Knowledge-Based Systems 94 (2016) 88–104

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Adapted ensemble classification algorithm based on multiple classifier system and feature selection for classifying multi-class imbalanced data

Li Yijing [a,b], Guo Haixiang [a,b,c,d,*], Liu Xiao [a,b], Li Yanan [a,b], Li Jinling [a,d]

[a] College of Economics and Management, China University of Geosciences, Wuhan 430074, China
[b] Research Center for Digital Business Management, China University of Geosciences, Wuhan 430074, China
[c] Mineral Resource Strategy and Policy Research Center of China University of Geosciences, Wuhan 430074, China
[d] The Joseph M. Katz Graduate School of Business, University of Pittsburgh, Pittsburgh, PA 15260, USA

ARTICLE INFO

Article history:
Received 14 May 2015
Revised 13 November 2015
Accepted 16 November 2015
Available online 1 December 2015

Keywords:
Imbalanced data
Multiple classifier system
Adaptive learning
Oil reservoir

ABSTRACT

Learning from imbalanced data, where the number of observations in one class is significantly rarer than in other classes, has gained considerable attention in the data mining community. Most existing literature focuses on binary imbalanced case while multi-class imbalanced learning is barely mentioned. What's more, most proposed algorithms treated all imbalanced data consistently and aimed to handle all imbalanced data with a versatile algorithm. In fact, the imbalanced data varies in their imbalanced ratio, dimension and the number of classes, the performances of classifiers for learning from different types of datasets are different. In this paper we propose an adaptive multiple classifier system named of AMCS to cope with multi-class imbalanced learning, which makes a distinction among different kinds of imbalanced data. The AMCS includes three components, which are, feature selection, resampling and ensemble learning. Each component of AMCS is selected discriminatively for different types of imbalanced data. We consider two feature selection methods, three resampling mechanisms, five base classifiers and five ensemble rules to construct a selection pool, the adapting criterion of choosing each component from the selection pool to frame AMCS is analyzed through empirical study. In order to verify the effectiveness of AMCS, we compare AMCS with several state-of-the-art algorithms, the results show that AMCS can outperform or be comparable with the others. At last, AMCS is applied in oil-bearing reservoir recognition. The results indicate that AMCS makes no mistake in recognizing characters of layers for oilsk81-oilsk85 well logging data which is collected in Jianghan oilfield of China.

1. Introduction

Classification is one of the crucial issues in the field of machine learning. Classical classifiers such as Decision Tree, Naïve Bayes, Artificial Neural Network (ANN), K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) operate under the assumption that data sample contains a faithful representation of the population of interest, which means a balanced sample distribution is required [1]. When facing skewed class distribution, the traditional classifiers often come up to a disappointed performance [2–4]. Imbalanced data refers to such a dataset in which one or some of the classes contain much more samples in comparison to the others. The most prevalent class is called the majority class, while the rarest class is called minority class. Imbalanced situation often occurs in real word applications like fraud detection, disease diagnoses, financial risk analysis, etc. [5,6]. When addressing imbalanced data problems, people tend to care more about the minority class, for the reason that the cost of misclassifying minority samples are much higher than the others [2,6,7]. Taking cancer diagnoses for example, the number of cancer patients is much less than healthy people, if cancer patients are diagnosed as healthy people, they will exceed the best therapy time, which may cause a serious medical incidence [6]. So does oil-bearing recognition that is studied in this paper. Oil-bearing recognition refers to recognize the characters of each layer in the well [8,9], the class distribution of logging data is skewed and cost of misclassifying oil layer is much higher than other misclassification situations. Therefore, oil-bearing recognition is a typical imbalanced data classification problem.

Imbalanced learning is a well-studied problem, dozens of sampling methods [10,11], cost sensitive algorithms [17,18], one-class classifiers [53,54,57] have been proposed in literature. More recently, ensemble learning becomes a popular solution of addressing

* Corresponding author at: College of Economics and Management, China University of Geosciences, Wuhan 430074, China. Tel.: +86 15927389298; fax: +86 027 67883201.

E-mail addresses: liyijing024@hotmail.com (L. Yijing), faterdumk0732@sina.com (G. Haixiang).

http://dx.doi.org/10.1016/j.knosys.2015.11.013
0950-7051/

imbalanced data. A common way for constructing ensemble learning model for imbalanced data is based on sampling methods, that is, employing sampling methods as a pre-process to generate several balanced datasets and training different base classifiers independently. The main idea of constructing ensemble learning model is to learn from data by multiple classifiers, thus a designed ensemble learning model can also be viewed as Multiple Classifier System(MCS) [47].Since ensemble learning has been proved to be the most efficient way to tackle imbalanced learning problems [1,15,16,47,12], we aim to focus on constructing ensemble model in this paper.

Though various MCSs have been proposed, most of them model different types of data consistently and train a universal ensemble classifier to address all imbalanced data. In fact, using a specific ensemble classifier to tackle all kinds of imbalanced data is inefficient. The learning quality of a model can be affected by the choices of sampling methods, base classifiers, and final ensemble rules. For example, when the samples of minority class are extremely rare (saying we just have 1 or 2 minority samples), under-sampling methods may not be valid since we need to abandon tremendous number of majority samples to construct a balanced training set. The same concern should be highlighted when deciding which base classifier to use. In many previous work, the authors tested several base classifiers such as SVM, Naïve Bayes, CART in their model, but just the overall performances of different classifiers have been pointed out [15,25,40,47]. However, performances of different classifiers may vary in characteristics of datasets. For example, CART may perform well in datasets with high Imbalance Ratio(IR), but come up to a disappointed performance when classifying low dimension datasets. More specifically, IR, the number of features, the number of classes are all crucial factors that have to be considered when applying base classifier into the ensemble model. Therefore, in this paper, we divide imbalanced data into eight types based on their IR, dimension (the number of features) and the number of classes. We attempt to conduct an adaptive ensemble algorithm that is able to learn from different types of imbalanced data by different yet most efficiency algorithms constructed from a union ensemble paradigm.

While most MCSs take sampling methods as pre-processing, few literature has considered another common pre-processing technique, that is, feature selection. Feature selection is often separated as another issue for imbalanced learning, as is discussed [5,49] and [50]. These studies focus on developing novel feature selection algorithms, while the contribution of feature selection for imbalanced data classification is not clearly discussed. It is obvious that removing irrelevant and redundant features reduces the noise in the training space and decrease the time complexity [20,21]. For imbalanced case, samples of minority class are more easily to be ignored as noise, if we remove the irrelevant features in the feature space, the risk of treating minority samples as noise may also be reduced. [47,14] employed feature selection algorithm as a pre-processing procedure before carrying out classification, which gained good results. This motivates us to employ both feature selection and sampling method as pre-processes before training MCS.

Multi-class classification has been pointed out as a hard task for classification [40,19], due to that multi-class classification might achieve a lower performance than binary classification as the boundaries among the classes may overlap. This issue may become more complex when facing imbalanced data. In [40] the authors studied two ways of extending binary classification algorithms into multi-class case: One-versus-one approach (OVO) and One-versus-all approach (OVA). The conclusion, as they suggested, is OVO approaches gain better accuracy than OVA approaches. However, when considering computational complexity, OVO approaches may sacrifice too much on time cost when the number of classes increases. In their empirical study, OVA approaches also outperformed OVO approaches in some cases, which implies that there is no dogmatic approach that suit for all kinds of imbalanced data. It should be noted that the training of the OVA approach is inherently imbalanced, as the set of all data points from all other classes is likely to outnumber the representatives of the target class, for each sub-classifier [19]. Taking this into account, OVA approach may not suitable for high IR datasets. The third option of addressing multi-class imbalanced data is standard ad-hoc learning algorithms (algorithms that are natural for addressing multiple class learning problems), such as KNN, Naïve Bayes based ensemble algorithms. In our study, we specifically focus on multi-class imbalanced data. In order to build adaptive ensemble algorithm for different kinds of imbalanced data, OVO, OVA approaches and ad-hoc approaches will all be considered and we expect to find criteria to select the best approach for each type of data.

We argue that the above mentioned concerns are crucial issues that need to be clarified. Therefore, in our study, we attempt to build an adaptive ensemble learning algorithm for multi-class imbalanced data, which is called Adaptive Multiple Classifier System(AMCS). For adaptive learning, Three widely-accepted ensemble frameworks are considered, that are, Adaboost.M1 [46], Under-Sampling Balanced Ensemble(USBE) [15,47] and Over-Sampling Balanced Ensemble(OSBE) [16]. For the later two frameworks, five different ensemble rules (such as Max, Min, Product etc. shown in Table 2) to fuse sub-classifiers are optional. Moreover, as feature selection might avail to reduce the risk of treating minority samples as noise, in all the ensemble frameworks feature selection is employed as a pre-processing, for which both wrapper and filter feature selection techniques are considered. In empirical study, we first test the three ensemble frameworks with different ensemble rules and base classifiers, then conclude the adaptive criteria for different types of imbalanced data. Finally, we apply AMCS to oil-bearing reservoir recognition adaptively base on the characteristic of Jianghan well-logging data. Four significant contributions of our study are as follows:

(1) We present a comprehensive categorization of several recent works related to imbalanced data classification and highlight the need for an adaptive algorithm to solve different kinds of imbalanced data. To do so, we categorize imbalanced data into eight types based on their IR, dimension and the number of classes. For each type of data, the order of choosing feature selection algorithm, ensemble framework, base classifier and ensemble rule can be viewed as a *route* of framing a MCS, our algorithm can choose the best route for different types of data.
(2) The proposed ensemble method AMCS employs both feature selection and sampling method as pre-processes, in which feature selection may be an option when there is no irrelevant or redundant feature exits.
(3) We focus only on multi-class imbalanced data problems, which may be ignored by many previous studies. Since most classical performance metrics such as AUC are binary metrics, we enable a novel multi-class AUC metric called AUCarea to evaluate models by setting probabilistic outputs for both base classifiers and ensemble classifiers.
(4) The goodness of this novel adaptive methodology and the criteria of choosing routes to form AMCS are studied by means of thorough experimental analyses. Each node of the route is selected in a selection pool. The selection pool contains two feature selection methods, three ensemble frameworks, five base classifiers and five ensemble rules. In empirical study, all the possible routes for eight types of data are tested, the best routes for each type of data is selected as AMCS. The superior of AMCS compared with several existing methods is tested using various benchmarks and a real-world case of oil reservoir recognition.

The remainder of this paper is organized as follows. Literature related to imbalanced data are categorized in Section 2. The main framework of AMCS is described in Section 3, where the two feature selection methods, three ensemble frameworks and five ensemble rules which we use to form AMCS are introduced. Section 4 presents the classical performance measures and the new metric AUCarea, in which the superiority of AUCarea for measuring the performance of algorithms for multi-class classification task is clarified. In the empirical study of Section 5, we first test all possible choices of constructing AMCS for different types of imbalanced data and discuss the adaptive criteria, then the AMCS is compared with several state-of-the-art algorithms via various metrics including AUCarea. In Section 6 we apply AMCS to oil-bearing reservoir recognition. At last, conclusions are placed in Section 7.

2. Related work

Methods addressing the class imbalance problems can be categorized into three groups: data-level methods, algorithmic-level methods and ensemble methods.

2.1. Data level approaches

Data level approaches focus on re-sizing the training datasets in order to balance all kinds of classes. Two main ideas of re-sizing are over-sampling and under-sampling. Over-sampling methods eliminate class imbalance by creating new minority class samples while under-sampling methods re-balance imbalanced class distribution by reducing the number of majority class samples. Dozens of over-sampling and under-sampling algorithms have been proposed previously. The most famous over-sampling algorithm is called SMOTE proposed by Chawla [10]. More recently, other under-sampling methods based on SMOTE have been discussed, such as bSMOTE [22] and V-synth [23]. For under-sampling methods, Random Under-Sampling (RUS) has been utilized in many previous work [1,25,12].Dehmeshki proposed a data filtering technology [11] to categorize the majority samples into safe region, borderline region and noise region, just samples from safe region are used as training samples. [26] presented a novel under-sampling method based on ant colony optimization algorithm.

2.2. Algorithmic level approaches

Algorithmic level approaches focus on modifying existing classification algorithms to strengthen their ability of learning from minority class [8,9]. Most of algorithms in this family are based on SVM and Neural Network. Ando [36] proposed a straightforward yet effective class-wise weighting scheme called SNN based on k-Nearest neighbor density model to cope with class imbalance issue. The main concern of SNN is to employ an adjusted k-radius to compensate the sparseness of the minority class. In [38] a wavelet support vector machine (WSVM) is presented. Concerning imbalanced scenarios, a filter feature selection technique is performed to remove the redundant and irrelevant information. Datta and Das [19] emphasized that the classical SVM moves the separating hyper-plane towards the minority class as majority class is more likely to dominate the region of overlap. By this motivation, they proposed a Near-Bayesian Support Vector Machine (NBSVM) that utilizes Bayesian posterior probabilities to achieve the boundary shift as well as the unequal regularization costs. Pérez-Godoy et al. [44] compared the performance of three types of modified RBF Networks(RBFN) in imbalanced data classification, that are, clustering based RBFN, incremental RBFN, evolutionary based RBFN, among which both Least Mean Square and the Singular Value Decomposition are considered in weights training phase.

Cost sensitive algorithms attempt to increase the learning ability of classifiers by assigning larger misclassifying cost for minority class samples. López et al. [17] proposed an algorithm to deal with large-scale imbalanced data using a fuzzy rule and cost-sensitive learning techniques. Krawczyk et al. [18] constructed a fusion algorithm based on cost-sensitive decision tree ensembles, in which the choice of cost matrix is estimated by ROC analysis. Nguyen et al. [55] introduced two empirical cost-sensitive algorithms, one combined sampling, cost-sensitive and SVM and the other treated the cost ratio as a hyper-parameter which needs to be optimized before training the final model. Another concept of learning from imbalanced data is treating minority samples as outliers and analogizing technologies of detecting noises and outliers to model minority class, such as one-class classifier [53,54].

