IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 30, 2019, accepted June 13, 2019, date of publication June 19, 2019, date of current version July 8, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2923846

A Classification Method Based on Feature Selection for Imbalanced Data

YI LIU[1,2], YANZHEN WANG[1,2,3], XIAOGUANG REN[1,2], HAO ZHOU[1,2], AND XINGCHUN DIAO[1,2]

[1]National Innovation Institute of Defense Technology, Beijing 100010, China
[2]Tianjin Artificial Intelligence Innovation Center, Tianjin 300457, China
[3]State Key Laboratory of High Performance Computing, National University of Defense Technology, Changsha 410073, China

Corresponding author: Xiaoguang Ren (renxiaoguang@nudt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 91648204 and Grant 61532007, in part by the National Key Research and Development Program of China under Grant 2017YFB1001900 and Grant 2017YFB1301104, in part by the National Science Foundation for Young Scientists of China under Grant 61802426, and in part by the National Science and Technology Major Project.

ABSTRACT Imbalanced data are very common in the real world, and it may deteriorate the performance of the conventional classification algorithms. In order to resolve the imbalanced classification problems, we propose an ensemble classification method that combines evolutionary under-sampling and feature selection. We employ the Bootstrap method in original data to generate many sample subsets. V-statistic is developed to measure the distribution of imbalanced data, and it is also taken as the optimization objective of the genetic algorithm for the under-sampling sample subsets. Moreover, we take F_1 and *Gmean* indicators as two optimization objectives and employ the multiobjective ant colony optimization algorithm for feature selection of resampled data to construct an ensemble system. Ten low-dimensional and four high-dimensional typical imbalanced datasets are used in experiments. The six state-of-the-art algorithms and four measures are taken for a fair comparison. The experimental results show that our proposed system has a better classification performance compared with other algorithms, especially for the high-dimensional imbalanced data.

INDEX TERMS Feature selection, imbalanced data, multiobjective ant colony optimization, genetic algorithm.

I. INTRODUCTION

With the development of information technology and industry applications, the volume of data is increasing rapidly. It is a popular trend that adopting machine learning, artificial intelligence and deep learning to get latent information from data for providing users with more smart service [1]–[3]. Traditional classification algorithms take the assumption that data has a good distribution; however, it is common that training data are imbalanced over classes, which leads to the bias of learning algorithms. The research on imbalanced classification has recently drawn much attention [4]–[6].

Taking binary classification as an example, the imbalanced problem means that the instances of one class are more than another, and the class of major samples is the major class and another is the minor class. We usually pay more attention to the classification performance of minor class. The problem of imbalanced distribution is widely existing in real world applications, such as fraud detection, cancer diagnosing, network intrusion detection, and entity resolution [7], [8]. Imbalanced problems can be divided into two types, i.e. relative imbalance and absolute imbalance. Relative imbalance means that the ratio of minor instances to major instances is less than one, but the number of minor instances may be also large, such as 1000 or more. Absolute imbalance denotes there are very few samples in minority class such as 10 or less in dataset. We can classify the causes of imbalance problem as intrinsic and extrinsic. The intrinsic reason indicates the inherent property of data. For example, the probability that an equipment fails is much lower than it runs normally. And the number of people having cancer is obviously less than that of healthy people. Extrinsic reason means other factors are leading to the imbalance of data. For example, sporadic interruptions occur when the balanced data is transmitting to the database [9].

There are some effective methods proposed to adapt traditional algorithms to imbalanced data, including data-level

The associate editor coordinating the review of this manuscript and approving it for publication was Arif Ur Rahman.

methods, algorithm-level methods and ensemble learning methods. Data-level methods employ sampling technologies to rebalance imbalanced data, including under-sampling, and over-sampling. Algorithm-level methods construct new algorithms or modify traditional algorithms to reduce the disadvantages of imbalanced data, including cost-sensitive learning, one class learning and feature methods. Ensemble learning methods combine ensemble learning with data-level or algorithm-level methods to further improve their performance.

