CURE-SMOTE algorithm and hybrid algorithm for feature selection and parameter optimization based on random forests

Li Ma and Suohai Fan

Abstract

Background: The random forests algorithm is a type of classifier with prominent universality, a wide application range, and robustness for avoiding overfitting. But there are still some drawbacks to random forests. Therefore, to improve the performance of random forests, this paper seeks to improve imbalanced data processing, feature selection and parameter optimization.

Results: We propose the CURE-SMOTE algorithm for the imbalanced data classification problem. Experiments on imbalanced UCI data reveal that the combination of Clustering Using Representatives (CURE) enhances the original synthetic minority oversampling technique (SMOTE) algorithms effectively compared with the classification results on the original data using random sampling, Borderline-SMOTE1, safe-level SMOTE, C-SMOTE, and k-means-SMOTE. Additionally, the hybrid RF (random forests) algorithm has been proposed for feature selection and parameter optimization, which uses the minimum out of bag (OOB) data error as its objective function. Simulation results on binary and higher-dimensional data indicate that the proposed hybrid RF algorithms, hybrid genetic-random forests algorithm, hybrid particle swarm-random forests algorithm and hybrid fish swarm-random forests algorithm can achieve the minimum OOB error and show the best generalization ability.

Conclusion: The training set produced from the proposed CURE-SMOTE algorithm is closer to the original data distribution because it contains minimal noise. Thus, better classification results are produced from this feasible and effective algorithm. Moreover, the hybrid algorithm’s F-value, G-mean, AUC and OOB scores demonstrate that they surpass the performance of the original RF algorithm. Hence, this hybrid algorithm provides a new way to perform feature selection and parameter optimization.

Keywords: Random forests, Imbalance data, Intelligence algorithm, Feature selection, Parameter optimization

Background

Tin Kam Ho proposed the random forests (RF) concept [1] and the Random Subspace algorithm [2] in 1995 and 1998, respectively. Breiman [3] proposed a novel ensemble learning classification, random forests, by combining bagging ensemble learning and Tin Kam Ho’s concept in 2001. The feature of random forests that allows for avoiding over-fitting makes it suitable for use as a data dimension reduction method for processing data with missing values, noise and outliers. Although random forests have been applied to many other fields such as biological prediction [4], fault detection [5], and network attacks [6], studies seeking to improve the algorithm itself are lacking. The RF algorithm still has some shortcomings; for example, it performs poorly for classification on imbalanced data, fails to control the model during specific operations, and is sensitive to parameter adjustment and random data attempts. Usually, there are two ways to improve RF: increase the accuracy of each individual classifier or reduce the correlation between classifiers.
First, it is possible to increase the classification accuracy in minor class samples of RF for imbalanced training sets through data preprocessing. Several types of methods [7–10] based on both data and algorithms exist for imbalanced data. Chen [11] found that undersampling provides results closer to the original samples than does oversampling for large-scale data. A novel sampling approach [12] based on sub-modularity subset selection was employed to balance the data and select a more representative data subset for predicting local protein properties. Similarly, an algorithm combining RF and a Support Vector Machine (SVM) with stratified sampling [13] yielded a better performance than did other traditional algorithms for imbalanced-text categorization, including RF, SVM, SVM with undersampling and SVM with oversampling. A novel hybrid algorithm [14] using a radial basis function neural network (RFBNN) integrated with RF was proposed to improve the ability to classify the minority class of imbalanced datasets. In addition, imbalanced data for bioinformatics is a well-known problem and widely found in biomedical fields. Applying RF with SMOTE to the CHOM, CHOA and Vero (A) datasets [15] is considered a remarkable improvement that is helpful in the field of functional and structural proteomics as well as in drug discovery. Ali S [16] processed imbalanced breast cancer data using the CSL technique, which imposes a higher cost on misclassified examples and develops an effective Cost-Sensitive Classifier with a GentleBoost Ensemble (Can-CSC-GBE). The Mega-Trend-Diffusion (MTD) technique [17] was developed to obtain the best results on breast and colon cancer datasets by increasing the samples of the minority class when building the prediction model.

Second, it is possible to improve algorithm construction. Because the decision trees in the original algorithm have the same weights, a weighted RF was proposed that used different weights that affected the similarity [18] between trees, out-of-bag error [19], and so on. Weighted RF has been shown to be better than the original RF algorithm [20]. Ma [21] combined Adaboost with RF and adaptive weights to obtain a better performance. The weight of attributes reduces the similarity among trees and improves RF [22]. Moreover, the nearest K-neighbour [23] and pruning mechanism can help achieve a better result when using margin as the evaluation criterion [24].

In this paper, the main work is divided into two parts: first, the CURE-SMOTE algorithm is combined with RF to solve the shortcomings of using SMOTE alone. Compared with results on the original data, random oversampling, SMOTE, Borderline SMOTE1, safe-level-SMOTE, C-SMOTE, and the k-means-SMOTE algorithm, CURE-SMOTE’s effectiveness when classifying imbalanced data is verified. Then, to simultaneously optimize feature selection, tree size, and the number of sub-features, we propose a hybrid algorithm that includes a genetic-random forests algorithm (GA-RF), a particle swarm-random forests algorithm (PSO-RF) and an artificial fish swarm-random forests algorithm (AFSA-RF). Simulation experiments show that the hybrid algorithm obtains better features, selects better parameter values and achieves a higher performance than traditional methods.

**Methods**

**Random forests algorithm review**

**Algorithm principle**

RF is a combination of Bagging and Random Subspace, consisting of many binary or multi-way decision trees \( h_1(x), h_2(x), \ldots, h_{n_{tree}}(x) \), as shown in Fig. 1. The final decision is made by majority voting to aggregate the predictions of all the decision trees. The original dataset \( T = \{(x_{11}, x_{12}, \ldots, x_{1M}, y_1)\}^{N}_{i=1} \) contains \( N \) samples, the vector \( x_{11}, x_{12}, \ldots, x_{1M} \) denotes the \( M \)-dimension attributes or features, \( Y = \{y_i\}^{N}_{i=1} \) denotes classification labels, and a sample is deduced as label \( c \) by \( y_i = c \).

There are two random procedures in RF. First, training sets are constructed by using a bootstrap [25, 26] mechanism randomly with replacement [Fig. 2 (I)]. Second, random features are selected with non-replacement from the total features when the nodes of the trees are split. The size \( \kappa \) of the feature subset is usually far less than the size of the total features, \( M \). The first step is to select \( \kappa \) features randomly, calculate the information gain of \( \kappa \) split and select the best features. Thus, the size of candidate features becomes \( M - \kappa \). Then, continue as shown in Fig. 2 (II).

**Classification rules and algorithmic procedure**

The best attribute can be computed by three methods: information gain, information gain rate and Gini coefficient, which correspond to ID3, C4.5 [27] and CART [28], respectively. When the attribute value is continuous, the best split point must be selected. We use the CART method in this paper; hence, a smaller Gini coefficient indicates a better classification result. Let \( P_i \) represent the proportion of sample \( i \) in the total sample size. Assume that sample \( T \) is divided into \( k \) parts after splitting by attribute \( A \).

\[
Gini(T) = 1 - \sum_{i=1}^{k} P_i^2
\]

\[
Gini(T, A) = \sum_{j=1}^{k} \frac{|T_j|}{|T|} Gini(T_j)
\]

There are several ways by which the termination criteria for RF can be met. For example, termination occurs when the decision tree reaches maximum depth, the impurity of the end node reaches the threshold, the number of final samples reaches a set point, and the candidate attribute is
used up. The RF classification algorithm procedure is shown in Algorithm 1.