2.3. Ensemble methods

Classifiers ensemble is regarded as a popular technology to tackle imbalanced learning, mainly due to their ability to significantly improve the performance of a single classifier [48]. Ensemble methods can be viewed as building multiple classifier system that combines a variety of base classifiers, for each base classifier data-level approaches are often employed as a pre-processing. The most widely used MCS is boosting algorithm proposed by Schapire [13], which has been applied in many well-known ensemble algorithms such as SMOTEBoost [16], RUSBoost [51], EasyEnsemble [25], EUSboost [1]. [63] denoted that a typical MCS generally contain 3 processes, that are, resampling, ensemble building, and fusion rule. Sun et al. [15] proposed a novel ensemble strategy for imbalanced data classification, this strategy converts an imbalanced dataset into multiple balanced subset, for each subset a base classifier is trained, similar strategy can be found in [12]. Krawczyk et al. [47] created an ensemble algorithm called PUSBE that contains sampling, pruning and boosting technologies. [52] first divided the data into non-overlapped region, borderline region and overlapped region and then trained different regions by different classifiers. The imbalance situation of different amount of data at the overlapped region and non-overlapped region is concerned. Zięba, and Tomczak [56] proposed a boosted SVM, in which an active strategy of selecting the borderline examples to train each SVM is designed. In this way, each training set used to construct the basic classifier is more balanced and noiseless.

2.4. Other issues related to class-imbalance learning

More recently, other significant problems related to data skewed characteristics have been taken into account, which include validation methods, performance metrics and data shift problems. Wallace, and Dahabreh [58] proposed a new metric named of stratified Brier score to capture class-specific calibration in imbalanced scenarios. The difference between the observed label and the estimated probability is measured to eliminate the underestimate of the probabilities for minority class instances. López et al. [59] focused on the problem of dataset shift which is defined as the case where training and testing data follow different distributions. A new validation technique was studied in [59] to avoid dataset shift issue when using k-ford cross-validation. The new validation approach first picks a random unassigned example, then finds its k-nearest unassigned neighbors of the same class and assigns each of those neighbors to a different fold, by this mean, it guarantees a consistent data distribution inside each fold. Song [64] introduced three strategies to select the operating point used for ROC, which is more suitable for evaluating imbalanced data. In their study, they tried to reveal the shifted-decision value by maximizing/minimizing a function of sensitivity and specificity in the ROC space. In [65] the rule based learning for imbalanced data was achieved by letting the experts annotate some of the “hard” learning examples. In their study, a specific method is proposed for identifying the examples which should be explained by an expert. Then, ABCN2 algorithm is used to induce a new set of rules.

Instead of proposing new methods, Ronaldo et al. [37] conducted a series of experiments to assess the performances of some proposed treatment methods like SMOTE [10], ADASYN [38] and MetaCost [39] for imbalanced data. These treatment methods are all based on sampling technologies and cost sensitive learning. In their research, they defined a value called “performance loss” to figure out whether all the learning models are equally affected by class imbalanced. Besides, they also defined a metric called “performance recovery” to evaluate how much of the performance losses caused by imbalanced distribution can be recovered by the treatment methods. The results showed not all the treatment methods are suit for all basic algorithms. For example, SMOTE are considered as the most common sampling method for imbalance data, but it seems to harm the performance of SVM and Naïve Bayes. The same study was presented in [28], though some ensemble models are dominating over the others in an overall scenario, they vary in different kinds of datasets. These studies inspired us that specified algorithm may not be an eternal solution for imbalanced learning, which implies an adaptive learning must be taken into account in order to provide a better solution.

3. AMCS: Adaptive multiple classifier system

Multiple classifier system, or ensemble classifier, has been certified to be more robust and effective compared with individual classifier. For building a MCS, one common method is to perform a data distribution shift on training space, which can be accomplished by resampling samples from training space to construct sub-training set for each base classifier. Data sampling approaches attempt to alleviate the problem of class imbalance by either removing examples from the majority class (under-sampling) or adding examples to the minority class (over-sampling) [51]. Another resampling idea is applied in Adaboost [24] called FiltEX. In [47], the authors listed three issues have to be considered when building a MCS, that are: choice of base classifiers, choice of ensemble rules and choice of pruning classifiers. In our study, we argue that the choice of feature space may also play a crucial role when building MCS for imbalanced data. Taking all these considerations into account, we attempt to build an Adaptive Multiple Classifier System(AMCS) that is able to select the base classifier, resampling method, ensemble rule and feature selection method adaptively based on different characteristic of imbalanced data. The AMCS is discussed in this section. First, we introduce three well-used resampling based ensemble framework in Section 3.1, namely Adaboost, Under-Sampling Balanced Ensemble(USBE) and Over-Sampling Balanced Ensemble(OSBE). After that, we describe two feature selection algorithms we considered in this paper in Section 3.2. The five ensemble rules are presented in Section 3.3. At last, the AMCS is shown in Section 3.4.

3.1. Three types of multiple classifier system

(1) Adaboost.M1
Boosting is regarded as the most common and effective method in ensemble learning. The first applicable approach of boosting is Adaboost proposed by Schapire and Freund [24].The basic Adaboost is implemented for binary classification problems, Adaboost works by sequentially applying a weak classifier to train the reweighed training dataset (generated by a distribution D) and taking majority vote as the ensemble rule to fuse all weak hypotheses. In each iteration, the sample distribution D is updated according to the hypothesis generated in each iteration. The goodness of Adaboost lies in, samples that failed to be assigned to the correct class gain higher weights, so that in the next iteration the classifier will focus more on learning those failed classified samples.
To solve different kinds of classification problems, AdaBoost.M1, AdaBoost.M2 [46], AdaBoost.MR, AdaBoost.MH [45] have been proposed as the extensions of basic Adaboost algorithm, in which AdaBoost.M1, AdaBoost.M2 are used to solve multi-class with single-label problems, the later two are used to solve multi-class with multi-label problems. Since in this paper we aim at multi-class data with single-label, so that Adaboost.M1 algorithm is chosen as the first ensemble model. The detail description of Adaboost.M1 can be found in [46].
Adaboost is designed for use with any learning algorithms, the main concern is that the training samples now have varying weights [24]. There are two main approaches to address these weights: boosting by resampling and boosting by reweighting [46]. As some classifiers cannot be generalized to use a given distribution directly, we choose to use boosting by resampling in our model. The resample algorithm is called FiltEX, which is described in algorithm 1, the main idea can be found in [13].

Algorithm 1
FiltEX algorithm.

1	Choose a real number x uniformly at random in the range $0 \le x < 1$
2	Perform a binary search for the index j for which $\sum_{i=1}^{j-1} w_i \le x < \sum_{i=1}^{j} w_i$
3	Return the example (x_i, y_i)

(2) Under-Sampling Balanced Ensemble(USBE)
An USBE model refers to employing under-sampling methods to build several roughly balanced training sets for multiple base classifiers. The training procedure can be described as Fig. 1. One of the most common yet simply under-sampling techniques is Randomly Under-Sampling (RUS). Unlike more complex data sampling algorithms, RUS makes no trail to “intelligently” select samples from the training space. Instead, RUS simply select samples from the majority classes randomly until a satisfied class distribution is achieved. In this paper, we utilize RUS as the under-sampling method. Note that we are facing multi-class dataset, the RUS method is employed in all the classes that contain more than 1.5 times as number of samples as in the rarest class, by this way, we guarantee that the IR of training set after RUS is less than 1.5 and do not abandon too much information of majority classes.

(3) Over-Sampling Balanced Ensemble(OSBE)
Similar as USBE, an OSBE model can also be viewed as Fig. 1, instead the over-sampling method is employed to build balanced sub-training sets. The most widely-used over-sampling method is SMOTE proposed by Chawla [10]. SMOTE generates synthetic samples by a linear interpolation between two neighbors from minority class. Neighbors of minority class are randomly chosen, and the number of additional samples depends on the required over-sampling amount. Since SMOTE makes no assumption on original data probability distribution when generating additional samples, the constructed classifiers may be unbiased ones. Therefore, for the third ensemble framework, we choose SMOTE as the sampling technique. Similar as RUS, SMOTE is employed in all the rare classes and generates new examples for those classes until the IR of training set is less than 1.5.

3.2. Feature selection based MCS

Samples of different classes may overlap, which makes it more difficult for classifiers to find the boundaries among different classes. When facing imbalanced data, the samples of minority class are scarce so that they might be treated as noise easily. In that case, if the boundaries of different classes are ambiguous, the classifier may fail to model the minority class samples. Feature selection is a powerful method to solve this problem. We take the logging data which is collected from Jianghan oil field of China to illustrate this observation. We select three features of logging data and plot the sample distribution, which is shown on the right side of Fig. 2, where the diamonds are the oil layer samples and the circles are non-oil layer samples. It is clear that the boundary between these two classes is ambiguous. On the left side of Fig. 2 we present a 2-D sample distribution by

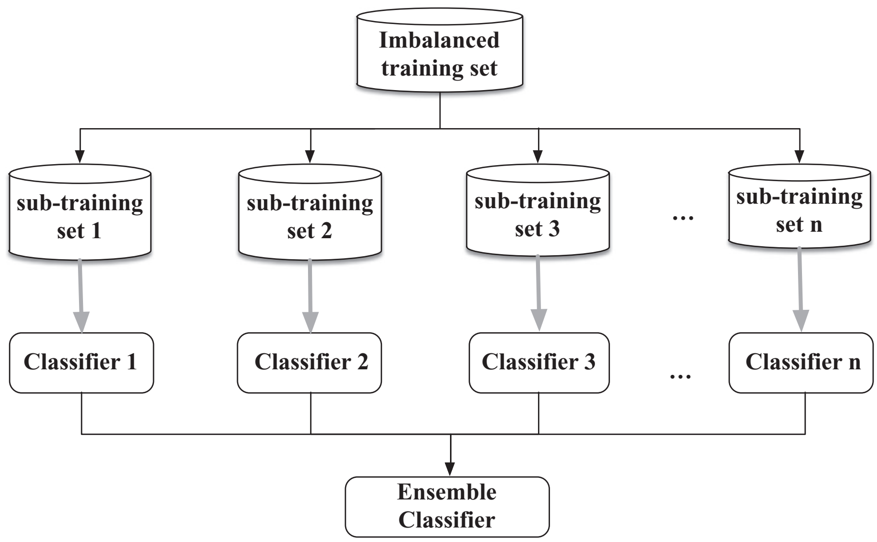

Fig 1. USBE model.

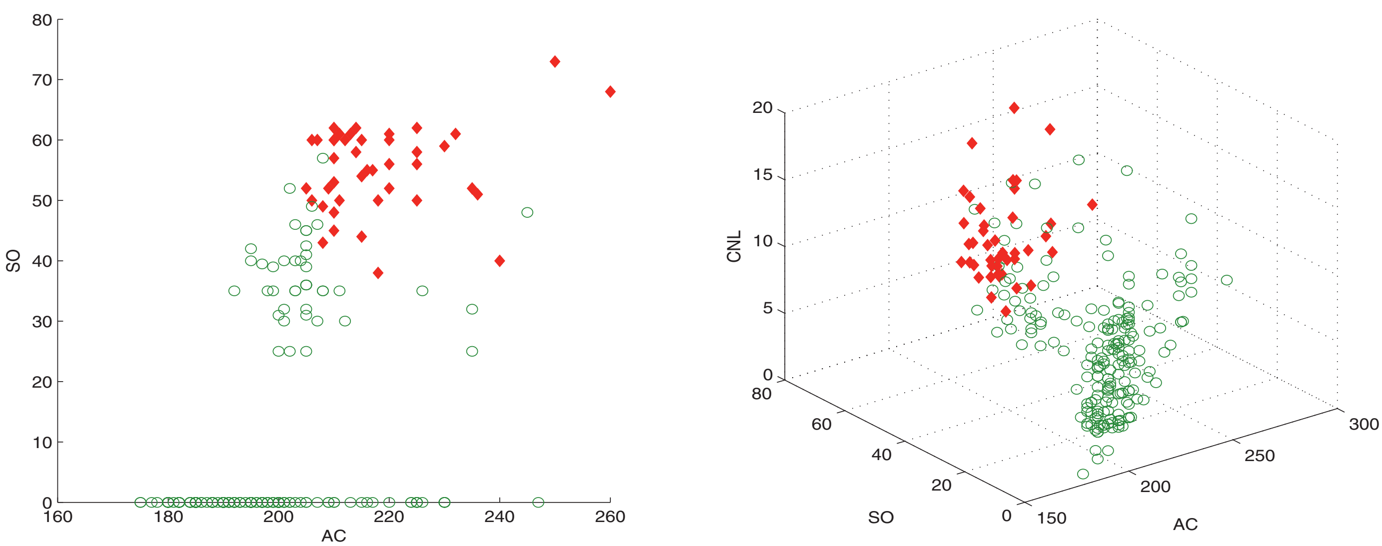

Fig. 2. Sample distribution before and after removing irrelevant feature.

removing one of the three features (CNL). It can be observed that the boundary is much clearer.

