost (AdaB)	SVM adaptive boosting
SMOTE-Boost (SBO)	Ada-Boost combined with SMOTE
RUS-Boost (RUSB)	Ada-Boost combined with random under-sampling

Table(s)

Table 3. Parameters for the related algorithms

Algorithm	Parameters
SMOTE	Number of k-nearest neighbors k=5, Quantity=Balance, Distance=Euclidean
DE-based US	Population size=10, F_0 =0.5, Cr=0.6, Gmax=100, Distance=Euclidean
DE-guided SMOTE	Population size=10, F_0 =0.5, Cr=0.6, Gmax=100, k=5

Table 4. Basic information of the experimental dataset

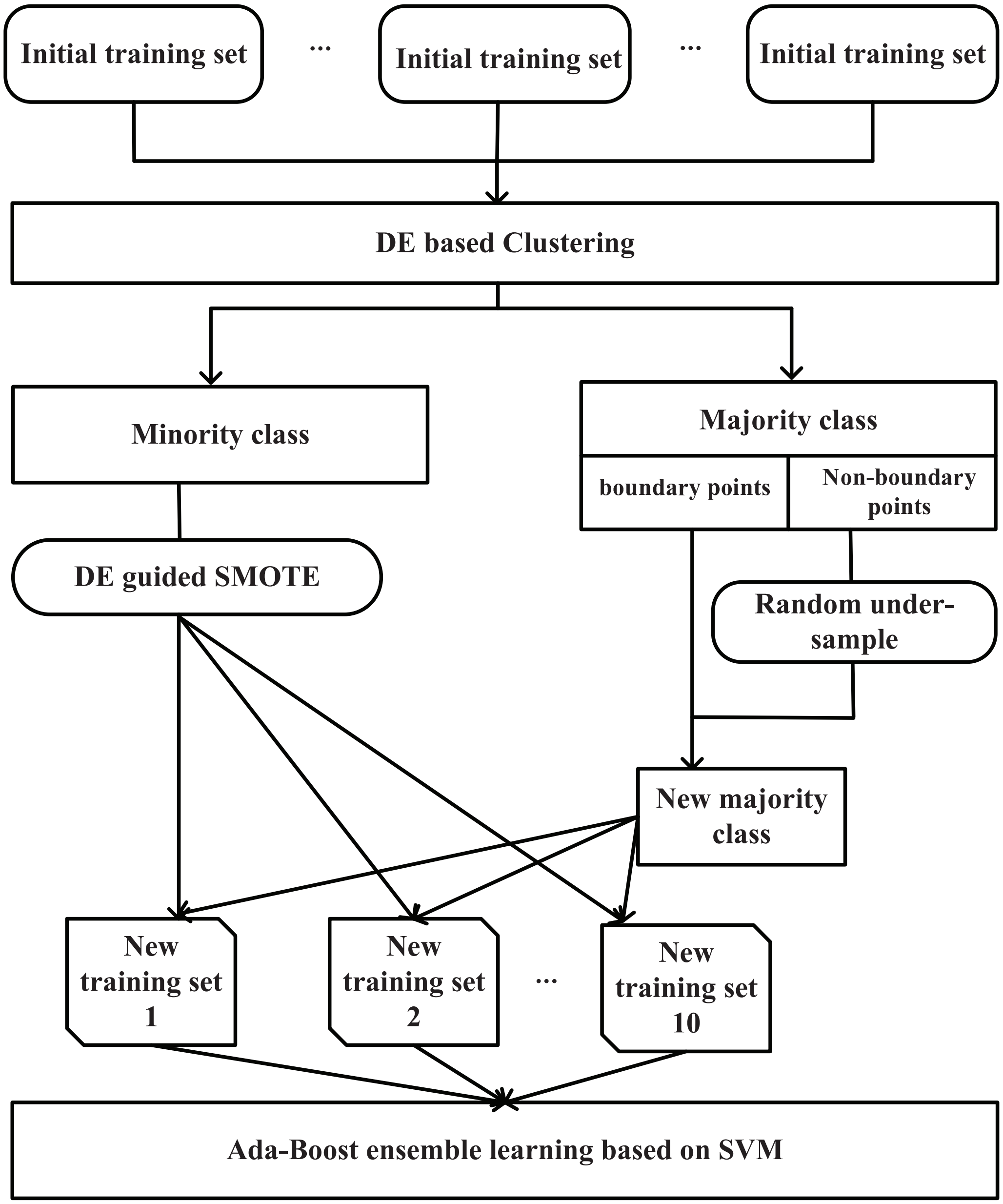
	Data set	size	Atts	(Min: Maj)	IR
1	Breast cancer	683	9	239:444	1.86
2	pima	768	8	268: 500	1.87
3	abalone	4177	8	1342: 2835	2.11
4	Credit-German	1000	24	300: 700	2.34
5	glass	214	9	51: 163	3.2
6	SPECT	187	44	15:172	11.47

Table(s)