**CURE-SMOTE algorithm**

**Definition and impact of imbalanced data**

In recent years, the problem of classifying imbalanced data [29] has attracted increasing attention. Imbalanced data sets generally refer to data that is distributed unevenly among different categories where the data in the smaller category is far less prevalent than data in the larger category. The Imbalance Ratio (IR) is defined as the ratio of the number of minor class samples to the number of major class samples. Therefore, imbalanced data causes the training set for each decision tree to be imbalanced during the first "random" procedure. The classification performance of traditional RF on imbalanced data sets [30] is even worse than that of SVMs [31].

**SMOTE algorithm**

Several methods exist for processing imbalanced data, including sample-based and algorithmic techniques, the combination of sampling and algorithm techniques, and feature selection. In particular, a type of synthesis resampling technique algorithm called the synthetic minority
oversampling technique (SMOTE) [32–34], has a positive effect on the imbalanced data problem. The specific idea is implemented as follows: obtain the \( k \)-nearest neighbours of sample \( X \) in the minor class, select \( n \) samples randomly and record them as \( X_i \). Finally, the new sample \( X_{\text{new}} \) is defined by interpolation as follows:

\[
X_{\text{new}} = X_{\text{origin}} + \text{rand} \times (X_i - X_{\text{origin}}), i = 1, 2, ..., n,
\]

where \( \text{rand} \) is a random number uniformly distributed within the range (0,1), and the ratio for generating new samples approximates \( 1/IR - 1 \).

However, some flaws exist in the SMOTE algorithm. First, the selection of a value for \( k \) is not informed by the nearest neighbours selection. Second, it is impossible to completely reflect the distribution of original data because the artificial samples generated by the minor class samples at the edges may lead to problems such as repeatability and noisy, fuzzy boundaries between the positive and negative classes.

Therefore, researchers have sought to improve the SMOTE algorithm. The Borderline–SMOTE1 algorithm [35] causes new samples to be more effective using interpolation along the border areas, but it fails to find all the boundary points. Definitions for this algorithm are shown in Table 1: \( m \) is the number of nearest-neighbour samples in the minor class, and \( k \) is the number of samples in the major class.

Motivated by Borderline–SMOTE1, safe-level-SMOTE [36] advocates calculating the safe level of minor class samples, but it can easily fall into overfitting. Cluster-SMOTE [37] obtains a satisfactory classification effect for imbalanced datasets by using K-means to find clusters of minor class samples and then applying SMOTE. In addition, spatial structures have been studied such as N-SMOTE [38] and nuclear SMOTE [39]. The authors of [40] proposed an interpolation algorithm based on cluster centres. SMOTE was combined with a fuzzy nearest-neighbour algorithm in [41]. In [42], a preferable classification effect promoted by hierarchical clustering sampling was shown. Recently, a SMOTE noise-filtering algorithm [43] and MDO algorithms with Markov distance [44] have been proposed. In general, many improved versions of the SMOTE algorithm have been proposed, but none of these improvements seem perfect. This paper seeks to solve the shortcomings of SMOTE.

The K-means algorithm is effective only for spherical datasets and its application requires a certain amount of time. The CURE [45] hierarchical clustering algorithm is efficient for large datasets and suitable datasets of any shape dataset. Moreover, it is not sensitive to outlier and can recognize abnormal points. Consequently, CURE is better than the BIRCH, CLARANS and DBSCAN algorithms [46]. In the CURE algorithm, each sample point is assumed to be a cluster. These points are merged using local clustering until the end of the algorithm. Thus, the CURE algorithm is appropriate for distributed extensions. In this paper, inspired by C-SMOTE [40] and the hierarchical clustering sampling adaptive semi-supervised weighted oversampling (A-SUWO)
[42] algorithms, the novel CURE-SMOTE algorithm is proposed to accommodate a wider range of application scenarios.

**Design and analysis of CURE-SMOTE**

The general idea of the CURE-SMOTE algorithm is as follows: cluster the samples of the minor class using CURE, remove the noise and outliers from the original samples, and then, generate artificial samples randomly between representative points and the centre point. The implementation steps of the CURE-SMOTE algorithm are as follows:

**Step 1.** Normalize the dataset, extract the minor class samples, \( X \), and calculate the distance \( dist \) among them. Each point is initially considered as a cluster. For each cluster \( U \), \( U_r \) and \( U_c \) represent the representative set and the centre point, respectively. For two data items \( p \) and \( q \), the distance between the two clusters \( U \) and \( V \) is:

\[
dist(U, V) = \min_{p \in U, q \in V} \min \{ p, q \}
\]

**Step 2.** Set the clustering number, \( c \), and update the centre and representative points after clustering and merging based on the smallest distance of the two clusters,

\[
U_c = \frac{|U||U_c| + |V||V_c|}{|U| + |V|}
\]

\[
U_r = \{ p + \alpha \cdot (U_c - p) | p \in U_r \}
\]

where \(|U|\) is the number of data items for class \( U \), and the shrinkage factor \( \alpha \) is generally 0.5. The class with slowest growth speed is judged to contain abnormal points and will be deleted. If the number of representative points is larger than required, select the data point farthest from the clustering centre as the first representative point. Then, the next representative point is the one farthest from the former. When the number of clustering centres reaches a predefined setting, the algorithm terminates, and clusters containing only a few samples are removed.

**Step 3.** Generate a new sample according to the interpolation formula. \( \overline{X} \) represents the samples after clustering by the CURE algorithm.

\[
X_{\text{new}}^n = \overline{X} + \text{rand}(0, 1) \times (U_r - \overline{X}).
\]

**Step 4.** Calculate \( IR \), and return to Step 3 if \( IR \leq IR_{\text{th}} \).

**Step 5.** Finally, classify the new dataset as \( X_{\text{new}} = X \cup \{X_{\text{new}}^n\} \) and add samples of the major class by RF. The distance is measured using Euclidean distance.

For example, the distance between sample \( X_1 = (X_{11}, X_{12}, \ldots, X_{1M}) \) and sample \( X_2 = (X_{21}, X_{22}, \ldots, X_{2M}) \) is \( d_{12} = \sqrt{\sum_{j=1}^{M} (X_{1j} - X_{2j})^2} \).

During the clustering process of the CURE-SMOTE algorithm, noisy points must be removed because they are far away from the normal points, and they hinder the merge speed in the corresponding class. When clustering is complete, the clusters containing only a few samples are also deemed to be noisy points. For the sample points after clustering, the interpolation can effectively prevent generalization and preserve the original distribution attributes of the data set. In the interpolation formula, \( X_i \) is replaced by the representative points; consequently, the samples are generated only between the representative samples and the samples in the original minor class, which effectively avoids the influence of boundary points. The combination of the clustering and merge operations serves to eliminate the noise points at the end of the process and reduce the complexity because there is no need to eliminate the farthest generated artificial samples after the SMOTE algorithm runs. Moreover, all the termination criteria such as reaching the pre-set number of clusters, the number of representative samples, or the distance threshold, avoid setting the \( k \) value of the original SMOTE algorithm and, thus, reduce the instability of the proposed algorithm.