When employing feature selection to imbalanced data, a main challenge is the trade-off between removing the irrelevant features and keeping useful features. Similar to re-sampling technologies, remove somewhat irrelevant features may also a risk of losing potentially useful information because the original data distribution may be altered by feature selection [15]. The feature selection algorithms can be split as two types, wrapper methods and filter methods. From our preliminary observation, the performances of different feature selection algorithms vary in the choice of base classifier. For this reason, instead of utilizing specific feature selection algorithm, we select two types of feature selection algorithms as optional in AMCS. We choose FCBF algorithm [66] as filter method and a meta-heuristic algorithm BPSO as wrapper method. Addressing feature selection through a meta-heuristic method is one of a worthwhile way for feature selection. Particle Swarm Optimization (PSO) is a widely used stochastic evolutionary algorithm for solving optimization problems, which is devised by Kennedy and Eberhart in 1995 [32]. Unlike other evolutionary algorithms (such as GA, DE, etc.), PSO does not contain crossover and mutation operations, which can help us reduce the complexity of our model. Basic PSO is proposed as an optimization technique applied in real space [33], while Kennedy and Eberhart proposed Binary Particle Swarm Optimization (BPSO) algorithm [34] that extending PSO to binary space case. A brief introduction of using BPSO as feature selection method can be found in supplementary material.

3.3. Ensemble rule

After training an MCS, for a new unseen sample, several individual hypotheses can be obtained from each classifier. Thus, an ensemble rule that combining these hypotheses is required. In [15] and [67], five ensemble rules for combining the multiple classification results of different classifiers are introduced, including Max Rule, Min Rule, Product Rule, Majority Vote Rule and Sum Rule. Consider a dataset S that contains D features, N samples and m classes, $S = \{s_1, s_2, \dots, s_N\}$ where $s_i = \{x_{i1}, x_{i2}, \dots, x_{iD}\}$, the class labels $C = \{C_1, C_2, \dots, C_m\}$. Suppose that there are K base classifiers in an MCS, for the ith classifier, it classifies the new data as $C_j$with the probability of p_{ij}. Moreover, RC_jdevotes the final probability of classifying the new data to class c_j. Table 1 shows the detailed ensemble strategies and descriptions for the five ensemble rules.

Table 1
Five ensemble rules.

Rule	Strategy	Description
Max	$\underset{C_1,C_2,\ldots,C_m}{\text{argmax}}\{R_{C_1}, R_{C_2}, \ldots, R_{C_m}\}, R_{C_j} = \max_{1\le i\le K} p_{ij}$	Use the maximum classification probability of K base classifiers for each class label, the new data is assigned to the class with maximum R
Min	$\underset{C_1,C_2,\ldots,C_m}{\text{argmax}}\{R_{C_1}, R_{C_2}, \ldots, R_{C_m}\}, R_{C_j} = \min_{1\le i\le K} p_{ij}$	Use the minimum classification probability of K base classifiers for each class label, the new data is assigned to the class with maximum R
Product	$\underset{C_1,C_2,\ldots,C_m}{\text{argmax}}\{R_{C_1}, R_{C_2}, \ldots, R_{C_m}\}, R_{C_j} = \Pi_{i=1}^{K} p_{ij}$	Use the maximum classification probability of K base classifiers for each class label, the new data is assigned to the class with maximum R
Majority vote	$\underset{C_1,C_2,\ldots,C_m}{\text{argmax}}\{R_{C_1}, R_{C_2}, \ldots, R_{C_m}\}$ $R_{C_j} = \underset{1\le i\le K}{\text{count}}(\underset{C_1,C_2,\ldots,C_m}{\text{argmax}}\, p_{ij})$	For the *i*th classifier, C_j gets a vote if p_{ij} is the largest probability of C, count all the votes of C_j in the K classifiers as R_{C_j}. The new data is assigned to the class with maximum R
Sum	$\underset{C_1,C_2,\ldots,C_m}{\text{argmax}}\{R_{C_1}, R_{C_2}, \ldots, R_{C_m}\}, R_{C_j} = \underset{1\le i\le K}{\text{sum}}\, p_{ij}$	Use the sum classification probability of K base classifiers for each class label, the new data is assigned to the class with maximum R

Table 2
Five weighed ensemble rules.

Rule	Strategy
Weighted max	$\underset{C_1,C_2,\ldots,C_m}{\text{argmax}}\{R_{C_1}, R_{C_2}, \ldots, R_{C_m}\}, R_{C_j} = \max_{1\le i\le K} AUCarea_i \cdot p_{ij}$
Weighted min	$\underset{C_1,C_2,\ldots,C_m}{\text{argmax}}\{R_{C_1}, R_{C_2}, \ldots, R_{C_m}\}, R_{C_j} = \min_{1\le i\le K} AUCarea_i \cdot p_{ij}$
Weighted product	$\underset{C_1,C_2,\ldots,C_m}{\text{argmax}}\{R_{C_1}, R_{C_2}, \ldots, R_{C_m}\}, R_{C_j} = \Pi_{i=1}^{K} AUCarea_i \cdot p_{ij}$
Weighted majority vote	$\underset{C_1,C_2,\ldots,C_m}{\text{argmax}}\{R_{C_1}, R_{C_2}, \ldots, R_{C_m}\}, R_{C_j} = \underset{1\le i\le K}{\text{count}}(\underset{C_1,C_2,\ldots,C_m}{\text{argmax}}\, AUCarea_i \cdot p_{ij})$
Weighted sum	$\underset{C_1,C_2,\ldots,C_m}{\text{argmax}}\{R_{C_1}, R_{C_2}, \ldots, R_{C_m}\}, R_{C_j} = \underset{1\le i\le K}{\text{sum}}\, AUCarea_i \cdot p_{ij}$

The ensemble rules described in Table 1 rely only on the posterior probabilities outputing by K classifiers. The training performance of each base classifier, however, has not been taken into account. In fact, a classifier that outperforms others should be assigned a higher "confidence", which represents our belief in the goodness of that classifier. The "confidence" of a classifier can be defined as the training accuracy of the classifier. By this mean, we can build a weighted ensemble rule in which the weight represents our confidence on a specific classifier. For the choice of weights, we choose a novel performance metric AUCarea, which is introduced in the next section. The weighted ensemble rule is defined as Table 2. The new weighted ensemble rules are simply accomplished by multiply p_{ij} by $AUCarea_i$, $AUCarea_i$ represents the value of AUCarea of base classifier *i*.

3.4. The way of building AMCS

Conclude above, our MCS model can be simply described as Fig. 3. For an imbalanced data, we first conduct feature selection to extract the best feature set. Then, the data is split into K sub-training bins by sampling method and each bin is for the use of training a base classifier. After training, we compute the AUCarea of each base classifier rely on its performance on training data. To classify an unseen data *X*, we obtain the probabilistic output P(c|X) of each base classifier, and use the weighted ensemble rule described in Table 2 to fuse the outputs of all base classifiers.

As we argued previously, an adaptive MCS should choose the best ensemble framework, base classifier, feature selection method and ensemble rule according to the characteristic of the imbalanced data. For this purpose, we divide the imbalanced data into 8 types based on their IR, dimension, and the number of classes, as is shown in Table 3. We expect to form specific AMCS for different type of imbalanced data, the specific AMCS can be implemented by following one of the route in the network described in Fig. 4. Each route in Fig. 4 represents the choice of ensemble framework, the choice of feature selection method, the choice of base classifier and the choice of ensemble rule. In this study we select five base classifiers as options, which are, C4.5, SVM, RBF-NN, DGC [68] and KNN. Note that Adaboost.M1 framework has already employed majority vote as the ensemble rule, we did not consider other choices of ensemble rules of Adaboost.M1.

Table 3
Description of 8 types of imbalanced data.

Type number	Description*
Type 1	High IR, High dimension, Large number of classes
Type 2	High IR, Low dimension, Large number of classes
Type 3	High IR, High dimension, Small number of classes
Type 4	High IR, Low dimension, Small number of classes
Type 5	Low IR, High dimension, Large number of classes
Type 6	Low IR, Low dimension, Large number of classes
Type 7	Low IR, High dimension, Small number of classes
Type 8	Low IR, Low dimension, Small number of classes

*High IR refers to a dataset whose IR is not less than10(*threshold1*).
*High dimension refers to a dataset whose number of features is not less than 10(*threshold2*).
*Large number of classes refers to a dataset whose number of classes is not less than 6(*threshold3*).
*Since there are no standard criteria of deciding the value of *threshold1, threshold2, threshold3* proposed in previous literature, the thresholds used in this paper are obtained by our preliminary study.

Table 4
Confusion matrix of binary classification problem.

	Predicted positive class	Predicted negative class
Positive class	TP	FN
Negative class	FP	TN

The best combination of types and routes will be obtained through empirical study that we carry out in Section 5.

4. Evaluation in imbalanced data

4.1. Evaluation metric

The evaluation measure is a key factor for both assessing the classification performance and guiding the learning progress of classifier [9]. Accuracy is the most commonly used evaluation metric. However, for imbalanced data classification problems, accuracy may not be a good choice because accuracy often has a bias toward majority class [27,28]. Performance metrics adapted into imbalanced data problems, such as Receiver Operating Characteristics (ROC) [30], G-Mean(GM), and F-measure(F_m) [9], are less likely to suffer from imbalanced distributions because they measure the classification performance of each class independently. In a binary classification problem, instances can be labeled as positive or negative. For binary imbalanced datasets, the minority class is usually considered as positive while the majority class is considered as negative. The confusion matrix as is shown in Table 4 records the results of correctly and incorrectly recognized situations of each class. ROC, G-mean and F-measure can all be

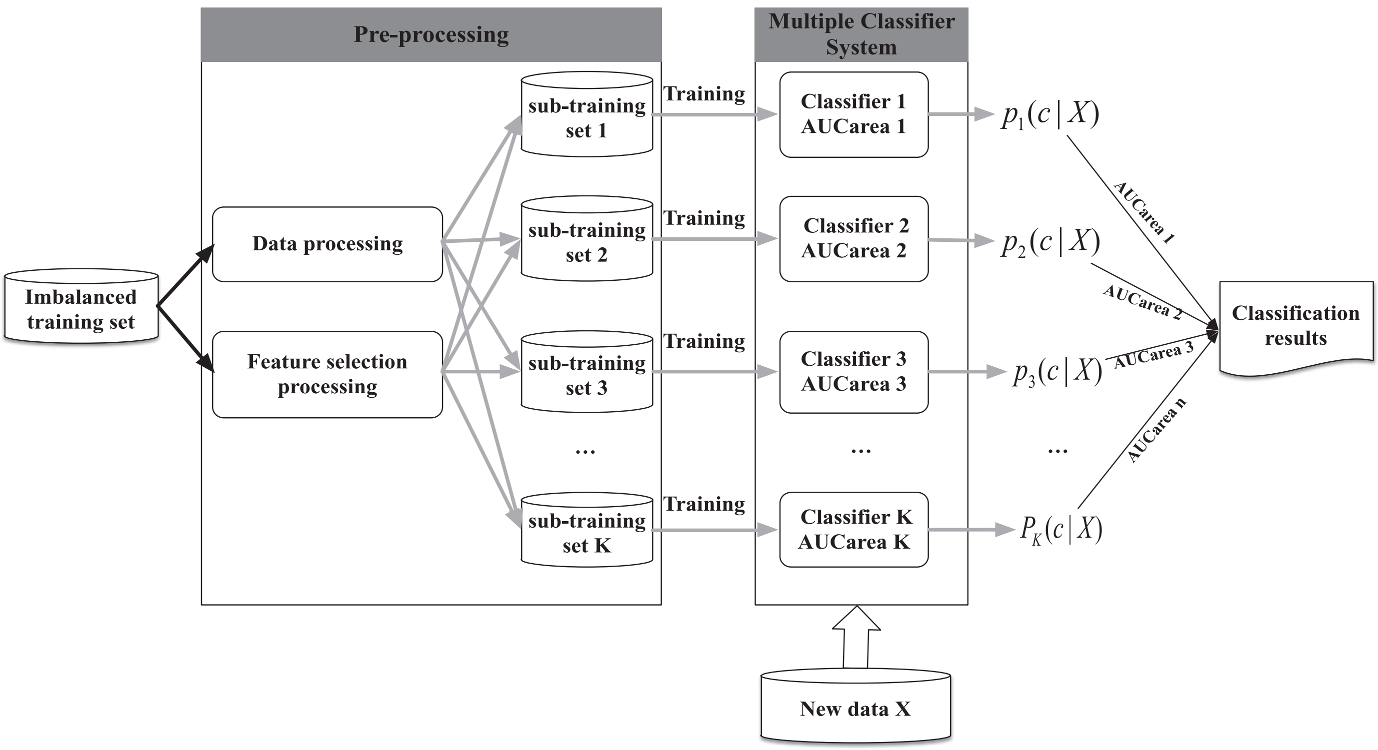

Fig 3. Architecture of AMCS.