Evolutionary under-sampling is an important data-level method, which selects the sample subset that maximizes or minimizes the predefined objective functions to eliminate the effect of imbalanced data [10]. Meanwhile, researchers also use feature selection to choose relevant variables to eliminate the disadvantages of imbalanced distribution and promote the performance of classifiers [11]. Additionally, integrating ensemble learning method into algorithms can further improve their classification and robust performance [12].

Previous methods use one or two advantages of basic algorithms. However, in this paper, we propose a more powerful imbalanced classification method called Genetic Under-sampling and Multiobjectie Ant Colony Optimization based Feature selection (GU-MOACOFS) which combines ensemble learning, evolutionary under-sampling and feature selection simultaneously. We employ Bootstrap to sample original data. When the dimension of data is high, we use symmetrical uncertainty (SU) to implement feature selection to reduce computation costs. Then we develop a new indicator called V-statistic to measure the distribution of data, which is adopted as an optimization objective of genetic algorithm for under-sampling without classifier. After that, we implement feature selection on sample subsets by multi-objective ant colony optimization to get training subsets as inputs of classifiers. GU-MOACOFS takes advantage of ensemble learning, under-sampling and feature selection at the same time. Exhaust experiments show its superiority compared with other state-of-art algorithms.

This paper is organized as follows. Section 2 reviews the related works of imbalanced classification methods. Section 3 describes our method in detail. Section 4 presents the results of the experiments based on 14 classical data sets. Conclusion is given in section 5.

II. RELATED WORKS

A. DATA-LEVEL METHODS

Under-sampling methods are to balance data by selecting some majority class instances and combining them with all minority class instances. Random under-sampling (RUS) is one of the most popular under-sampling methods [13]. It is obvious that under-sampling is a combinatorial problem, so the under-sampling methods based on evolutionary algorithms are widely used in real applications [14].

Yu *et al*. propose an under-sampling method based on ant colony optimization [15]. It encodes instances of majority class as edges traveled by ants and uses 0 or 1 to denote whether the current edge is selected or not. In order to avoid over-fitting, it divides original data into three groups randomly; two of them are training sample datasets and the rest is testing sample dataset. Then it selects instances from the majority class and repeats this process for 100 times to ensure every sample can be chosen as one of the training samples at least once. At last, the 100 results are united, and the algorithm selects majority class instances based on their frequencies.

Krawczyk *et al*. develop an algorithm based on Boosting and regard genetic algorithm under-sampling as its component, which is different from adopting an evolutionary algorithm under-sampling directly [16]. Boosting is an ensemble learning method, and diversity is important for its classifiers. Hence, they propose an indicator to measure classification performance and diversity among classifiers at the same time, and apply it to guide genetic algorithm to select samples maximizing classifiers' classification performance and diversity simultaneously.

In order to balance big imbalanced data, Triguero *et al*. combine genetic algorithm under-sampling with MapReduce framework [17]. They split original data into M partitions at Mapper stage, use genetic algorithm to implement under-sampling on every partition to balance samples to train classifiers and integrate results at Reduce stage.

It is obvious that the evolutionary algorithm's under-sampling is straight and deployed easily, and it has attracted attention and been widely used. However, the most disadvantage of evolutionary algorithm under-sampling is time-consuming. Besides, traditional RUS may lose important information.

Over-sampling methods reduce the effects of imbalanced data by adopting over-sampling or generating new minority instances. There are two popular methods, i.e. random over-sampling (ROS) and synthetic minority over sampling technique (SMOTE), including its variants [18]–[20].

SMOTE is a very famous over-sampling method [19]. It selects one instance of minority class and gets its k nearest neighbors by Euclidean metric based on their features' space. Then it chooses one instance from its k nearest neighbors randomly, and generates new minority class instance by the difference between itself and its selected neighbors.

Adaptive synthetic sampling (ADASYN) is an excellent algorithm based on SMOTE. It tunes the instances of minority class generated by SMOTE according to the probability distribution of minority class to improve the classification accuracy of classifiers. In detail, it yields fewer instances if the distribution of minority class is simple and more instances if the boundary of the minority class is complexity [21].