**Research concerning feature selection and parameter optimization**

Classification [47] and feature selection [48–50] are widely applied in bioinformatics applications such as gene selection [51, 52] and gene expression [53–55]. Chinnaswamy A [56] proposed a hybrid feature selection using correlation coefficients and particle swarm optimization on microarray gene expression data. The goal of feature selection is to choose a feature subset that retains most of the information of the original dataset, especially for high-dimensional data [57]. The authors of [58] showed that machine-learning algorithms achieve better results after feature selection. Kausar N. [59] proposed a scheme-based RF in which useful features were extracted from both the spatial and transform domains for medical image fusion. During the second “random” time of RF, a number of attributes were selected randomly to reduce the correlation between trees, but this operation promotes redundant features that may affect the generalization ability to some degree. Thus, new types of evaluation mechanisms were proposed based on the importance of the attributes [21, 60, 61], using weighted features as well as cost-sensitivity features [62], and so on; however, their calculations are comparatively complicated. Recently, researchers
have combined the RF algorithm with intelligent algorithms. Such combinations have achieved good results in a variety of fields. In [5], an improved feature selection method based on GA and RF was proposed for fault detection that significantly reduces the OOB error. The results of [4, 6] indicate that a type of hybrid PSO-RF feature selection algorithm is widely applied in certain fields. However, the works mentioned above do not involve parameter optimization.

Three main parameters influence the efficiency and performance of RF: nTree—the size of the tree, MinLeaf—the minimum sample number of leaf nodes, and \( \kappa \)—the attribute subset size. Previous studies have shown that the classification performance of RF is less sensitive to MinLeaf [63]. A larger nTree increases the number of trees in the classifier, helps ensure the diversity of individual classifiers and, thus, improves performance. However, a larger nTree also increases the time cost and may lead to less interpretable results, while a small nTree results in increased classification errors and poor performance. Usually, \( \kappa \) is far less than the number of total attributes [64]. When all the similar attributes are used for splitting the tree nodes in the Bagging algorithm, the effect of the \( \kappa \) on the decision tree worsens due to the higher similarity degree among trees [65]; when \( \kappa \) is smaller, the stronger effects of randomness lower the classification accuracy. The hyper parameter \( \kappa \) behaves differently for different issues [66]; hence, an appropriate value can cause the algorithm to have excellent performance for a specific problem. Breiman pointed out that selecting the proper \( \kappa \) value has a great influence on the performance of the algorithm [3] and suggested that the value should be \( 1, \sqrt{M}, \frac{1}{2} \sqrt{M}, 2^{\sqrt{M}} \) and \( \lfloor \log_2(M) \rfloor + 1 \). Generally, \( \kappa \) is fixed as \( \sqrt{M} \), but that value does not guarantee obtaining the best classifier. Therefore, the authors of [67] suggested that the minimum OOB error be used to obtain the approximate value to overcome the shortcomings of the orthogonal validation method. Moreover, OOB data has been used to estimate the optimal training sample proportion to construct the Bagging classifier [68]. To sum up, it is difficult for traditional parameter values to achieve an optimal performance. In terms of the search for the optimal parameter, typical approaches have incorporated exhaustive search, grid search, and orthogonal selection, but these methods have a high time complexity.

**Review of intelligent algorithms**

Because intelligent algorithms are superior for solving NP-hard problems and for optimizing parameters, they have been the subject of many relevant and successful studies [69–72].

The main idea behind the genetic algorithm (GA) is to encode unknown variables into chromosomes and change the objective function into fitness functions. The fitness value drives the main operations—selection, crossover and mutation—to search for the best potential individuals iteratively. Eventually the algorithm converges, and the optimal or a suboptimal solution of the problem is obtained. GA has the advantage of searching in parallel, and it is suitable for a variety of complex scenarios.

The particle swarm optimization (PSO) algorithm is theoretically simpler and more efficient than the GA [73]. The main idea behind PSO is to simulate the predation behaviour of birds. Each particle represents a candidate solution and has a position, speed and a fitness value. Historical information on the optimal solution instructs the particle to fly toward a better position.

The artificial fish swarm algorithm (AFSA) [74] is a novel algorithm with high potential. The main idea behind AFSA is to imitate the way that fish prey, swarm, follow and adopt random behaviours. The candidate solution is translated into the individual positions of the fish, while the objective function is converted to food concentration.

Diagrams for GA, PSO and AFSA are shown in Fig. 3. There is little research on optimizing the hyper parameter \( \kappa \) of random forests. In [67], the size of the decision tree is fixed at 500, but this approach achieves the optimal parameter on only half the dataset. Worse, it requires considerable time and is suitable for single parameter optimization only. This paper proposes combining a new hybrid algorithm for feature selection and parameter optimization with RF is proposed based on [4–6].

**The proposed hybrid algorithm for feature selection and parameter optimization**

We propose the hybrid GA-RF, PSO-RF or AFSA-RF algorithm for feature selection, parameter optimization and classification. The algorithm seeks to remove redundant features and attain the optimal feature subset and, finally, to explore the relation between performance and nTree, as well as the hyper parameter \( \kappa \).

Generally, \( p \)-fold cross validation is used to traverse the parameter and to estimate the algorithm in the experiment, but time complexity is high. In this paper, OOB error replaces the cross-validation algorithm for binary classification, while the full misclassification error is used for multi-classification. Hence, the time complexity is reduced to \( 1/p \). During the process, cross validation is required for classification.

Objective function:

\[
  f(n\text{Tree}^*, \kappa^*, \{\text{Attribute}|j = 1, 2, ..., M\}) = \arg \min (\text{avg OOB error})
\]  

(8)

Studies have shown that the larger nTree is, the more stable the classification accuracy will be. We set nTree
and $\kappa$ in the range $[0, 500]$ and $[1, M]$, respectively, by considering both the time and space complexities.

Optimization variables: $n_{Tree}$, $\kappa$, $\{\text{Attribute}_i | i = 1, 2, ..., M\}$

Binary encoding involves two tangent points and three steps. Let $n_{Tree}$ and $\kappa$ be numbers in the binary system. A value of 0 in $\{\text{Attribute}_i | i = 1, 2, ..., M\}$ represents an unselected feature in the corresponding position, while a 1 represents the selected features.

The constraint condition is $\kappa \leq \sum_{i=1}^{M} \text{Attribute}_i$.

Then, an $n_{Tree}$ is generated randomly between $[0, 500]$. Because $2^9 = 512$, a 9-bit length ensures a full set of variables. The bits used for $\kappa$ and the bits used for the attributes are different for different data sets. The bits of $\kappa$ are the binary representation of $M$, while the number of bits of the attributes are $M$ (Fig. 4). The initialization continues until a valid variable is generated.

The diagram for a hybrid algorithm based on RF and an artificial algorithm for feature selection and parameter optimization is shown in Fig. 5.

### Hybrid GA-RF

**Step 1.** Initialize the population: Perform binary encoding. The population size is set to $popsize$, the max iteration time is set to $maxgen$, the crossover probability is $P_c$, and the mutation probability is $P_m$.

**Step 2.** Combine the GA with RF classification and calculate the fitness function, $F = \max(1/f)$, gen = 1. **Step 3.** Perform the selection operation with the roulette method: the probability of selecting an individual is dependent on the proportion of the overall fitness value that the individual represents:

$$p_i = \frac{F_i}{\sum_{i=1}^{popsize} F_i}.$$  \hspace{1cm} (9)

**Step 4.** Conduct the crossover operation with the single-point method: two selected individuals cross at a random position with different values. The offspring generation will be regenerated until it turns out to be legal. The process is shown in Fig. 6.