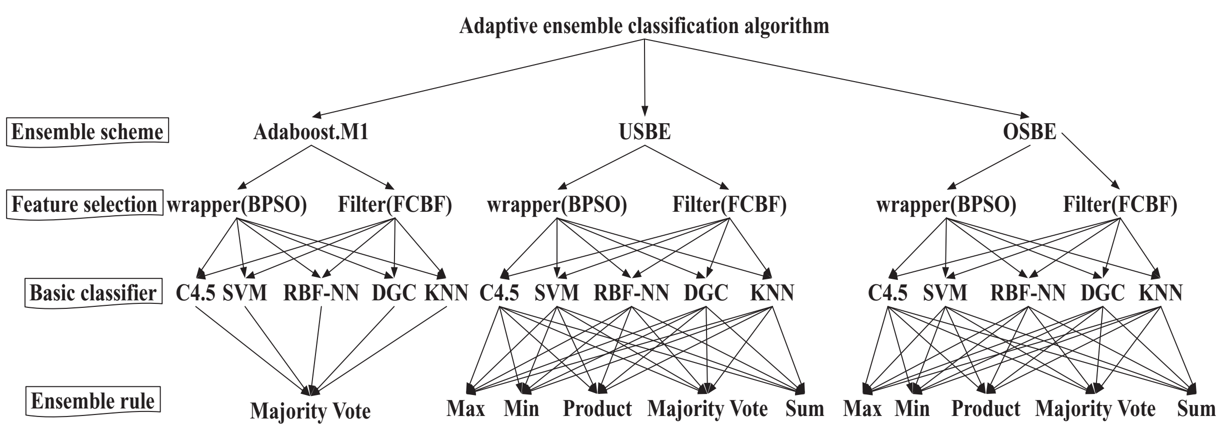

Fig 4. Adapting path of AMCS.

obtained from Table 4:

$$GM = \sqrt{\frac{TP}{TP+FN} \cdot \frac{TN}{FP+TN}} \quad (1)$$

$$F_m = \frac{(1+\beta^2)\left(\frac{TP}{TP+FP} \cdot TPR\right)}{\beta^2 \frac{TP}{TP+FP} + TPR} = \frac{(1+\beta^2)\left(\frac{TP}{TP+FP} \cdot \frac{TP}{TP+FN}\right)}{\beta^2 \frac{TP}{TP+FP} + \frac{TP}{TP+FN}} \quad (2)$$

ROC (Receiver Operating Characteristic) curve is recognized as the most rational choice for imbalanced data, which depicts relative trade-offs between the benefits and costs [29].ROC curve plots $\frac{FP}{TN+FP}$ (FPR) on the X-axis and plots $\frac{TP}{TP+FN}$ (TPR) on the Y-axis, where different pairs of (FPR, TPR)s can be obtained by changing the thresholds of the classifier [30]. The threshold of a classifier presents the degree to which an instance is a member of a class [29]. In practice, we often use the Area Under ROC Curve (AUC) as a scalar measure instead of ROC curve, the larger AUC value is the better. As we can see, AUC, F-measure and G-Mean are only able to be applied to binary classification problems.

There are three ways to extend binary metrics into multi-class case, OVO approach [58], OVA [59] approach and directly extension [60,61]. OVO and OVA approaches aim to measure the overall performance by evaluating the performance of each class. Specifically, OVO approaches consider the classification performances between all pairs of classes while OVA approaches measure the classification performance between a single class against the rest of all classes. More directly extensions of AUC have been studied in [60–62] based on Volume Under the ROC Surface(VUS). Though VUS represents a theoretical justification of the classifiers, it is hard to visualize and compute. For C-classes cases, each point of ROC surface lies in an $C \cdot (C-1)$ space, and the complexity of computing an n-points convex hull is $O(n^D)$ [61]. Researches in [60,61] provide some methods of estimating VUS through the maximum and minimum of VUS, in which a cost matrix is needed. The advantages of OVO and OVA approaches lie in the natural intuition and easy computation. In particular, through OVO approach we can clarify the mutual classification mistake between pair of classes, which is essential for applications

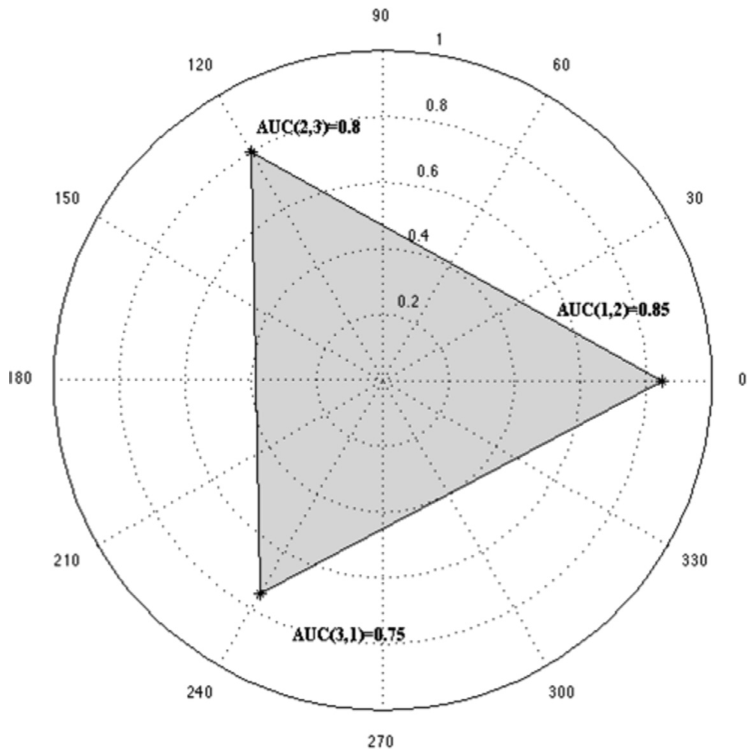

Fig. 5. Polar diagram of three-classes case ($AUC_{1,2}$=0.85, $AUC_{2,3}$=0.8, $AUC_{1,3}$=0.75).

like fault detection. In this study, we utilize a novel extension of AUC by OVO approach proposed in [31] and actualize it as the evaluation criterion.

For a dataset that contains C classes, we calculate AUC values for each pair of classes respectively. Finally we can get C_C^2 AUC values. In order to obtain a scalar metric, we need a fusion strategy to integrate all the AUC values as the final metric. A natural fusion strategy is to output the average value of all the AUC values, but this approach has a defect. For different classifiers, each single AUC value may be different, but the average value may remain the same, thus we cannot estimate which hypothesis is better. In this paper, we choose a novel method of fusing AUC values proposed in [31], the idea is as follows: all the AUC values are plotted in a polar coordinate, we calculate the area covered by the polar diagram as the final evaluation metric, the larger the area is the better. This measure is named as AUCarea. An AUC polar diagram for a three-classes problem is shown in Fig. 5.

The way to compute AUCarea is described as Eq. (3) (the detail analysis can be found in [31]):

$$\mathrm{AUCarea} = \frac{1}{2}\sin\left(\frac{2\pi}{q}\right)\left(\left(\sum_{i=1}^{q-1} r_i \times r_{i+1}\right) + (r_q \times r_1)\right) \tag{3}$$

Here q $= C_C^2$(C is the number of classes), r_i is the ith AUC value.

Considering most of metrics are ranged within [0,1], we normalized AUCarea by $\frac{\mathrm{AUCarea}}{\mathrm{Maximum\_AUCarea}}$, where Maximum_AUCarea is calculated by Eq. (3) when setting all r_is to 1. The normalized AUCarea can be obtained by Eq. (4). In the following paper, we will use normalized AUCareainstead of AUCarea proposed in [31] (but we simply call normalized AUCarea as AUCarea).

$$\begin{aligned}\mathrm{AUCarea} &= \frac{\frac{1}{2}\sin\left(\frac{2\pi}{q}\right)\left(\left(\sum_{i=1}^{q-1} r_i \times r_{i+1}\right) + (r_q \times r_1)\right)}{\frac{1}{2}\sin\left(\frac{2\pi}{q}\right)\cdot q} \\ &= \frac{\left(\sum_{i=1}^{q-1} r_i \times r_{i+1}\right) + (r_q \times r_1)}{q}\end{aligned} \tag{4}$$

Another reason we believe AUCarea is better than average AUC is that AUCarea is more sensitive when mutual mistakes of a pair of classes increase. In other words, it turns out that if there exists awful AUC, the value of AUCarea will decrease sharply. We compare average AUC with AUCarea to illustrate this issue. The average AUC(AVG_AUC) can be computed through Eq. (5).

$$\mathrm{AVG\_AUC} = \frac{\sum_{i=1}^{q} r_i}{q} \tag{5}$$

Assuming that r_i is changed to $r_i - l$, then we can compute the relatively change of AVG_AUC and AUCarea: $\Delta\mathrm{AVG\_AUC} = \frac{l}{q}$ and $\Delta\mathrm{AUCarea} = \frac{l\times(r_{i-1}+r_{i+1})}{q}$. Since single AUC values are larger than 0.5, ΔAUCarea would slightly larger than ΔAVG_AUC. This suggests that if any single AUC decreased, AUCarea would decrease more than AVG_AUC. The above analysis illustrates that AUCarea is more sensitive when any pairs of classes gain poor AUC value. For imbalanced data, the AUCs between minority class and some other classes are often poor. For this reason, AUCarea is superior to AVG_AUC.

4.2. Posterior probability estimate

The five ensemble rules introduced in Table 2 require a posterior probability of base classifiers for each unseen sample. Since ROC curve also needs the learning models return posterior probability for the prediction [29], we estimate the posterior probability both for base classifiers and AMCS.

Consider a dataset S that contains D features, N samples and mclasses, $S = \{s_1, s_2, \ldots, s_N\}$ where $s_1 = \{x_1, x_2, \ldots, x_D\}$, the class labels $C = \{c_1, c_2, \ldots, c_m\}$.We denote the posterior probability for s_i belonging to c_j as $P_{base}(c_j|s_i)$ and $P_{en}(c_j|s_i)$ for base classifiers and ensemble classifiers respectively. Since SVM, C4.5, RBF-NN and DGC all make their prediction based on posterior probability, there is no need to estimate $P_{base}(c_j|s_i)$ for them specifically (For DGC we use the gravity as posterior probability). Here we only discuss the $P_{base}(c_j|s_i)$ for KNN. Let n_{ij} represents the number of s_i's neighbors that

Table 5
Detailed information of test datasets.

Type	Dataset	Short name	# Samples	# Classes	Class distribution	IR	# Features
Type 1	Zoo	–	101	7	(41/20/5/13/4/8/9)	1:10	16
	Soybean	–	307	18	(10/40/20/6/4)	1:10	35
	Autos	–	159	6	(2/14/33/32/20/9)	1:16	25
	Wine-Quality-Red	WineQR	1599	11	(3/16/205/192/60/5)	1:68	11
Type 2	Yeast	-	1484	10	(324/3/24/…/170/300/14/21)	1:23	8
	Shuttle	–	2175	7	(1194/2/4/236/86)	1:853	9
	Abalone	Aba.	4177	29	(15/57/115/391/689/…/6/9/2)	1:345	8
Type 3	Page-Blocks	PageB	5472	5	(4913/329/87/115/28)	1:175	10
	Lymphography	Lym.	148	4	(2/42/56/2)	1:21	18
	Thyroid	–	720	3	(11/25/466)	1:37	21
	Wine-Quality-White	WineQW	4898	5	(2198/1457/880/175/163)	1:14	11
Type 4	Car	–	1728	4	(1210/384/69/65)	1:18	6
	Ecoli*	–	358	4	(143/77/2/102)	1:72	7
	Kr-vs-k*	–	8425	4	(839/430/857/6299)	1:15	6
Type 5	Dermatology	Derm.	366	6	(77/42/49/33/14)	1:6	34
	Landsat	–	2000	6	(461/224/397/211/237/410)	1:2	36
	Penbased	Pen.	1100	10	(80/79/41/73/80/73/74)	1:2	16
Type 6	Led7digit	Led	500	7	(14/12/16/…/64)	1:5	7
	Glass	–	214	6	(70/76/17/13/9/29)	1:8	9
	Mf-mor*	–	2000	6	(200/200/../1000)	1:5	6
Type 7	Splice	–	3190	4	(767/768/1655)	1:2	60
	Mf-kar*	–	2000	4	(200/200/800/800)	1:5	64
	Wine	–	178	3	(41/49/33)	1:2	13
Type 8	New_thyroid	NewT	215	3	(150/35/30)	1:5	4
	Contraceptive	Contr.	1473	3	(440/233/357)	1:2	9
	Hayes-Roth	HayesR	160	3	(65/64/31)	1:2	5
	Balance	–	625	3	(49/288/288)	1:6	4

* The original Mf-mor contained 10 classes and each class has 200 samples, we merged the rest 5 classes as a majority class in our study. The original Ecoli has 8 classes, we merged the rest 3 classes. Mf-kar has the same structure as Mf-mor, we merged 8 classes into 2 large classes, 800 samples for each. The original Kr-vs-k contains 17 classes, we merged the 4–17 classes in order to construct an imbalanced dataset

belongs to jth class,$\sum_{j=1}^{C} n_{ij} = k$, then $P_{base}(c_j|s_i)$ of KNN can be computed by $P_{base}(c_j|s_i) = n_{ij}/k$.

Assuming the number of base classifiers of ensemble classifier is T, for each base classifier, the AUCarea and $P_{base}(c_j|s_i)$ are computed. The corresponding $P_{en}(c_j|s_i)$ can be defined as the mean of weighted probability according to AUCarea and $P_{base}(c_j|s_i)$, that is,

$$P_{en}\left(c_j|s_i\right) = \sum_{t=1}^{T}\left(\frac{AUCarea_t}{\sum_{t=1}^{T} AUCarea_t} \times P_{base_t}\left(c_j|s_i\right)\right) \tag{6}$$

where $\frac{AUCarea_t}{\sum_{t=1}^{T} AUCarea_t}$ is the weight of tth base classifier.

5. Empirical analyses

The current section includes two phases. In the first phase we aim to find the dominating routes of Fig. 4 to frame the AMCS for 8 types of imbalanced datasets. In the second phase we apply AMCS to untested benchmarks and compare it against other six state-of-the-art algorithms.

According to the previous aims, we divide this section into three parts: in Section 5.1 we introduce the experimental framework and imbalanced datasets used in our empirical study. Secondly, experiments carried out in order to find the dominating routes to form AMCS for different types of imbalanced datasets are presented in Section 5.2. After that, comparisons among AMCS and the state-of-the-arts are conducted in Section 5.3.

5.1. Experimental setup and datasets

In this study we utilize 27 multi-class imbalanced datasets from KEEL datasets repository for multiple class imbalanced problems at http://www.keel.es/datasets.php and UCI database [35] to test the performance of AMCS. The detail information about these datasets is summarized in Table 5, where IR is the sample ratio between the most affluent class and the rarest class. In Table 5 we also order all the datasets based on the types they belong to. Besides, the second column corresponding to the short name for some datasets, we will use these short names in the rest of this paper.