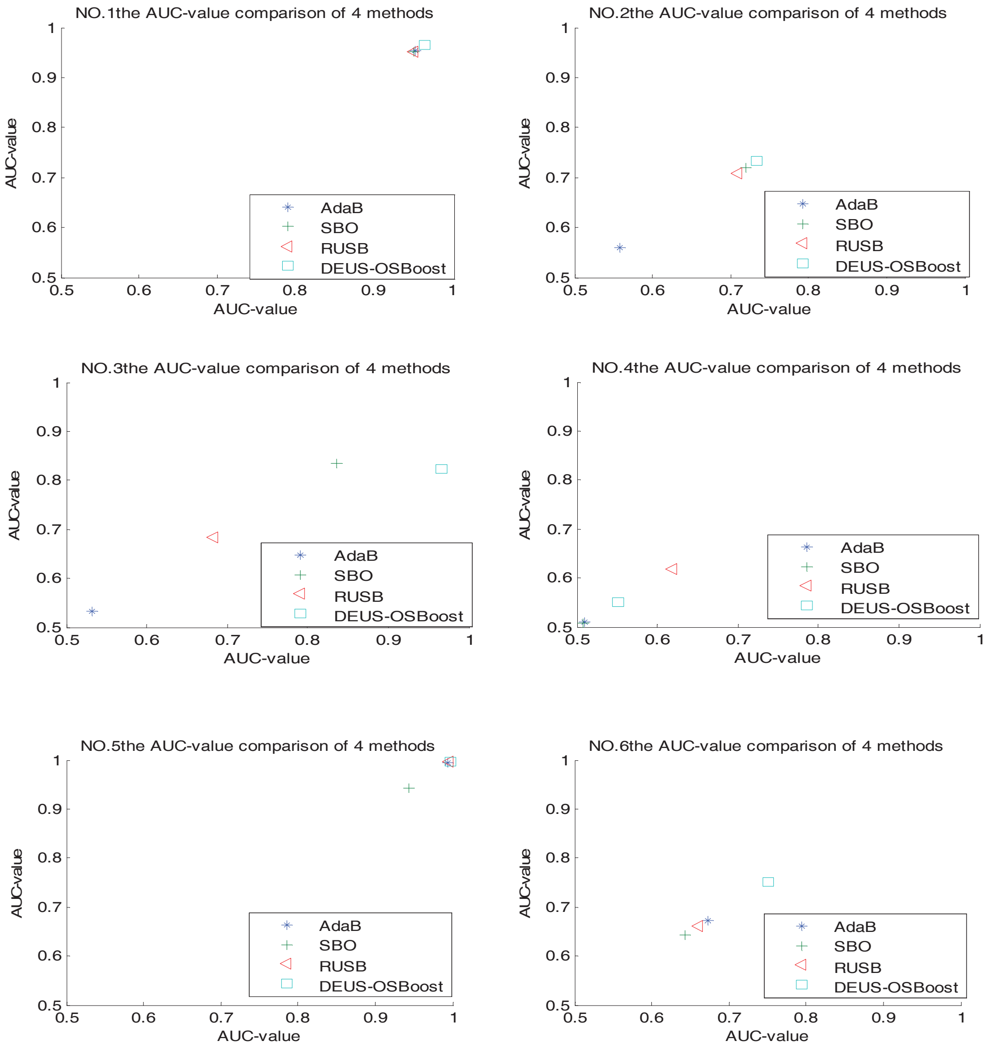
Table 5. The Gmeans values of four methods

	IR	AdaB	SBO	RUSB	DERS-Boost
1	1.86	1.3886	**1.3931**	1.3871	1.3920
2	1.87	1.1124	1.2136	1.2112	**1.2160**
3	2.11	1.0503	1.1282	**1.2833**	1.2792
4	2.34	1.0044	1.1078	1.0962	**1.1239**
5	3.20	1.3039	1.3852	1.3766	**1.3946**
6	11.47	1.0000	1.0000	1.0000	**1.2100**

Figure(1)


Initial training set
...
Initial training set
...
Initial training set
DE based Clustering
Minority class
Majority class
boundary points
Non-boundary points
DE guided SMOTE
Random under-sample
New majority class
New training set 1
New training set 2
...
New training set 10
Ada-Boost ensemble learning based on SVM

Figure(2)


NO.1the AUC-value comparison of 4 methods
AUC-value
AUC-value
0.5
0.6
0.7
0.8
0.9
1
✳ AdaB
+ SBO
◁ RUSB
□ DEUS-OSBoost
NO.2the AUC-value comparison of 4 methods
AUC-value
AUC-value
0.5
0.6
0.7
0.8
0.9
1
✳ AdaB
+ SBO
◁ RUSB
□ DEUS-OSBoost
NO.3the AUC-value comparison of 4 methods
AUC-value
AUC-value
0.5
0.6
0.7
0.8
0.9
1
✳ AdaB
+ SBO
◁ RUSB
□ DEUS-OSBoost
NO.4the AUC-value comparison of 4 methods
AUC-value
AUC-value
0.5
0.6
0.7
0.8
0.9
1
✳ AdaB
+ SBO
◁ RUSB
□ DEUS-OSBoost
NO.5the AUC-value comparison of 4 methods
AUC-value
AUC-value
0.5
0.6
0.7
0.8
0.9
1
✳ AdaB
+ SBO
◁ RUSB
□ DEUS-OSBoost
NO.6the AUC-value comparison of 4 methods
AUC-value
AUC-value
0.5
0.6
0.7
0.8
0.9
1
✳ AdaB
+ SBO
◁ RUSB
□ DEUS-OSBoost