**Step 5.** Mutation operation: select an individual and a position $j$ randomly to mutate by switching 0 and 1. When a feasible solution is achieved, calculate the fitness value and update the optimal solution. The mutation operation is shown in Fig. 7.

**Step 6.** When gen > maxgen, the algorithm will terminate; otherwise, return to Step 3.

| $n_{Tree}$ | $\kappa$ | Attributes |
|------------|----------|------------|
| 0 0 1 1    | 0 1 0    | 0 1 1 0    |
| 1 1 1 1    | 1 1 1    | 1 1 1      |

*Fig. 4 Binary coding*
Hybrid PSO-RF

**Step 1.** Initialize the population. The population size is set to $popsize$, the max iteration time is set to $maxgen$, the position of the binary particle is $X_k = \{Z_{k,1}, Z_{k,2}, \ldots\}$, $k = 1, 2, \ldots, popsize$, the velocity is $V$, the learning factors are $c_1, c_2$, and the weight is $w$.

**Step 2.** Combine the PSO with RF classification and calculate the fitness function $F = \max(1/f)$, $\text{gen} = 1$.

**Step 3.** Update the velocities $V_{k+1}$ and positions $X_{k+1}$ of particles. Let $P^k$ be the optimal position of an individual particle, $P^k_{g}$ be the optimal position of all particles, and $\text{rand}$ be a random number uniformly distributed in the range $(0, 1)$:

$$V_{k+1} = wV_k + c_1 r_1 (P^k - X_k) + c_2 r_2 (P^k_{g} - X_k), r_1, r_2 \in [0, 1]$$  \hspace{1cm} (10)

$$\text{sigmoid}(V_{k+1}) = \frac{1}{1 + e^{-V_{k+1}}},$$  \hspace{1cm} (11)

**Step 4.** If $\text{gen} > maxgen$, the algorithm will terminate; otherwise, return to Step 3.

Hybrid AFSA-RF

**Step 1.** Initialize the population. The population size is set to $popsize$, the max iteration number is set to $maxgen$, the fish positions are $X_k = \{Z_{k,1}, Z_{k,2}, \ldots\}$, $k = 1, 2, \ldots, popsize$, the visual distance is $\text{visual}$, the crowding degree factor is $\text{delta}$, and the maximum number of behaviours to try is $\text{try\_number}$.

**Step 2.** Combine with RF classification and calculate the food concentration $F = \max(1/f)$.

**Step 3.** Swarm and follow at the same time.

a) Swarm behaviour: The current state of a fish is $X_i$, the number of partners in view is $n_f$, and the centre position is $X_c$. When $\frac{F_i}{n_f} > \text{delta}\cdot \text{Fitness}_i$, move to the centre position according to $Z_{k+1,j}$:

$$Z_{k+1,j} = \begin{cases} \text{rand} > \text{sigmoid}(V_{k+1}) & \text{rand} \leq \text{sigmoid}(V_{k+1}) \\ \text{rand} \in (0, 1). \end{cases}$$  \hspace{1cm} (12)

**Step 4.** If $\text{gen} > maxgen$, the algorithm will terminate; otherwise, return to Step 3.
to the following formula; otherwise, conduct the prey behaviour.

\[
Z_{k+1,i} = \begin{cases} 
Z_{k,i} = Z_{c,i} \\
0 & Z_{k,i} \neq Z_{c,i}, \text{rand} > 0.5 \\
1 & Z_{k,i} \neq Z_{c,i}, \text{rand} \leq 0.5.
\end{cases}
\] (13)

b) Follow behaviour: Find the fish \( X_{\text{max}} \) with the maximum food concentration value, \( F_{\text{max}} \).

If \( \frac{F_{\text{max}}}{m} > \delta F \), move to \( X_{\text{max}} \) and calculate the food concentration value. Then, update the food concentration value by comparing it with the value of the swarm behaviour; otherwise, conduct the prey behaviour.

\[
Z_{k+1,i} = \begin{cases} 
Z_{k,i} = Z_{\text{max},i} \\
0 & Z_{k,i} \neq Z_{\text{max},i}, \text{rand} > 0.5 \\
1 & Z_{k,i} \neq Z_{\text{max},i}, \text{rand} \leq 0.5.
\end{cases}
\] (14)

c) Prey behaviour: The current state is \( X_k = [Z_{d,i}] \), and the random selection state is \( X_i = [Z_{r,i}] \) around the vision range with \( d_{ij} = \text{visual} \). When \( F_k > F_i \), restart to generate the next state, \( X_{k+1} \), and calculate the food concentration value until \( \text{try_number} \) is reached; otherwise, terminate the prey behaviour according to the following function:

\[
Z_{k+1,i} = \begin{cases} 
Z_{k,i} = Z_{i,i} \\
0 & Z_{k,i} \neq Z_{i,j}, \text{rand} > 0.5 \\
1 & Z_{k,i} \neq Z_{i,j}, \text{rand} \leq 0.5.
\end{cases}
\] (15)

Step 4. Update the state of the optimal fish. When \( \text{gen} > \text{maxgen} \), the algorithm will terminate; otherwise, return to Step 3.

Results and discussion

The experiments in this paper are divided into two parts. Experiment 1 explores the validity of the CURE-SMOTE algorithm. Experiment 2 investigates the effectiveness of the hybrid algorithm.

Performance evaluation criteria

Referring to the evaluation used in [75], the measures of the quality of binary classification are built using a confusion matrix, where TP and FN are the numbers of correctly and incorrectly classified compounds of the actual positive class, respectively. Similarly, TN and FP denote the numbers of correctly and incorrectly classified compounds of the actual negative class.

| Table 2 Dataset | Id | Dataset | N  | M  | Positive class | Negative class | IR       | Label          |
|-----------------|----|---------|----|----|----------------|----------------|---------|----------------|
| 1 Circle        | 1  | 1362    | 2  | 229| 1133           | 0.2021:1       | 1:0     |                |
| 2 Blood-transfusion | 2 | 748     | 4  | 178| 570            | 0.3123:1       | 4:2     |                |
| 3 Haberman’s survival | 3 | 306     | 3  | 81 | 225            | 0.361:1        | 2:1     |                |
| 4 Breast-cancer-wisconsin | 4 | 702     | 10 | 243| 459            | 0.5249:1       | 1:0     |                |
| 5 SPECT.train   | 5  | 80      | 23 | 26 | 54             | 0.4815:1       | 1:0     |                |

The measures accuracy, sensitivity, specificity and precision are defined as follows.

\[
\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} = \frac{(TP + TN)}{N}
\] (16)

\[
\text{Sensitivity or Recall} = \frac{TP}{(TP + FN)}
\] (17)

\[
\text{Specificity} = \frac{TN}{(FP + TN)}
\] (18)

\[
\text{Precision} = \frac{TP}{(TP + FP)}
\] (19)

The classifiers may have a high overall accuracy with 100% accuracy in the majority class while achieving only a 0–10% accuracy in the minority class because the overall accuracy is biased towards the majority class. Hence, the accuracy measure is not a proper evaluation metric for the imbalanced class problem. Instead, we suggest using F-value, Geometric Mean (G-mean) and AUC for imbalanced data evaluations.