All experimental studies were conducted by employing 5-fold cross validation, where each fold follows the same distribution as the original dataset. The relevant parameters in BPSO are the same as the corresponding parameters set in [34], where inertia weight is set to 0.729, learning factor c_1, c_2 are both set to 1.49445. Parameters of FCBF are referenced [66]. The number of iterations of the three ensemble models is 20. For the base classifiers, we set k of KNN equals to 5 for those datasets that the rarest class contains at least 5 samples, otherwise, k is set to be equal with the number of samples of the rarest class. Besides, we employ MATLAB RBF-NN tool box and libsvm for RBF-NN and SVM.

5.2. Analyzing the dominating components to form AMCS

To find the dominating feature selection algorithm, ensemble framework, base classifier and ensemble rule to frame AMCS for 8 types of imbalanced datasets, we select 16 datasets from Table 5, for each type 2 datasets are selected. We test all the 110 routes in Fig. 4 regarding to AUCarea, overall accuracy, accuracy refer to the most flourish class(Maj. acc), accuracy refer to the rarest class(Min. acc), average G-mean and average F-measure(AVG-Fm) computed through OVO(AVG-GM). The main criterion of evaluating the best MCS as the AMCS is AUCarea. Table 6 lists the performances of algorithms using DGC and C4.5 as base classifiers[1], where the best ensemble rules are pointed out in the parentheses bellowing to the value of AUCarea. The outperforming algorithms are highlighted in bold, from which we can

[1] For the limit of space we did not list all the experimental results for exploring best routes of AMCS, to view the entire results, please go to https://github.com/liyijing024/AMCS.

Table 6
Performances of DGC and C4.5 based algorithms regarding to AUCarea.

Type	Dataset	BUD*	FUD	BOD	FOD	BAD	FAD	BUC	FUC	BOC	FOC	BAC	FAC
Type 1	Soybean	0.9512 (5)*	0.9004 (1)	0.9401 (2,4,5)	0.9045 (1)	**0.9534 (4)**	0.9391 (4)	0.9174 (4,5)	0.9253 (4,5)	0.9365 (4,5)	0.9365 (4,5)	**1.0000 (4)**	**1.0000 (4)**
	Zoo	0.9345 (all)	0.8542 (1)	0.9345 (all)	0.8316 (all)	0.9372 (4)	**0.9524 (4)**	0.4601 (4,5)	0.5853 (1)	0.9492 (4,5)	0.9492 (4,5)	**1.0000 (4)**	**1.0000 (4)**
Type 2	Yeast	0.5988 (5)	0.5874 (1)	0.5585 (2)	0.6354 (2)	**0.6528 (4)**	0.6495 (4)	0.598 (2,3)	0.5980 (2,3)	0.2776 (2,3)	0.2788 (2,3)	**0.6916 (4)**	0.6708 (4)
	Ada	0.4408 (4)	**0.6601 (3,5)**	0.3993 (2)	0.3197 (2)	0.3789 (4)	0.2896 (4)	0.4413 (4,5)	0.4232 (4,5)	0.3732 (4,5)	0.2587 (4,5)	**0.6022 (4)**	0.4905 (4)
Type 3	Thyroid	0.8431 (1)	0.7478 (1)	0.8908 (2)	0.8911 (2)	**0.9011 (4)**	0.8204 (4)	0.8952 (4,5)	0.9472 (4,5)	0.8927 (1,4,5)	0.9411 (1,4,5)	0.9967 (4)	**1.0000 (4)**
	lym	0.7952 (1,4)	0.7077 (3,5)	0.8012 (1)	0.8735 (1)	**0.9735 (4)**	0.9060 (4)	0.8426 (4,5)	0.9734 (4,5)	0.8802 (4,5)	0.7875 (1)	0.6983 (4)	**1.0000 (4)**
Type 4	Ecoli	0.7469 (1)	0.6734 (1)	0.8272 (1)	0.8491 (2)	0.8067 (4)	**0.8757 (4)**	0.9068 (4,5)	0.9068 (4,5)	0.8361 (4,5)	0.8671 (1)	0.8334 (4)	0.8751 (4)
	Car	0.6709 (1)	0.7138 (1)	0.7359 (1)	0.7539 (1)	0.7803 (4)	**0.8311 (4)**	0.6572 (2,3)	0.7856 (4,5)	0.9224 (2,3)	0.8075 (2,3)	0.7700 (4)	**1.0000 (4)**
Type 5	Landsat	0.7377 (1)	0.8000 (2)	0.7279 (4)	0.8919 (4)	0.8706 (4)	**0.8952 (4)**	0.5149 (4,5)	0.9256 (2,3)	0.8284 (4,5)	**0.9256 (4,5)**	**0.9256 (4)**	0.5396 (4)
	Pen	0.8814 (1,4)	0.5291 (1)	0.8827 (1)	0.8756 (1)	0.9041 (4)	**0.9245 (4)**	0.8825 (4,5)	0.3682 (4,5)	0.8793 (4,5)	0.5855 (4,5)	**0.9807 (4)**	0.5983 (4)
Type 6	Mf-mor	0.6737 (1)	0.2726 (1,5)	0.6618 (2)	0.6639 (2)	**0.9086 (4)**	0.7197 (4)	0.8704 (4,5)	0.4596 (4,5)	0.7498 (4,5)	0.7308 (4,5)	**0.9931 (4)**	0.9398 (4)
	Led	0.6944 (1,3,5)	0.7011 (1,3,5)	0.8810 (all)	0.7251 (all)	0.7885 (4)	**0.7948 (4)**	0.6681 (1,4,5)	0.6481 (4,5)	0.6481 (4,5)	0.6481 (4,5)	0.8310 (4)	**0.8646 (4)**
Type 7	Wine	0.9087 (5)	0.2583 (all)	0.9264 (1)	0.7572 (4)	**0.9699 (4)**	0.8952 (4)	0.9699 (4,5)	0.7088 (4,5)	**1.0000 (4,5)**	0.7310 (4,5)	**1.0000 (4)**	0.8815 (4)
	Splice	0.2818 (1,3,5)	0.2584 (2)	**0.6929 (1)**	0.2817 (4)	0.2292 (4)	0.3027 (4)	0.5054 (4,5)	0.5054 (4,5)	0.3376 (4,5)	0.3376 (4,5)	**0.9552 (4)**	0.9418 (4)
Type 8	NewT	0.9153 (all)	0.9887 (2)	0.9543 (all)	0.9167 (1,4,5)	**1.0000 (4)**	0.9167 (4)	**1.0000 (4,5)**	0.8815 (1,2,3)	0.9251 (4,5)	0.8968 (1,2,3)	**1.0000 (4)**	**1.0000 (4)**
	HayesR	0.3577 (4)	0.4987 (5)	0.3793 (4)	0.3682 (5)	0.6682 (4)	0.6682 (4)	0.7539 (4,5)	0.5615 (2,3)	0.7543 (4,5)	0.5476 (4,5)	**0.8760 (4)**	0.8314 (4)

* "B" :BPSO, "F":FCBF, "U":USBE, "O":OSBE, "A":Adaboost, "D":DGC, "C":C4.5.So, BUD presents a MCS that employs BPSO as feature selection algorithm, USBE as the ensemble framework and DGC as base classifier. So do other MCS like BOD, BAC, etc.

* The numbers in parentheses represent the best ensemble rule when conducing the corresponding algorithm, "1" denotes weighted max, "2" denotes weighted min, "3" denotes weighted product, "4" denotes weighted majority vote, "5" denotes weighted sum, and "all" means five ensemble rules obtain the same results. The five ensemble rule are listed in Table 2.

Table 7
Dominating adapting routes for different types of imbalanced data.

Type	Dataset	Best AUCarea	Best route
Type 1	Soybean	1.0000	BPSO/**FCBF-Adaboost-C4.5**
	Zoo	1.0000	BPSO/**FCBF-Adaboost-C4.5**
Type 2	Yeast	0.9554	**BPSO-Adaboost-KNN**
	Ada	0.6901	**BPSO-Adaboost-KNN**
Type 3	Thyroid	1.0000	**FCBF-Adaboost-C4.5**, BPSO-Adaboost-RBF
	lym	1.0000	**FCBF-Adaboost-C4.5**/KNN/SVM
Type 4	Ecoli	0.9361	**BPSO-Adaboost-KNN**
	Car	1.0000	**BPSO-Adaboost-KNN**/RBF,FCBF-Adaboost-C4.5/KNN/RBF
Type 5	Landsat	0.9291	FCBF-Adaboost-KNN
	Pen	0.9993	**BPSO-Adaboost-KNN**
Type 6	Mf-mor	1.0000	BPSO/**FCBF-Adaboost-SVM**
	Led7digit	0.8546	**FCBF-Adaboost-SVM**/C4.5
Type 7	Wine	1.0000	**BPSO-Adaboost-C4.5**/RBF/SVM, BPSO-OSBE-C4.5
	Splice	0.9552	**BPSO-Adaboost-C4.5**
Type 8	NewT	1.0000	**BPSO-Adaboost-SVM**/DGC/KNN/RBF/C4.5, FCBF-Adaboost-C4.5/SVM, BPSO-USBE-C4.5
	HayesR	1.0000	**BPSO-Adaboost-SVM**

Table 8
Friedman test for Type 5.

	Metric	Landsat	Pen	Rank
Adaboost-KNN	Acc	0.9252	0.9940	1.75
	Maj_acc	1.0000	1.0000	
	Min_acc	0.9667	1.0000	
	AUCarea	0.8969	0.9943	
	GM	0.9487	0.9967	
	FV	0.9320	0.9942	
BPSO-Adaboost-KNN	Acc	0.9327	0.9970	**2.42**
	Maj_acc	1.0000	1.0000	
	Min_acc	0.9556	1.0000	
	AUCarea	0.9102	0.9993	
	GM	0.9535	0.9983	
	FV	0.9386	0.9970	
FCBF-Adaboost-KNN	Acc	0.9633	0.9664	1.83
	Maj_acc	1.0000	1.0000	
	Min_acc	0.9081	1.0000	
	AUCarea	0.9291	0.9698	
	GM	0.9510	0.9373	
	FV	0.9495	0.9838	

observe that dominating algorithms vary in types. For most types of datasets, the algorithms employing Adaboost outperform those algorithms where USBE and OSBE are involved, while BPSO based algorithms are better than FCBF based algorithms in a general perspective.

We summarized all the validation results of all 110 routes and picked out the best routes for all datasets, as is shown in Table 7. It shows that for most types a consistent dominating algorithm can be found apart from *Type 5*. We consider the unanimous algorithms as the initially adapting criteria of AMCS. Though there is no consistent adapting route for *Type 5*, it is clear that Adaboost and KNN are the best ensemble framework and base classifier, while the choice of feature selection algorithm is ambiguous. Therefore, we recall Friedman test [42] for BPSO-Adaboost-KNN and FCBF-Adaboost-KNN and choose Adaboost-KNN as the baseline, the results are listed in Table 8. Since BPSO significantly outperforms FCBF, BPSO-Adaboost-KNN is selected as the initial adapting route for *Type* 5. For those types that not only one best route is found, we compare their running time and choose the less time-consuming algorithm as the initial adapting algorithm. The initially adapting route for AMCS are highlighted in bold in Table 7.

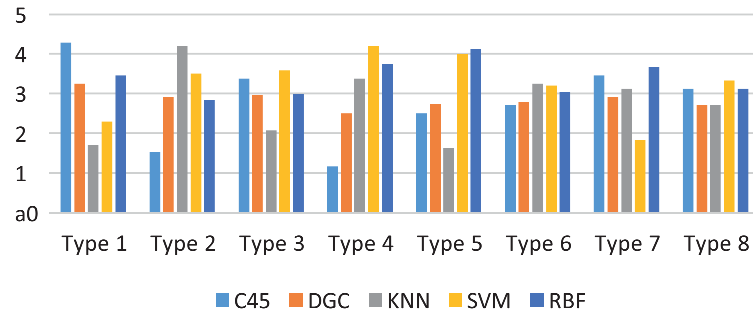

Fig. 6. Friedman test of base classifiers, where the horizontal axis is the type number of imbalanced dataset, the vertical axis represents the rank.

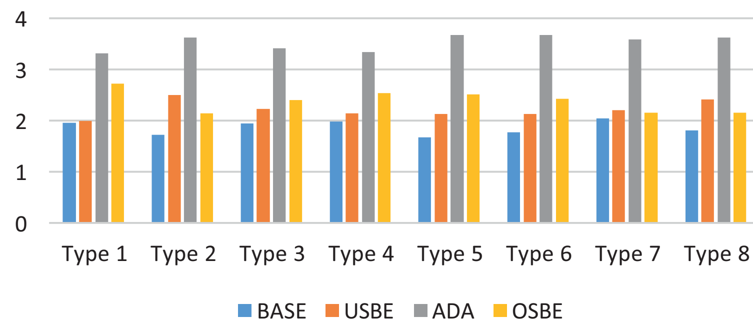

Fig. 7. Friedman test of ensemble frameworks, the corresponding base classifiers is treating as baselines, denoted as BASE in the figure. The horizontal axis is the type number of imbalanced dataset, the vertical axis represents the rank.

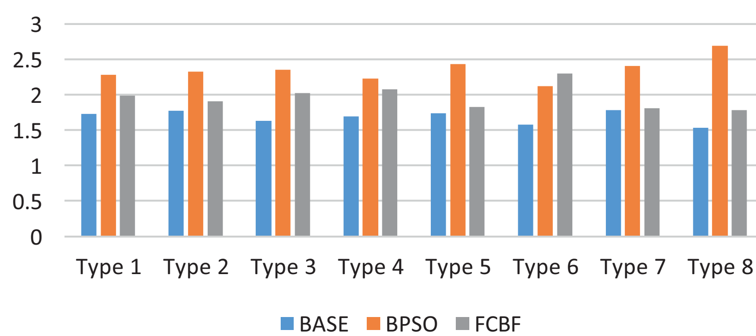

Fig. 8. Friedman test of feature selection algorithms, the corresponding base classifiers is treating as baselines, denoted as BASE in the figure. The horizontal axis is the type number of imbalanced dataset, the vertical axis represents the rank.