The F-value measure is defined following [26]. A larger F-value indicates a better classifier. F-value is a performance metric that links both precision and recall:

\[
F = \frac{2}{1/\text{Precision} + 1/\text{Recall}}.
\] (20)

The G-mean [76] attempts to maximize the accuracy across the two classes with a good balance and is defined as follows. Only when both sensitivity and specificity are high can the G-mean attain its maximum, which indicates a better classifier:

\[
\text{G–mean} = \sqrt{\text{Sensitivity} \cdot \text{Specificity}}.
\] (21)

AUC is the area under the receiver operating characteristics (ROC) curve. AUC has been shown to be a reliable performance measure for imbalanced and cost-

| Table 3 Comparison of algorithms and references |
|-----------------------------------------------|-----------------------------------------------|
| Algorithm | Reference | Algorithm | Reference |
|-----------|-----------|-----------|-----------|
| SMOTE     | [32]      | Safe-level SMOTE | [36]      |
| Borderline-SMOTE 1 | [35] | C-SMOTE   | [36]      |
| k-means-SMOTE  | [37] | -         | -         |
Fig. 8 CURE-SMOTE algorithm diagram

Fig. 9 Artificial samples generated by different methods
sensitive problems. An AUC–based permutation variable is presented in [77]; this approach is more efficient than the approach based on the OOB error.

The training set is obtained by using the bootstrap method. Because of repeated extraction, it contains only 63% of the original data; the 37% of the data that never appear are called "out-of- -bag" (OOB) data [78]. OOB estimation is an unbiased estimate of the RF algorithm and can be used to measure the classifier’s generalization ability. A smaller OOB error indicates a better classification performance. OOB error is defined as follows:

$$\text{OOB error} = \frac{1}{n_{\text{Tree}}} \sum_{i} \text{OOB error}_i.$$  \hspace{1cm} (22)

Margin is a new evaluation criterion that has been applied to the classification of remote sensing data [79]. The larger the margin is, the higher the classifier’s credibility is:

$$\text{margin} = \frac{1}{n_{\text{Tree}}} \sum_{i} \text{margin}_i.$$  \hspace{1cm} (23)

Experiment 1 and parameter settings

The experiments were implemented using Matlab 2012a on a workstation with a 64-bit operating system, 2 GB of RAM and a 2.53 GHz CPU. Artificial Data Circle and UCI imbalanced datasets were selected for the experiments. More detailed information about five datasets is listed in Table 2. To simulate the actual situation appropriately and preserve the degree of imbalance of the original data, the training set and testing set were divided using stratified random sampling at a ratio of 3:1, except for SPECT. The SPECT.test dataset incorporates 187 samples, and the proportions of the classes labelled 1 and 0 are 84:103, respectively. The tree size is 100 and the depth is 20.

To verify the effectiveness of the CURE-SMOTE algorithm it was compared with the original data, random oversampling, SMOTE, Borderline-SMOTE1, safe-level SMOTE, C-SMOTE (using mean value as the centre) and k-means-SMOTE (shown in Table 3) algorithms. To evaluate the performance of the different algorithms, F-value, G-mean, AUC and OOB error are used as performance measures. The results of each experiment were averaged over 100 runs to eliminate random effects.

To facilitate the comparisons, \(m\) and \(k\) were set to 20 and 5, respectively, in SMOTE, Borderline-SMOTE1 and
| Dataset                  | Method                | F         | G-Mean    | AUC       | OOB error |
|-------------------------|-----------------------|-----------|-----------|-----------|-----------|
| 1. Circle               | Original data         | 0.9081    | 0.9339    | 0.9389    | 0.0296    |
|                         | Random oversampling   | 0.9249    | 0.9553    | 0.9567    | 0.0163    |
|                         | SMOTE                 | 0.9086    | 0.9535    | 0.9579    | 0.0348    |
|                         | Borderline-SMOTE1     | 0.9110    | 0.9534    | 0.9619    | 0.0438    |
|                         | Safe-level-SMOTE      | 0.9146    | 0.9595    | 0.9559    | 0.0431    |
|                         | C-SMOTE               | 0.9302    | 0.9713    | 0.9813    | 0.0702    |
|                         | k-means-SMOTE         | 0.9262    | 0.9589    | 0.9602    | 0.0323    |
|                         | CURE-SMOTE            | **0.9431** | **0.9808** | **0.9855** | **0.0323** |
| 2. Blood-transfusion    | Original data         | 0.3509    | 0.5094    | 0.5083    | 0.2548    |
|                         | Random oversampling   | 0.3903    | 0.5490    | 0.5449    | 0.2250    |
|                         | SMOTE                 | 0.4118    | 0.5798    | 0.5537    | 0.2152    |
|                         | Borderline-SMOTE1     | 0.4185    | 0.5832    | 0.5424    | 0.1630    |
|                         | Safe-level-SMOTE      | 0.4494    | 0.6174    | 0.5549    | 0.2479    |
|                         | C-SMOTE               | 0.4006    | 0.5549    | 0.5531    | 0.2418    |
|                         | k-means-SMOTE         | 0.4157    | 0.5941    | 0.5433    | 0.1872    |
|                         | CURE-SMOTE            | **0.5393** | **0.6719** | **0.6533** | **0.2531** |
| 3. Haberman’s survival  | Original data         | 0.3279    | 0.5018    | 0.6063    | 0.3149    |
|                         | Random oversampling   | 0.3504    | 0.5178    | 0.5959    | 0.1534    |
|                         | SMOTE                 | 0.4350    | 0.5971    | 0.6259    | 0.1728    |
|                         | Borderline-SMOTE1     | 0.4523    | 0.6119    | 0.6298    | 0.2589    |
|                         | Safe-level-SMOTE      | 0.4762    | 0.6008    | 0.6030    | 0.3077    |
|                         | C-SMOTE               | 0.4528    | 0.5487    | 0.5656    | 0.2780    |
|                         | k-means-SMOTE         | 0.4685    | 0.6249    | 0.6328    | 0.1828    |
|                         | CURE-SMOTE            | **0.5000** | **0.6282** | **0.6940** | **0.2717** |
| 4. Breast-cancer-wisconsin | Original data         | 0.9486    | 0.9619    | 0.9491    | 0.0446    |
|                         | Random oversampling   | 0.9451    | 0.9623    | 0.9620    | 0.0301    |
|                         | SMOTE                 | 0.9502    | 0.9666    | 0.9627    | 0.0341    |
|                         | Borderline-SMOTE1     | 0.9506    | 0.9661    | 0.9635    | 0.0379    |
|                         | Safe-level-SMOTE      | 0.9509    | 0.9671    | 0.9638    | 0.0404    |
|                         | C-SMOTE               | 0.9491    | 0.9636    | 0.9561    | 0.0380    |
|                         | k-means-SMOTE         | 0.9449    | 0.9616    | 0.9562    | 0.0373    |
|                         | CURE-SMOTE            | **0.9511** | **0.9664** | **0.9621** | **0.0427** |
| 5. SPECT.train          | Original data         | 0.6348    | 0.6764    | 0.6579    | 0.3634    |
|                         | Random oversampling   | 0.6539    | 0.6924    | 0.6753    | 0.3468    |
|                         | SMOTE                 | 0.6618    | 0.6990    | 0.6825    | 0.3688    |
|                         | Borderline-SMOTE1     | 0.6710    | 0.6926    | 0.6746    | 0.3489    |
|                         | Safe-level-SMOTE      | 0.6770    | 0.7074    | 0.6913    | 0.3160    |
|                         | C-SMOTE               | 0.6564    | 0.6936    | 0.6764    | 0.3448    |
|                         | k-means-SMOTE         | 0.6796    | 0.6941    | 0.6846    | 0.3599    |
|                         | CURE-SMOTE            | **0.6855** | **0.7155** | **0.6951** | **0.1108** |

From the classification results obtained by the different sampling algorithms discussed in Table 4, the best F-value, G-mean and AUC were achieved on the Circle dataset by CURE-SMOTE, and its OOB error is second-best, behind only random sampling. The overall classification result on the blood-transfusion dataset is poorer, but the CURE-SMOTE algorithm achieves the best F-value, G-mean and AUC, while its OOB error is inferior to the original data. On the Haberman’s survival dataset, the F-value, G-mean and AUC achieved by CURE-SMOTE are superior to the other sampling algorithms. For the breast-cancer-wisconsin dataset, CURE-SMOTE achieves the best F-value, but its G-mean and AUC are slightly lower, although they are little different from the other sampling algorithms. On the SPECT dataset, CURE-SMOTE surpasses the other sampling algorithms with regard to F-value, G-mean, AUC and OOB error.

The best value of every performance evaluation criteria obtained by the algorithms are marked in boldface.
safe-level-SMOTE. The number of clusters in C-SMOTE and k-means-SMOTE were set to five. Following the suggested setting for the CURE algorithm, the cluster results are better when the constriction factor is in the range [0.2, 0.7] and when the number of representative points is greater than 10. Thus, the constriction factor was set to 0.5 and the number of representative points was set to 15. The number of clusters was set to two in the circle, while the others were all five. Samples were removed when the number of representative points did not increase for ten iterations or when the sample size of the cluster class was less than 1/(10c) of the total sample size when clustering was complete. In the experiments in this paper, IR0 was fixed at 0.7. The CURE-SMOTE algorithm diagram is depicted in Fig. 8.

Results and discussion of CURE - SMOTE algorithm
Figure 9 shows the results of the original data, random sampling, SMOTE sampling, Borderline-SMOTE1 sampling, safe-level SMOTE sampling, C-SMOTE sampling, K-means SMOTE sampling and CURE-SMOTE sampling, as well as the CURE clustering result. The black circles and the red star represent the major class sample and minor class sample, respectively, in the original data, and the blue squares represent the artificial samples generated by different methods. Figure 10 shows the CURE clustering results of the minor class sample. The clustering centre is two, the stars show the centres, and the blue diamonds indicate the representative points.

Figure 9 shows that a large number of data are obtained repeatedly by random sampling, and some data are not selected at all. The SMOTE algorithm also produces repeated data and generates mixed data in other classes as well as noise. Borderline-SMOTE1 picks out the boundary point of minor class by calculating and comparing the samples of the major class around the minor class; consequently, the generated data are concentrated primarily at the edges of the class. Safe-level SMOTE follows the original distribution, but still generates repeated points and distinguishes the boundary incorrectly. Although C-SMOTE can erase the noise, the generated data are too close to the centre to accurately identify other centres. K-means-SMOTE can identify the area of the small class and slightly improves on the SMOTE effect. The proposed CURE-SMOTE algorithm generates data both near the centre and the representative points; overall, it follows the original distribution. Moreover, the representative points help to avoid noise being treated as a constraining boundary during the generating process. Detailed results are listed in Table 4.

In conclusion, the classification results of the CURE-SMOTE algorithm as measured by the F-value, G-means, and AUC are substantially enhanced, whereas the results using SMOTE alone are not particularly stable. Meanwhile, Borderline-SMOTE1, C-SMOTE, and the k-means-SMOTE algorithm are even worse than random sampling on some datasets. Thus, the CURE-SMOTE algorithm combined with RF has a substantial effect on classification.

Experiment 2 and parameter settings
In this section, to test the effectiveness of the hybrid algorithm for feature selection and parameter optimization, we selected the representative binary classification and multi-classification imbalanced datasets shown in Table 5. These data are randomly stratified by sampling them into four parts with a training set to testing set ratio of 3:1. In this procedure, 4-fold stratified cross validation is used for classification. The parameter settings are listed in Table 6. The depth is set to 20 for experiment 2.

Results and discussion of the hybrid algorithm
According to the proposed settings in previous works, the parameters for all of the methods were set as follows: nTree = 100, κ = 1, \( \sqrt{M} \), \( \lfloor \log_2(M) + 1 \rfloor \) and M. Accuracy, OOB error and margin were selected as the evaluation criteria. The detailed results are listed in Table 7 and Table 8.

### Table 5 Dataset

| id | Dataset           | N   | M   | Positive class | Negative class | IR   | Label |
|----|-------------------|-----|-----|----------------|----------------|------|-------|
| 1  | Connectionist Bench | 208 | 17  | 97            | 111           | 0.8739 | RM    |
| 2  | Wine              | 130 | 13  | 59            | 71            | 0.831 | 1:2   |
| 3  | Ionosphere        | 351 | 34  | 126           | 225           | 0.56  | b:g   |
| 4  | Breast-cancer-wisconsin | 702 | 10  | 243           | 459           | 0.5249 | 10 |
| 5  | Steel Plates Faults | 1,941 | 27 | -             | -             | 7     | labels |
| 6  | Libras Movement   | 360 | 90  | -             | -             | 15    | labels |
| 7  | mfeat-factors     | 2,000 | 216 | -             | -             | 10    | labels |

### Table 6 Parameter settings

| Hybrid | GA-RF | PSO-RF | AFSA-RF |
|--------|-------|--------|---------|
| popsize | 5     | 5      | 5       |
| maxgen | 20    | 20     | 20      |
| Pc     | 0.6   |        |         |
| Pm0.1  |       |        |         |
| c1/1.5 |       | c2/1.5 |         |
| r1, r2∈[0,1] |       | Vmin/Vmax = -0.5/0.5 | w∈[0,5] |
| visual | 3     |        |         |
| try_number | 5     |        |         |
| delta  | 0.618 |        |         |
### Table 7 The binary classification results