To validate the properness of the results presented in Table 7, we recall Friedman test to compare the overall performance of each component of AMCS, as is shown in Figs. 6–8, the better algorithm gains higher rank. In our Friedman test, six metrics mentioned previously are all considered. Fig. 6 shows the average ranks of five base classifiers. The highest ranking base classifiers for *Type 1, 2* and *8* are the same as the corresponding dominating base classifiers in Table 7. For *Type 3,4,5,6*, initial AMCSs do not select the highest ranking base classifiers, but still the superiority base classifiers since the selected classifiers earn a fairly ranking. However, for *Type 7* the lowest ranking base classifier SVM are selected by initial AMCS. Fig. 7 shows significantly that Adaboost ensemble framework outperforms USBE and OSBE for all types of imbalanced data, which is consistent with the results shown in Table 7. Fig. 8 shows the ranks of two feature selection algorithms for all types of imbalanced data. For *Type 3* and *Type 4*, the dominating ranking algorithms are discrepant with the selected feature selection algorithms of AMCS.

Since the initially selected algorithms of AMCS for *Type 3–7* are different from the highest ranking algorithms, for the sake of justice, we executed Wilcoxon paired signed-rank test to do pairwise comparisons between the algorithm selected by AMCS and the corresponding highest ranking algorithm for *Type 3–7* [41]. The results are presented in Table 9, the outperforming adapting routes are listed in the last column, where the routes that are different from the initial AMCS are in bold. Therefore, we modify the initial AMCS based on the Wilcoxon tests.

To summarize, the adapting routes of AMCS are as follows (For Adaboost only majority vote is considered as the ensemble rule, so we do not list the ensemble rule if Adaboost is employed):

- For *High IR, High dimension, Large number of classes* datasets, the adapting route of AMCS is: FCBF as feature selection algorithm, Adaboost as the ensemble framework, C4.5 as the base classifier.
- For *High IR, Low dimension, Large number of classes* datasets, the adapting route of AMCS is:BPSO as feature selection algorithm, Adaboost as the ensemble framework, KNN as the base classifier.
- For *High IR, High dimension, Small number of classes* datasets, the adapting route of AMCS is: FCBF as feature selection algorithm, Adaboost as the ensemble framework, C4.5 as the base classifier.
- For *High IR, Low dimension, Small number of classes* datasets, the adapting route of AMCS is: BPSO as feature selection algorithm, Adaboost as the ensemble framework, SVM as the base classifier.

Table 9
Wilcoxon tests for Type 3–7.

Type	Comparison	R^+	R^-	Hypothesis ($\alpha = 0.05$)	|Z|	p-value	Selected
Type 3	FCBF-Adaboost-C4.5 vs. BPSO-Adaboost-SVM	10.00	0.00	FCBF-Adaboost-C4.5 < BPSO-Adaboost-SVM	1.826	0.068	FCBF-Adaboost-C4.5
Type 4	FCBF-Adaboost-KNN vs. BPSO-Adaboost-SVM	15.00	21.00	FCBF-Adaboost-KNN < BPSO-Adaboost-SVM	0.421	0.674	**BPSO-Adaboost-SVM**
Type 5	BPSO-Adaboost-KNN vs. BPSO-Adaboost-RBF	16.00	50.00	BPSO-Adaboost-KNN < BPSO-Adaboost-RBF	1.511	0.131	**BPSO-Adaboost-RBF**
Type 6	FCBF-Adaboost-SVM vs. FCBF-Adaboost-KNN	65.00	1.00	FCBF-Adaboost-SVM < FCBF-Adaboost-KNN	2.845	0.004	FCBF-Adaboost-SVM
Type 7	BPSO-Adaboost-C4.5 vs. BPSO-Adaboost-RBF	21.00	3.50	BPSO-Adaboost-C4.5 < BPSO-Adaboost-RBF	2.201	0.028	BPSO-Adaboost-C4.5

- For *Low IR, High dimension, Large number of classes* datasets, the adapting route of AMCS is: BPSO as feature selection algorithm, Adaboost as the ensemble framework, RBF as the base classifier.
- For *Low IR, Low dimension, Large number of classes* datasets, the adapting route of AMCS is:FCBF as feature selection algorithm, Adaboost as the ensemble framework, SVM as the base classifier.
- For *Low IR, High dimension, Small number of classes* datasets, the adapting route of AMCS is:BPSO as feature selection algorithm, Adaboost as the ensemble framework, C4.5 as the base classifier.
- For *Low IR, Low dimension, Small number of classes*, the adapting route of AMCS is: BPSO as feature selection algorithm, Adaboost as the ensemble framework, SVM as the base classifier.

When testing the adapting rule for AMCS, we also find some comprehensive findings for imbalanced learning. Firstly, for all types of imbalanced datasets, AMCS always choose Adaboost as the ensemble framework, which implies over-sampling and under-sampling may not always be a good choice. Secondly, FCBF is chosen for all the three types of datasets that are high dimensional datasets. So as a suggestion, filter feature selection method may be better than wrapper method for high dimensional datasets. Lastly, as a guideline of choosing base classifier, C4.5 and KNN are better than other three classifiers when classifying high IR datasets and SVM outperforms other base classifiers for low dimensional datasets.

5.3. Validate AMCS and compare it with state-of-the-arts

To validate the efficiency of AMCS[2], we test AMCS on the rest datasets which are not selected in Section 5.2 and compare AMCS with some state-of-the-arts. we choose six efficient algorithms proposed to cope with imbalanced learning, i.e., SMOTEBoost [16], EasyEnsemble [25], PUSBE [47], a cost-sensitive decision tree ensemble algorithm [18], a novel ensemble algorithm that using K-means to split majority samples(ClusterBal) [15] and Imbalanced DGC(IDGC) [14].

SMOTEBoost and EasyEnsemble are two popular ensemble algorithms that have been used for comparisons in many previous studies [1,4,15,18,47]. PUSBE is a novel ensemble algorithm that a filter method for feature selection is carried out. It is interesting to take PUSBE into comparison since it also considered feature selection in the ensemble model. The proposed ensemble algorithm in [18] is based on a cost-sensitive basic classifier and uses stochastic evolutionary algorithm to fusion basic classifiers. ClusterBal uses clustering method to split majority samples, converts an imbalanced dataset into multiple balanced sub-datasets and then trains multiple base classifiers. Finally, the classification results of all the base classifiers are combined by a specific ensemble rule called MaxDistance. IDGC does not employ any ensemble strategy but earns good results as is suggested in [14]. All of these algorithms select different base classifiers, such as SVM, Ripper, CART, DGC, etc(we choose KNN as basic classifier in ClusterBal). Note that these algorithms except for IDGC have only been used in binary classification problems in the corresponding papers, we generalize these methods into multiple case by using OVO approach. The brief descriptions of the ensemble strategy for five ensemble algorithms are shown in Table 10, where CS-MCS stands for the cost-sensitive decision tree ensemble algorithm proposed in [18].

Table 11 presents the performances of ACMS and these state-of-the-arts in terms of AUCarea, average G-mean and average F-measure. The superior algorithm regarding to AUCarea, average G-mean and average F-measure are highlighted in bold in Table 11. The results show that all the seven algorithms are comparable, as all of them have obtained the best performance in different datasets with respect to the others. Overall, PUSBE and AMCS are slightly better than the other algorithms. Considering PUSBE is also feature selection based ensemble algorithm, this finding can illustrate the crucial role of feature selection for imbalanced learning. All the ensemble algorithms outperform IDGC, which implies ensemble algorithms are more efficient than straightforward algorithm. Another observation is that under-sample based methods like EasyEnsemble, CS-MCS and ClusterBal perform bad in benchmarks that are highly imbalanced, such as PageB and Shuttle. This may because under-sample methods abandon too much information of majority classes in order to balance the sub-datasets. Taking ClusterBal for example. In data balancing process, ClusterBal split majority samples into K clusters, where K is set as the ratio of majority class samples over the minority class samples. When dataset is extremely unbalance, say only one sample in the minority class of the training set, then ClusterBal will split the majority class into huge amount of clusters while each cluster contains only one or a few samples. In this way, all the base classifiers cannot be trained well because the information the sub-training-sets carrying are too scarce (although they may be balanced). Another reason some state-of-the-art algorithms like ClusterBal failed to achieve as ideal results in multi-class datasets as in binary case is that boundaries among multiple classes in each balanced sub-dataset become distorted since we just use a small cluster of multiple majority classes to generate balanced datasets.

We recall nonparametric statistical tests to clarify the performances of AMCS and state-of-the-arts. First, Friedman test is employed to detect the overall performances of all tested algorithms regarding to AUCarea, average G-mean, average F-measure and accuracy. After that, we chose post-hoc test to check out if the highest ranking algorithm in Friedman test is significantly better than the rest [43]. Friedman testing results are presented in Fig. 9, which supports AMCS gains the highest rank. Next we control AMCS and compare it with the others to see if there exist significant differences between AMCS and other six algorithms using post-hoc test, the results are shown in Table 12. The results indicate all the six algorithms have significant differences comparing with AMCS with low p-value, which implies AMCS significantly outperforms the others.

We also compare the runtime of all the algorithms used in this section, as is shown in Table 13. Since all the seven algorithms except IDGC are ensemble algorithms, the computational complexity of these six algorithms are relatively expensive. In general, ClusterBal wins in the most of benchmarks besides some large and highly imbalanced datasets. For those imbalanced datasets that AMCS

[2] The MATLAB source code of AMCS is available at https://github.com/liyijing024/AMCS

Table 10
Ensemble strategies of five state-of-the-art algorithms.

Algorithm	Ensemble Strategy
SMOTEBoost	SMOTE+Adaboost+Ripper (base classifier)
EasyEnsemble	Random under-sample +Bagging+Adaboost+CART (base classifier)
PUSBE	Filter feature selection+Boosting+SVM(base classifier)+Double-fault diversity based pruning method+Classifier fusion based on BP
CS-MCS	Random under-sample +Cost sensitive decision tree+Classifier fusion based on GA
ClusterBal	Split majority classes by Kmeans+KNN(base classifier)+MaxDistance ensemble rule

Table 11
Compare AMCS with state-of-the-arts regarding to AUCarea, average G-mean and average F-measure.

Dataset		Autos	WineQR	Shuttle	PageB	WineQW	Derm.	Glass	Mf-kar	Contr.	Balance	Kr-vs-k
	AUCarea	**0.9782**	0.8267	**1.0000**	0.8522	**0.8637**	**1.0000**	**0.9000**	0.9931	**0.8889**	**0.8622**	0.9196
AMCS	AVG_GM	**0.9887**	0.8339	**1.0000**	0.9445	**0.9223**	**1.0000**	**0.9473**	0.9962	**0.9388**	**0.9847**	0.9148
	AVG_Fm	0.9346	0.6696	**1.0000**	0.9469	0.7451	**1.0000**	0.9186	0.9914	**0.9310**	**0.9400**	0.9679
	AUCarea	0.7970	0.8370	0.9960	0.8280	0.7040	0.9730	0.7590	**0.9974**	0.5420	0.6782	0.8840
SMOTEBoost	AVG_GM	0.9530	0.7731	0.9620	0.9630	0.8610	0.9580	0.8550	0.9934	0.5700	0.9287	0.9434
	AVG_Fm	0.8820	**0.8760**	0.9940	0.9880	0.8140	0.9900	0.8830	0.9836	0.7360	0.9421	**0.9719**
	AUCarea	0.7270	0.7712	0.8250	0.8780	0.8140	0.9340	0.8520	0.9834	0.5500	0.7743	**0.9202**
EasyEnsemble	AVG_GM	0.9270	0.8217	0.9000	0.9720	0.8540	0.9720	0.8860	**0.9967**	0.5900	0.9462	**0.9687**
	AVG_Fm	0.8710	0.8524	0.9880	0.9970	0.8770	0.9940	0.8920	**0.9916**	0.7940	0.9456	0.9559
	AUCarea	0.8040	0.7168	0.9140	**0.8990**	0.7800	**1.0000**	0.8580	0.9249	0.5370	0.6289	0.8634
PUSBE	AVG_GM	0.7980	0.7674	0.8990	**0.9780**	0.8090	**1.0000**	0.8990	0.9643	0.6340	0.8530	0.8953
	AVG_Fm	**0.9430**	0.7333	0.9990	**0.9650**	0.8260	**1.0000**	**0.9840**	0.9225	0.6990	0.8943	0.9515
	AUCarea	0.8320	**0.8380**	0.9160	0.7900	0.7000	0.9700	0.8070	0.9060	0.4930	0.6512	0.8682
CS-MCS	AVG_GM	0.9150	0.8549	0.9810	0.8150	0.7990	0.9730	0.8650	0.9900	0.6150	0.7180	0.9083
	AVG_Fm	0.9690	0.8585	0.9610	0.8600	0.8450	0.9950	0.8980	0.9444	0.7510	0.8645	0.9029
	AUCarea	0.8290	0.8304	0.6000	0.4010	0.6000	0.8000	0.8330	0.9078	0.7490	0.6804	0.8119
ClusterBal	AVG_GM	0.8140	**0.8605**	0.6790	0.3100	0.6640	0.8450	0.8990	0.9658	0.8160	0.8070	0.8434
	AVG_Fm	0.8930	0.8663	0.8400	0.5790	**0.8800**	0.9070	0.9020	0.9379	0.7990	0.7908	0.8719
	AUCarea	0.6790	0.4555	0.9735	0.8899	0.4633	0.9718	0.8086	0.9438	0.5516	0.6310	0.8629
IDGC	AVG_GM	0.8348	0.5578	0.9876	0.9384	0.5561	0.9812	0.8534	0.9890	0.6603	0.7669	0.8898
	AVG_Fm	0.6580	0.2150	0.9833	0.8946	0.5592	0.9692	0.8748	0.9530	0.6448	0.6310	0.9296

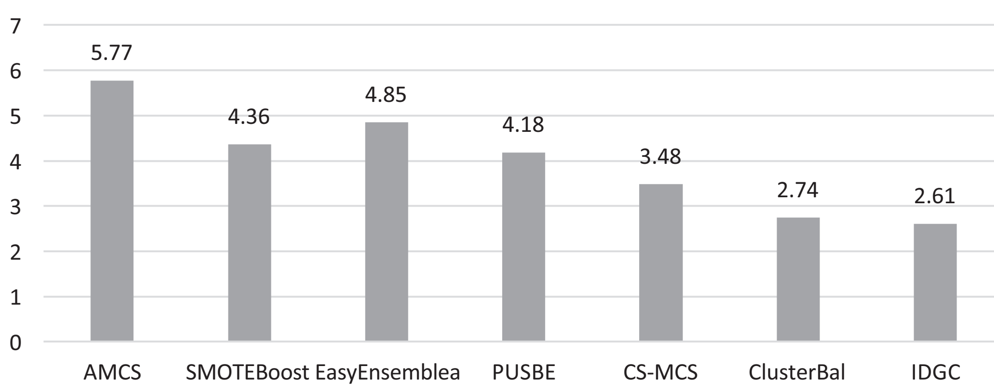

Fig. 9. Friedman test of AMCS and the state-of-the-arts.

employs filter feature selection algorithm, the computational time are lower than the others, while for those wrapper method is utilized, AMCS is more time-consuming. In particular, PUSBE, CS-MCS, IDGC employ evolutionary algorithms as a sub-component of the entire model, which lead to a higher computational cost with respect to SMOTEBoost and EasyEnsemble.

Table 12
Post hoc tests of the-state-of-the-arts (taking AMCS as the control method).

Algorithm(Rank)	Z	p-value	Hypothesis ($\alpha = 0.05$)
SMOTEBoost (4.36)	2.493	0.013	Rejected for AMCS
EasyEnsemble (4.85)	2.135	0.033	Rejected for AMCS
PUSBE (4.18)	3.034	0.002	Rejected for AMCS
CS-MCS (3.48)	3.618	0.003	Rejected for AMCS
ClusterBal (2.74)	4.315	0.000	Rejected for AMCS
IDGC (2.61)	4.324	0.000	Rejected for AMCS

6. The application of AMCS in Oil-bearing of reservoir recognition

In this section, we apply AMCS in oil-bearing of reservoir recognition. The descriptions of logging datasets are shown in Section 6.1, the experimental results and analyses are described in Section 6.2.

6.1. Data description

The oil-bearing of reservoir recognition data has 6 attributes, including: Acoustic travel time(AC), Compensated Neutron Logging(CNL), Pesistivity(RT), Porosity(POR), Oil Saturation(SO) and Permeability(PERM). AC is for use of analyzing the property that the sonic propagation varies when it comes to different rocks and fluids.

Table 13
Computational cost for each algorithm (runtime in terms of milliseconds).

Dataset	AMCS	SMOTEBoost	EasyEnsemble	PUSBE	CS-MCS	ClusterBal	IDGC
Autos	5305	5799	5152	6021	6910	**1340**	1645
WineQR	44040	28790	**6120**	39480	23200	27800	66960
Shuttle	13329	44192	11750	13494	18310	50434	**7740**
PageB	**7781**	129871	11075	26560	39250	183209	8820
WineQW	**4156**	74187	15386	24968	32029	69678	63960
Derm.	1731	2304	2100	3478	5078	**1524**	3720
Glass	1083	1821	1679	1604	1998	3320	**1468**
Mf-kar	25244	11347	106740	21300	**4572**	2366	73640
Contr.	6060	9571	**9140**	11194	23733	23434	9468
Balance	7496	9200	5042	9624	7430	**737**	3119
Kr-vs-k	133649	191965	188847	639957	197930	**107448**	146434

Table 14
Oilsk81 well logging explanation results.

Reservoir number	AC (μs/m)	CNL (%)	RT (Ω m)	POR (%)	SO (%)	PERM (mμm^2)	Conclusion
1	195	7.5	13.0	6.0	0	0	Dry layer
2	225	10.0	7.3	11.0	0	0	Water layer
3	230	14.0	5.5	12.0	0	0	Water layer
4	220	9.0	25.0	9.0	56	1.3	Oil layer
5	225	8.0	30.0	9.0	58	2.3	Oil layer
6	210	7.0	26.0	6.0	0	0	Dry layer
…	…	…	…	…	…	…	…
30	201	6.0	16.0	7.0	40	0.4	Inferior oil layer
31	213	9.5	12.0	9.0	61	2	Oil layer

Table 15
The distribution of each well logging.

Well number	The distribution of each sample (dry layer, water layer, oil layer, inferior oil layer)	IR
Oilsk81	(14, 2, 12, 3)	1:7
Oilsk82	(28, 2, 7, 11)	1:14
Oilsk83	(28, 3, 12, 7)	1:9
Oilsk84	(32, 6, 5, 9)	1:6
Oilsk85	(44, 4, 6, 11)	1:11

Generally, AC would increase dramatically if there were oil vapor in the void. Various effects of interaction between CNL and other substances can be used to study rock formation properties of the cross section. RT is a main parameter to judge fluids properties of reservoir. POR is defined as the ratio between the void space in a rock and the bulk volume of that rock. SO is defined as ratio of void volume occupied by crude oil to total void volume of rock in oil reservoir. Allowable capability of fluid passing to the rock in some difference of pressure is called PERM. Recognizing oil-bearing formation means to recognize the characters of each layer in the well. These characters include *oil layer, inferior oil layer, water layer and dry layer*. The experimental data Oilsk81, Oilsk82, Oilsk83, Oilsk84, Oilsk85 come from Jianghan oil field of China. We test each logging data separately while training with the rest four logging data. Table 14 shows a portion of Oilsk81 well logging data and the corresponding conclusions of logging explanations, the sample distributions of 5 logging datasets are shown in Table 15.

6.2. Results and analysis

According to the characteristics of five logging datasets, we employ corresponding AMCS for Oilsk81-Oilsk85 respectively. For example, when testing in Oilsk81, the training set is “low IR, low dimension and small number of classes (*Type 8*)” datasets, AMCS will apply BPSO as feature selection algorithm, Adaboost as the ensemble framework and SVM as the base classifier. While testing in Oilsk84, the corresponding training set is “high IR, low dimension and small number of classes (*Type 5*)” datasets, AMCS will apply BPSO as feature selection algorithm, Adaboost as the ensemble framework and RBF as the base classifier. We test AMCS and the dominating three state-of-the-arts in Section 5.2 (SMOTEBoosting, Easyensemble and PUSBE) on Jianghan logging data, the results are shown in Table 16. Only AMCS makes no mistakes on predicting the character of each layer, while SMOTEBoosting and PUSBE are compatible and Easyensemble makes most mistakes.

Using AUCarea as evaluation metric, we can also clearly justify the performances of algorithms between any two classes. If the AUC of two classes is 1, it demonstrates that there is no mutual misclassification exists between these two classes, which also suggests these two classes are more separable. The AUC less than 1 means that the classifier makes some mistakes when distinguishing a pair of classes, the closer the AUC value to 1, the lower misclassified cases occur. In order

Table 16
Classification results of logging data in terms of accuracy and AUCarea.

	AMCS		SMOTEBoosting		EasyEnsemble		PUSBE	
Datasets	AUCarea	Accuracy	AUCarea	Accuracy	AUCarea	Accuracy	AUCarea	Accuracy
Oilsk81	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
Oilsk82	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9162	0.9592	**1.0000**	**1.0000**
Oilsk83	**1.0000**	**1.0000**	0.9532	0.9800	0.9532	0.9800	**1.0000**	**1.0000**
Oilsk84	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.8867	0.9615	**1.0000**	**1.0000**
Oilsk85	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.8988	0.9231

Table 17
AUC of each pair of layers.

	Oilsk82		Oilsk83			Oilsk84		Oilsk85	
	AMCS	EasyEnsemble	AMCS	SMOTEBoost	EasyEnsemble	AMCS	EasyEnsemble	AMCS	PUSBE
Dry vs. water layer	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Dry vs. oil layer	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Dry vs. inferior oil layer	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Water vs. oil layer	1.0000	1.0000	1.0000	**0.9286**	1.0000	1.0000	1.0000	1.0000	**0.9432**
Water vs. inferior oil layer	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Oil vs. inferior oil layer	1.0000	**0.8701**	1.0000	**0.9286**	**0.9286**	1.0000	**0.8222**	1.0000	**0.6591**

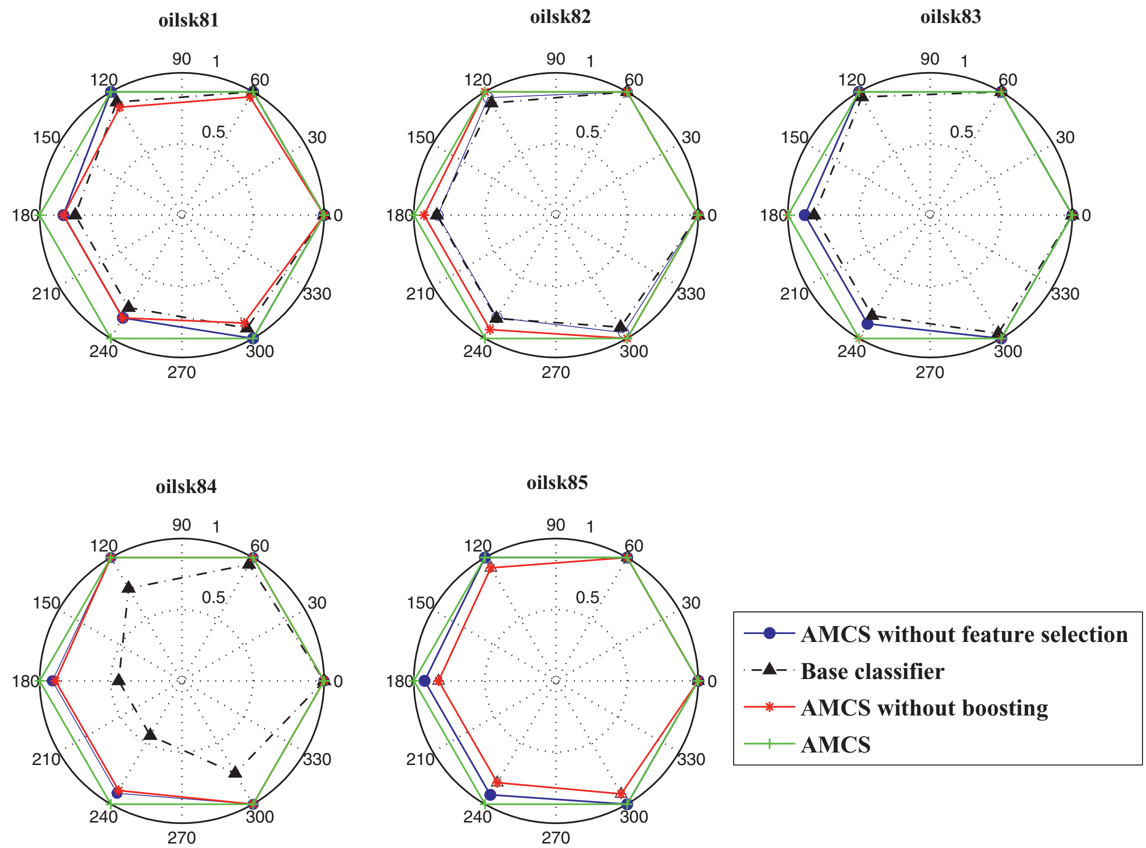

Fig. 10. AUC diagrams obtained in 5 test datasets.

to see the details of misclassifications, we list AUC value of each pair of layers for the algorithms make mistakes on the logging datasets, as is presented in Table 17. It is obvious that most misclassifications happen between oil layer and inferior oil layer. What's more, water layer samples are also misclassified in some cases, since the number of water layer samples is extremely rare. Since recognizing oil layer is the most crucial task for this application, AMCS is significantly better than other three algorithms since it does good job on distinguishing oil layer samples from other layers' samples. At last, we also plot the AUCarea diagram for AMCS in Fig. 10 to clarify how AMCS improves the performance of base classifier.

7. Conclusions

In this paper we propose an adaptive multiple classifier system to deal with imbalanced data classification tasks. The AMCS is framed by employing feature selection and resampling as preprocessors and training with multiple classifiers. We consider two feature selection methods, three ensemble frameworks, five base classifiers and five ensemble rules as options, for different type of imbalanced datasets, the AMCS chooses one specific route of conducing feature selection, resample and ensemble learning. In our study we divide imbalanced data into eight types regarding to their IR, dimension and the number of classes. To find the best routes for different types of imbalanced datasets to form AMCS, we test all the 110 possible routes and set the best route of each type as the adapting criterion. After constructing adapting criteria of AMCS, we compare the AMCS with other state-of-the-arts, the results show that AMCS can outperform or be comparable with many well-known algorithms. Besides, this paper has practical contribution since we applied the proposed algorithm to oil reservoir recognition. The largest advantage of AMCS in this application lies in detecting oil layer efficiently through logging data. The results show that AMCS has significant superiority when distinguishing oil layer from other layers.