|                      | 1  | $\sqrt{M}$ | $\lceil \log_2(M) + 1 \rceil$ | $M$ | GA-RF | PSO-RF | AFSA-RF |
|----------------------|----|-------------|-------------------------------|-----|-------|--------|---------|
| **Connectionist Bench** |    |             |                               |     |       |        |         |
| Accuracy             | 0.6442 | 0.6442 | 0.6058 | 0.6635 | 0.6538 | 0.7308 | 0.6827 |
| Sensitive            | 0.5882 | 0.6122 | 0.6500 | 0.7556 | 0.5741 | 0.6744 | 0.5870 |
| Precision            | 0.6522 | 0.6250 | 0.4906 | 0.5862 | 0.7045 | 0.6744 | 0.6585 |
| Specificity          | 0.6981 | 0.6727 | 0.5781 | 0.5932 | 0.7400 | 0.7705 | 0.7586 |
| F                    | 0.6186 | 0.6186 | 0.5591 | 0.6602 | 0.6327 | 0.6744 | 0.6207 |
| G-mean               | 0.6408 | 0.6418 | 0.6130 | 0.6695 | 0.6518 | 0.7209 | 0.6673 |
| AUC                  | 0.4107 | 0.4119 | 0.3758 | 0.4482 | 0.4248 | 0.5196 | 0.4453 |
| margin               | 0.1078 | 0.1632 | 0.1991 | 0.2084 | 0.2056 | 0.1468 | 0.2418 |
| nTree                | 100 | 100 | 100 | 100 | 315 | 193 | 151 |
| $\kappa$             | 1  | 3  | 3  | 4  | 5  | 6  | 8  |
| **num (Attribute)**  | 17 | 17 | 17 | 17 | 13 | 16 | 15 |
| **Wine**             |    |             |                               |     |       |        |         |
| Accuracy             | 0.9846 | 0.9692 | 0.9846 | 0.9692 | 0.9846 | 0.9846 | 0.9692 |
| Sensitive            | 1.0000 | 0.9286 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| Precision            | 0.9655 | 1.0000 | 0.9677 | 0.9333 | 0.9706 | 0.9643 | 0.9355 |
| Specificity          | 0.9730 | 1.0000 | 0.9714 | 0.9459 | 0.9688 | 0.9737 | 0.9444 |
| F                    | 0.9825 | 0.9630 | 0.9836 | 0.9655 | 0.9851 | 0.9818 | 0.9667 |
| G-mean               | 0.9864 | 0.9636 | 0.9856 | 0.9726 | 0.9843 | 0.9868 | 0.9718 |
| AUC                  | 0.9730 | 0.9286 | 0.9714 | 0.9459 | 0.9688 | 0.9737 | 0.9444 |
| margin               | 0.0442 | 0.0502 | 0.0288 | 0.0748 | 0.0246 | 0.0156 | 0.0238 |
| nTree                | 100 | 100 | 100 | 100 | 349 | 354 | 354 |
| $\kappa$             | 1  | 3  | 4  | 3  | 5  | 6  | 8  |
| **num (Attribute)**  | 13 | 13 | 13 | 13 | 11 | 12 | 12 |
| **Ionosphere**       |    |             |                               |     |       |        |         |
| Accuracy             | 0.9200 | 0.9314 | 0.9371 | 0.9257 | 0.9371 | 0.9257 | 0.9314 |
| Sensitive            | 0.9107 | 0.8475 | 0.8889 | 0.8824 | 0.8333 | 0.9032 | 0.9107 |
| Precision            | 0.8500 | 0.9434 | 0.9057 | 0.9231 | 0.9804 | 0.8889 | 0.8793 |
| Specificity          | 0.9244 | 0.9741 | 0.9587 | 0.9533 | 0.9913 | 0.9381 | 0.9412 |
| F                    | 0.8793 | 0.8929 | 0.8972 | 0.9003 | 0.9009 | 0.8960 | 0.8947 |
| G-mean               | 0.9175 | 0.9086 | 0.9231 | 0.9171 | 0.9089 | 0.9205 | 0.9258 |
| AUC                  | 0.8956 | 0.8651 | 0.9002 | 0.8975 | 0.8548 | 0.8835 | 0.9029 |
| margin               | 0.1096 | 0.0860 | 0.1132 | 0.0884 | 0.0668 | 0.0831 | 0.0825 |
| nTree                | 100 | 100 | 100 | 100 | 339 | 321 | 350 |
| $\kappa$             | 1  | 3  | 4  | 3  | 5  | 6  | 2  |
| **num (Attribute)**  | 34 | 34 | 34 | 34 | 29 | 30 | 28 |
| **Breast-cancer-wisconsin** |    |             |                               |     |       |        |         |
| Accuracy             | 0.9801 | 0.9658 | 0.9715 | 0.9573 | 0.9544 | 0.9801 | 0.9658 |
| Sensitive            | 0.9914 | 0.9474 | 0.9583 | 0.9748 | 0.9919 | 1.0000 | 0.9474 |
| Precision            | 0.9504 | 0.9474 | 0.9583 | 0.9063 | 0.8905 | 0.9421 | 0.9474 |
| Specificity          | 0.9745 | 0.9747 | 0.9784 | 0.9483 | 0.9342 | 0.9705 | 0.9747 |
| F                    | 0.9701 | 0.9474 | 0.9583 | 0.9393 | 0.9385 | 0.9702 | 0.9474 |
| G-mean               | 0.9829 | 0.9609 | 0.9683 | 0.9614 | 0.9626 | 0.9851 | 0.9609 |
| AUC                  | 0.9844 | 0.9555 | 0.9595 | 0.9547 | 0.9601 | 0.9850 | 0.9474 |
| OOB                  | 0.0422 | 0.0399 | 0.0433 | 0.0467 | 0.0304 | 0.0411 | 0.0372 |
GA-RF, PSO-RF and AFSA-RF represent the hybrid algorithm.

From the Connectionist Bench results, we find that the AFSA-RF achieves the minimum OOB error and the maximum margin. The best parameter combination is (151, 4), and $\kappa$ is the same as the traditional value, $\sqrt{M}$. The features selected by AFSA-RF were [1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1], meaning that the 7th and 10th features were removed. PSO-RF obtained the best F-value, G-mean and AUC. On the wine dataset, PSO-RF achieved the minimum OOB error and the maximum G-mean and AUC scores. The best parameter combination is (354, 1), and $\kappa$ is the same as the traditional value, 1. There are 15 features selected in total. Moreover, GA-RF achieved the best OOB error, F-value and margin. For Ionosphere, we find that GA-RF achieved the best OOB error, F-value and margin. The best parameter combination is (339, 9), but the value of $\kappa$ is considerably different from the classic value. There are 29 total features selected. The best G-mean and AUC scores were obtained by AFSA-RF. For breast-cancer-wisconsin, we GA-RF achieved the best performance for OOB error and margin. The best parameter combination is (319, 3), and $\kappa$ is the same as the traditional value, $\sqrt{M}$. There are nine features selected in total. PSO-RF achieved the maximum F-value, G-mean and AUC.

The multi-classification results show that the hybrid GA-RF, PSO-RF and AFSA-RF almost always discover better features and select better parameter values than the traditional value. There are some differences between the best $\kappa$ and the traditional value. The more features there are originally, the greater the number of redundant features that are removed.

Figure 11 demonstrates that, overall, the OOB error values for all the hybrid algorithms are lower than the traditional value with fixed parameters for the six datasets. Although the traditional value is reasonable for some datasets, it fails to achieve good performance over the entire problem set. In conclusion, the hybrid algorithm effectively eliminates redundant features and

| Table 7 The binary classification results (Continued) |
|------------------------------------------------------|
| margin 0.8247 0.8569 0.8509 0.8652 0.8842 0.8179 0.8616 |
| nTree 100 100 100 100 319 420 351 |
| $\kappa$ 1 3 4 10 3 1 3 |
| num (Attribute) 10 10 10 10 9 9 7 |

The best value of every performance evaluation criteria obtained by the algorithms are marked in boldface.