When testing the adapting rule for AMCS, we also find some comprehensive guidelines for imbalanced learning. First, over-sampling and under-sampling may not always be a good choice since AMCS always chooses Adaboost as the ensemble framework, in which a filter resample method is employed. Second, for high dimension datasets, filter feature selection method is better than wrapper method for both accuracy and time-complexity. At last, for high IR datasets, decision tree based classifiers and KNN are good choices. For low dimension datasets, SVM outperforms other base classifiers.

Note that in AMCS we did not employ cost sensitive learning due to the difficulty of discriminating cost of misclassifying each class. The cost of misclassification should be case-specific. However, we would like to employ cost sensitive learning in AMCS specially for oil reservoir recognition in the future. For this specific case, when oil layer is misclassified, we would rather to assign oil layer samples into inferior oil layer than into other layers. This is because inferior oil layers are suspected oil layers, which means the companies may not abandon them immediately until it is confirmed that there is no oil exist. This strategy can be implemented by cost sensitive learning, but the cost for each layer remains to be analyzed.

Acknowledgments

This research has been supported by National Natural Science Foundation of China under Grant nos. 71103163, 71573237; New Century Excellent Talents in University of China under Grant no. NCET-13-1012; Research Foundation of Humanities and Social Sciences of Ministry of Education of China under Grant no. 15YJA630019; Special Funding for Basic Scientific Research of Chinese Central University under Grant nos. CUG120111, CUG110411, G2012002A, CUG140604; Open Foundation for the Research Center of Resource Environment Economics in China University of Geosciences (Wuhan) (Grant no. H2015004B); Structure and Oil Resources Key Laboratory Open Project of China under Grant no.TPR-2011-11.

Supplementary Materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.knosys.2015.11.013.

References

[1] J.F. Díez-Pastor, J.J. Rodríguez, C. García-Osorio, et al., Random balance: ensembles of variable priors classifiers for imbalanced data, Knowl.-Based Syst. 85 (2015) 96–111.
[2] M. Galar, A. Fernández, E. Barrenechea, EUSBoost: enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling, Pattern Recogn. 46 (2013) 3460–3471.
[3] P. Vorraboot, S. Rasmequan, K. Chinnasarn, Improving classification rate constrained to imbalanced data between overlapped and non-overlapped regions by hybrid algorithms, Neurocomputing 152 (2015) 429–443.
[4] Z. Sun, Q. Song, X. Zhu, A novel ensemble method for classifying imbalanced data, Pattern Recogn. 48 (2015) 1623–1637.
[5] S.R. Earle, Linear Models for Unbalanced Data, Wiley, 1987.
[6] G. Menardi, N. Torelli, Training and assessing classification rules with imbalanced data, Data Min. Knowl. Discov. (2014) 92–122.
[7] S. Maldonado, R. Weber, F. Famili, Feature selection for high-dimensional class-imbalanced data sets using support vector machines, Inf. Sci. 286 (2014) 228–246.
[8] G.Q. Feng, Y. Li, D.H. Tan, Application of fuzzy closeness degree in reservoir recognition, J. Southwest Pet. Inst. 21 (4) (1999) 46–49.
[9] H.X. Guo, X.W. Liao, K.J. Zhu, Optimizing reservoir features in oil exploration management based on fusion of soft computing, Appl. Soft Comput. 11 (2011) 1144–1155.
[10] N.V. Chawla, K.W. Bowyer, L.O. Hall, SMOTE: synthetic minority over-sampling technique, Artif. Intell. Res. 16 (2002) 321–357.
[11] P. Vorraboot, S. Rasmequan, K. Chinnasarn, Improving classification rate constrained to imbalanced data between overlapped and non-overlapped regions by hybrid algorithms, Neurocomputing 152 (2015) 429–443.
[12] J.F. Díez-Pastor, J.J. Rodríguez, C. García-Osorio, et al., Random balance: ensembles of variable priors classifiers for imbalanced data, Knowl.-Based Syst. 85 (2015) 96–111.
[13] Freund Yoav, Boosting a weak learning algorithm by majority, Inf. Comput. 121 (2) (1995) 256–285.
[14] L. Peng, H. Zhang, B. Yang, et al., A new approach for imbalanced data classification based on data gravitation, Inf. Sci. 288 (2014) 347–373.
[15] Z. Sun, Q. Song, X. Zhu, A novel ensemble method for classifying imbalanced data, Pattern Recogn. 48 (2015) 1623–1637.
[16] V.C. Nitesh, L. Aleksandar, O.H. Lawrence, SMOTEBoost: improving prediction of the minority class in boosting, in: Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, 2003, pp. 107–119.
[17] V. López, S. Río, J.M. Benítez, et al., Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce framework for imbalanced big data, Fuzzy Sets Syst. 258 (2015) 5–38.
[18] B. Krawczyk, M. Wozniak, G. Schaefer, Cost-sensitive decision tree ensembles for effective imbalanced classification, Appl. Soft Comput. 14 (2014) 554–562.
[19] S. Datta, S. Das, Near-bayesian support vector machines for imbalanced data classification with equal or unequal misclassification costs, Neural Netw. 70 (2015) 39–52.
[20] L. Yin, Y. Ge, K. Xiao, Feature selection for high-dimensional imbalanced data, Neurocomputing 105 (2013) 3–11.
[21] M. Alibeigi, S. Hashemi, A. Hamzeh, DBFS: an effective density based feature selection scheme for small sample size and high dimensional imbalanced data sets, Data Knowl. Eng. 81-82 (4) (2012) 67–103.
[22] H. Han, W.Y. Wang, B.H. Mao, Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning, Springer, Berlin Heidelberg, 2005, pp. 878–887. Advances in Intelligent Computing.
[23] W.A.Y. Ii, S.L. Nykl, G.R. Weckman, et al., Using Voronoi diagrams to improve classification performances when modeling imbalanced datasets, Neural Comput. Appl. 26 (2015) 1–14.
[24] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, in: Proceedings of the Thirteenth International Conference on Machine Learning, 1996, pp. 148–156.
[25] X.Y. Liu, J. Wu, Z.H. Zhou, Exploratory Under-sampling for class-imbalance learning, bioinformatics, in: Proceedings of the IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 39(2), 2009, pp. 539–550.
[26] H. Yu, J. Ni, J. Zhao, ACOSampling: an ant colony optimization-based undersampling method for classifying imbalanced DNA microarray data, Neurocomputing 101 (2013) 309–318.
[27] M.D. Martino, A. Fernández, P. Iturralde, Novel classifier scheme for imbalanced problems, Pattern Recogn. Lett. 34 (2013) 1146–1151.
[28] V. López, A. Fernández, S. García, et al., An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics, Inf. Sci. 250 (2013) 113–141.
[29] Tom Fawcett. An introduction to ROC analysis, Pattern Recogn. Lett., 27: 861–874.
[30] M. Richard, E. Jonathan, Multi-class ROC analysis from a multi-objective optimization perspective, Pattern Recogn. Lett. 27 (2006) 916–927.
[31] Md. Hassan, R. Kotagiri, K. Chandan, et al., A Novel scalable multi-class ROC for effective visualization and computation, in: Advances in Knowledge Discovery and Data Mining14th Pacific-Asia Conference June, 21-24, 2010.
[32] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks, 1995, pp. 1942–1948.
[33] W. Bin, P. Qinke, Z. Jing, A binary particle swarm optimization algorithm inspired by multi-level organizational learning behavior, Eur. J. Oper. Res. 219 (2012) 224–233.
[34] J. Kennedy, R.C. Eberhart, A discrete binary version of the particle swarm algorithm, in: IEEE International Conference on Systems Man and Cybernetics Computational Cybernetics and Simulation, 1997.
[35] K. Bache, M. Lichman, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml], University of California, School of Information and Computer Science, Irvine, CA, 2013.
[36] S. Ando, Classifying imbalanced data in distance-based feature space, Knowl. Inf. Syst. (2015) 1–24.
[37] R.C. Prati, E. Gustavo, DiegoF. Silva, Class imbalance revisited: a new experimental setup to assess the performance of treatment methods, Knowl. Inf. Syst. 45 (2015) 247–270.
[38] Z. Liu, H. Cao, X. Chen, et al., Multi-fault classification based on wavelet SVM with PSO algorithm to analyze vibration signals from rolling element bearings, Neurocomputing 99 (1) (2013) 399–410.
[39] X. Wang, S. Matwin, N. Japkowicz, X. Liu, Cost-sensitive boosting algorithms for imbalanced multi-instance datasets, in: Canadian Conference on Artificial Intelligence, 2013, pp. 174–186.
[40] F. Alberto, L. Victoria, G. Mikel, et al., Analysing the classification of imbalanced data-sets with multiple classes: binarization techniques and ad-hoc approaches, Knowl.-Based Syst. 42 (2013) 97–110.
[41] F. Wilcoxon, Individual comparisons by ranking methods, Biometr. Bull. 6 (1945) 80–83.
[42] D. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, Chapman & Hall/CRC, London, U.K., 2006.
[43] S. Holm, A simple sequentially rejective multiple test procedure, Scand. J. Stat. 6 (1979) 65–70.
[44] M.D. Pérez-Godoy, A.J Rivera, C.J. Carmona, et al., Training algorithms for radial basis function networks to tackle learning processes with imbalanced data-sets, Appl. Soft Comput. 25 (2014) 26–39.
[45] RobertE. Schapire, Yoram Singer, Improved boosting algorithms using confidence-rated pre-dictions, Mach. Learn. 37 (3) (1999) 297–336.
[46] Yoav Freund, RobertE. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139.
[47] B. Krawczyk, G. Schaefer, An improved ensemble approach for imbalanced classification problems, in: 8th IEEE International Symposium on Applied Computational Intelligence and Informatics, 2013.
[48] Q.Y. Yin, J.S. Zhang, C.X. Zhang, et al., A novel selective ensemble algorithm for imbalanced data classification based on exploratory undersampling, Math. Prob. Eng. 71 (3) (2014) 741–764.
[49] L. Yin, Y. Ge, K. Xiao, Feature selection for high-dimensional imbalanced data, Neurocomputing 105 (3) (2013) 3–11.

[50] A. Shanab, T. Khoshgoftaar, R. Wald, et al., Comparison of approaches to alleviate problems with high-dimensional and class-imbalanced data, in: IEEE International Conference on Information Reuse and Integration, 2011, pp. 234–239.
[51] C. Seiffert, T.M. Khoshgoftaar, J. Van Hulse, et al., RUSBoost: a hybrid approach to alleviating class imbalance, IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. 40 (1) (2010) 185–197.
[52] P. Vorraboot, S. Rasmequan, C. Lursinsap, et al., Improving classification rate constrained to imbalanced data between overlapped and non-overlapped regions by hybrid algorithms, Neurocomputing 152 (2015) 429–443.
[53] B. Krawczyk, M. Wozniak, F. Herrera, Weighted one-class classification for different types of minority class examples in imbalanced data, in: IEEE Computational Intelligence and Data Mining (CIDM), 2014, pp. 337–344.
[54] J. Tian, H. Gu, W. Liu, Imbalanced classification using support vector machine ensemble, Neural Comput. Appl. 20 (2) (2011) 203–209.
[55] N. Thai-Nghe, Z. Gantner, L. Schmidt-Thieme, Cost-sensitive learning methods for imbalanced data, in: IEEE International Joint Conference Neural Networks (IJCNN), 2010, pp. 1–8.
[56] M. Zięba, J.M. Tomczak, Boosted SVM with active learning strategy for imbalanced data, Soft Comput. (2014) 1–12.
[57] B. Krawczyk, M. Woźniak, Hypertension Type Classification Using Hierarchical Ensemble of One-Class Classifiers for Imbalanced Data, Springer International Publishing, 2015, pp. 341–349.
[58] B.C. Wallace, I.J. Dahabreh, Improving class probability estimates for imbalanced data, Knowl. Inf. Syst. 41 (1) (2014) 33–52.
[59] V. López, A. Fernández, F. Herrera, On the importance of the validation technique for classification with imbalanced datasets: addressing covariate shift when data is skewed, Inf. Sci. 257 (2) (2014) 1–13.
[60] L. Thomas, P.W.D. Robert, A simplified extension of the Area under the ROC to the multiclass domain, in: Proceedings 17th Annual Symposium of the Pattern Recognition Association of South Africa, PRASA, 2006, pp. 241–245.
[61] C. Ferri, J. Hernández-orallo, M.A. Salido, Volume under the ROC surface for multi-class problems. exact computation and evaluation of approximations, in: Proc. of 14th European Conference on Machine Learning, 2003, pp. 108–120.
[62] Lachiche, N., Flach, P.A., 2003. Improving accuracy and cost of two-class and multi-class probabilistic classifiers using roc curves. ICML-2003.
[63] L. Nanni, C. Fantozzi, N. &Lazzarini, Coupling different methods for overcoming the class imbalance problem, Neurocomputing 158 (2015) 48–61.
[64] B. Song, G. Zhang, Z. Wei, et al., ROC operating point selection for classification of imbalanced data with application to computer-aided polyp detection in CT colonography, Int. J. Comput. Assist. Radiol. Surg. 9 (1) (2014) 79–89.
[65] S. Jerzy, N. Krystyna, Addressing imbalanced data with argument based rule learning, Exp. Syst. Appl. 24 (2015) 9468–9481.
[66] L. Yu, H. Liu, L. Yu, et al., Efficient feature selection via analysis of relvance and redundancy, J. Mach. Learn. Res. 5 (4) (2004) 1205–1224.
[67] J. Kittler, M. Hatef, R.P.W. Duin, et al., On combining classifiers, IEEE Trans. Pattern Anal. Mach. Intell. (1998) 226–239.
[68] L. Peng, Y. Chen, B. Yang, et al., A novel classification method based on data gravitation, neural networks and brain, 2005, in: ICNN&B '05. International Conference on IEEE, 2005, pp. 667–672.