| Table 8 The multi-classification results |
|-----------------------------------------|
| Steel Plates Faults Accuracy 0.7464 0.7485 0.7598 0.7814 0.7981 0.7998 0.7914 |
| OOB 0.3152 0.2819 0.2746 0.2640 0.2437 0.2276 0.2115 |
| margin 0.2456 0.3384 0.3484 0.3789 0.3803 0.3812 0.3810 |
| nTree 100 100 100 100 319 420 351 |
| $\kappa$ 1 5 5 27 8 6 6 |
| num (Attribute) 27 27 27 27 22 22 |

| Libras Movement Accuracy 0.7167 0.7556 0.6889 0.6444 0.7606 0.7767 0.7928 |
| OOB 0.3546 0.3397 0.3480 0.3163 0.3030 0.3323 0.3116 |
| margin 0.1464 0.1798 0.1990 0.2180 0.2443 0.2677 0.2910 |
| nTree 100 100 100 100 258 348 135 |
| $\kappa$ 1 5 7 90 12 8 9 |
| num (Attribute) 90 90 90 90 56 76 49 |

| mfeat-fac Accuracy 0.4280 0.9030 0.8010 0.9620 0.9673 0.9620 0.9611 |
| OOB 0.6949 0.1823 0.3192 0.0486 0.0416 0.0410 0.0361 |
| margin −0.0987 0.4561 0.2361 0.8708 0.8749 0.8615 0.8698 |
| nTree 100 100 100 100 377 270 196 |
| $\kappa$ 1 15 8 215 14 18 11 |
| num (Attribute) 215 215 215 215 145 112 164 |

The best value of every performance evaluation criteria obtained by the algorithms are marked in boldface.
obtains a suitable combination of parameters. Therefore, it enhances the classification performance of RF on imbalanced high-dimensional data.

**Conclusions**

To improve the performance of the random forests algorithm, the CURE-SMOTE algorithm is proposed for imbalanced data classification. The experiments show that the proposed algorithm effectively resolves the shortcomings of the original SMOTE algorithm for typical datasets and that various adaptive clustering techniques can be added to further improve the algorithm. We plan to continue to study the influence of feature selection and parameter settings on RF. The proposed hybrids of RF with intelligent algorithms are used to optimize RF for feature selection and parameter optimization. Simulation results show that the hybrid algorithms achieve the minimum OOB error, the best generalization ability and that their F-value, G-mean and AUC scores are generally better than those obtained using traditional values. The hybrid algorithm provides new effective guidance for feature selection and parameter optimization. The time and data dimensions of the experiments can be increased to further verify the algorithm’s effectiveness.

**Abbreviations**

AFSA: Artificial fish swarm algorithm; AFSA-RF: Artificial fish swarm-random forests algorithm; AUC: Area under the ROC curve; Can-CSC-GBE: Cost-Sensitive Classifier with a GentleBoost Ensemble; CURE: Clustering using representatives; GA: Genetic algorithm; GA-RF: Genetic-random forests; G-mean: Geometric mean; IR: Imbalance ratio; MTD: Mega-trend-diffusion; OOB: Out of bag; PSO: Particle swarm optimization; PSO-RF: Particle swarm-random forests; RBFNN: Radial basis function neural network; RF: Random forests; ROC: Receiver operating characteristics; SMOTE: Enhances the original synthetic minority oversampling technique; SVM: Support vector machine

**Acknowledgements**

The authors would like to thank the editor and the anonymous reviewers for their helpful suggestions and comments which provide a great contribution to the research of this paper, and Wenseing Ye for linguistic improvements of the paper.

**Funding**

This work is supported in part by the National Natural Science Foundation of China (Grant No. 61572233) and the National Social Science Foundation of China (Grant No. 16BTJ032).

**Availability of data and materials**

All data generated or analysed during this study are included in this published article. The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

**Authors’ contributions**

LM wrote the paper and conducted all analyses. SHF developed the paper. Both authors contributed to the design of the analyses and substantially edited the manuscript. Both authors read and approved the final manuscript.

**Competing interests**

The authors declare that they have no competing interests.

**Consent for publication**

Not applicable.

**Ethics approval and consent to participate**

Not applicable.

**Publisher’s note**

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.
57. Destrero A, Mosci S, De Mol C, et al. Feature selection for high-dimensional data. Comput Manag Sci. 2009;6(1):25–40.
58. Zhu S, Wang D, Yu K, et al. Feature selection for gene expression using model-based entropy. IEEE/ACM Trans Comput Biol Bioinform. 2010;7(1):25–36.
59. Kausar N, Majid A. Random forest-based scheme using feature and decision levels information for multi-focus image fusion. Pattern Anal Applic. 2016;19(1):221–36.
60. Menze BH, et al. A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data. BMC bioinformatics. 2009;10(1):213.
61. Strobl C, et al. Bias in random forest variable importance measures: Illustrations, sources and a solution. BMC bioinformatics. 2007;8(1):1.
62. Zhou Q, Zhou H, Li T. Cost-sensitive feature selection using random forest: Selecting low-cost subsets of informative features. Knowl-Based Syst. 2016;95:1–11.
63. Díaz-Uriarte R, De Andres SA. Gene selection and classification of microarray data using random forest. BMC bioinformatics. 2006;7(1):1.
64. Lariviere B, Van den Poel D. Predicting customer retention and profitability by using random forests and regression forests techniques. Expert Systems with Applications. 2005;29:472–84.
65. Rodriguez-Galliano VP, Ghimire B, Rogan J, Chica-Olmo M, Rigol-Sanchez JP. An assessment of the effectiveness of a random forest classifier for landcover classification. ISPRS J Photogramm Remote Sens. 2012;67:93–104.
66. Bernard S, Heutte L, Adam S. Influence of Hyper parameters on Random Forest Accuracy [C]/Proceedings of the 8th International workshop on multiple classifier systems, Berlin, Heidelberg: Springer; 2009. p. 171–80.
67. Yu L, Chun-xia Z. Estimation of the hyper-parameter in random forest based on out-of-bag sample. J Syst Eng. 2011;26(4):566–72.
68. Martinez-Munoz G, Suarez A. Out-of-bag estimation of the optimal sample size in bagging. Pattern Recogn. 2010;43(1):143–52.
69. Ming-yuan Z, Yong T, Chong F, Ming-tian Z. Feature selection and parameter optimization for SVM based on genetic algorithm with feature chromosomes. Control and Decision. 2010;25(8):1133–8.
70. Lei L, Gao L, Shijie Z. Application of artificial fish-swarm algorithm in SVM parameter optimization selection. Comput Eng Appl. 2013;49(23):86–90.
71. Xiu-jiang SHAO, Hui-zhong YANG, Gang CHEN. Parameters selection and application of support vector machines based on particle swarm optimization algorithm. Control Theory & Applications. 2006;23(3):740–4.
72. Chen J, Tang YY, Fang B, et al. In silico prediction of toxic action mechanisms of phenols for imbalanced data with Random Forest learner. J Mol Graph Model. 2012;35:21–7.
73. Espindola R P, Ebecken N F F. On extending f-measure and g-mean metrics to multi-class problems [C]/Sixth international conference on data mining, text mining and their business applications, Wessex Institute of Technology, UK, 2005. 35: 25–34.
74. Janitzka S, Strobl C, Boulesteix AL. An AUC-based permutation variable importance measure for random forests. BMC bioinformatics. 2013;14(1):119.
75. Breiman L. Out-of-bag Estimation [R]. Berkeley: Statistics Department, University of California; 1996.
76. Mellor A, Boukri S, Haywood A, et al. Exploring issues of training data imbalance and mislabeling on random forest performance for large area land cover classification using the ensemble margin. ISPRS J Photogramm Remote Sens. 2015;105:155–68.