Other variants of SMOTE select more reasonable instances generally through some specific approaches. Ramentol *et al*. use fuzzy rough set to measure the degrees of newly generated instances and original instances, and remove samples whose degrees are lower than predefined value [22]. It is a key that they use two different predefined values, the one is lower for

majority class instances in order to preserve original information, and another is higher for newly generated instances in order to save more suitable data.

ROS is simple and direct for users, but it is over-fitting easily. SMOTE and its variants can handle the disadvantage of ROS effectively, however many real-world applications rely on real data such as medical diagnose and fault detection etc., so it is not possible to adopt those methods [23], [24].

B. ALGORITHM-LEVEL METHODS

Unlike under-sampling and over-sampling methods, cost-sensitive methods do not change data distribution. They construct cost matrices to assign higher costs for the misclassification of minority class instances with respect to majority class instances. There are three categories of cost-sensitive methods, i.e. methods based on translation theorem, methods based on meta cost framework and methods directly design appropriate cost functions for specific classifier. Though it is superior to the data-level method, it is difficult to predefine an appropriate cost function when facing complex imbalanced problem [25], [26].

One class learning methods also do not change data distribution. They obtain similarity degree values between instances by their features and classify each instance on the basis of predefined similarity thresholds. One class learning methods have better predictive performance and could resolve over-fitting in part. However, their accuracy dependent on the similarity thresholds which need to be empirically tuned to achieve the desired performance [27], [28]

Feature methods include feature selection and feature extraction. Classifiers may regard instances of minority class as outliers or noise data as features of imbalanced data may have a bias towards majority class. Feature selection method can shift the focus on the features optimizing the contrast between classes rather than the training examples.

Yin *et al*. propose a feature selection method based on decomposition for the binary imbalanced problem [29]. They cluster training data into C virtual classes, measure feature's correlation based on its relationship with class labels, and select former N best features based on their evaluation values.

It attracts much attention that implementing feature selection for resolving the imbalanced problem. Moayedikia *et al*. apply feature selection based on harmony search to eliminate the influence of high dimensional imbalanced data [30]. They measure the feature's correlation by SU firstly, and adopt harmony search to implement feature selection. Besides, they introduce a vector tuning operation to add or remove features from feature subset according to their values obtained by SU. Du *et al*. employ a genetic algorithm to implement feature selection on multiclass imbalanced data to improve classifier's performance [31]. Moreover, they combine the extension of geometric mean and the ratio between the number of selected features and the number of original features as optimization objectives. Besides, Fernandez *et al*. integrate sampling with feature selection, encode training samples w and features n into chromosome whose length is $|w|+|n|$ [32]. They take the area under the ROC curve and the difference between the number of selected samples and original samples as two optimization objectives simultaneously, and non-dominated sorting genetic algorithm ć̀ò is selected to realize their method.

It is a new research direction that using feature selection to eliminate the drawback of imbalanced data, and it is becoming a hot point though it has a short history.

Feature selection chooses features from original data, and it does not change original features. Feature extraction, which is different from feature selection, transforms data into a low-dimensional space based on original features information by some methods. Feature extraction contains some technologies such as singular value decomposition, principal component analysis and non-negative matrix factorization etc. [33], [34]. The disadvantage of feature extraction is the difficulty to interpret new generated features, and some technologies are time-consuming, such as non-negative matrix factorization.

C. ENSEMBLE LEARNING METHODS

Ensemble learning methods combine ensemble learning technologies with data-level methods or algorithm-level methods to improve algorithms' classification performance. Some popular ensemble learning methods include Bagging, Boosting and Adaboost. Bagging is an inherent parallel ensemble learning technology whose components can be running at the same time, and uses majority voting or weighted majority voting to aggregate results. Boosting and Adaboost are iterative ensemble learning methods, and they train a classifier every iteration and focus on the misclassification samples classified at the next iteration to promote whole classifiers' performance [35].

There are some ways that combine ensemble learning with data-level algorithms. Sun *et al*. propose a multiple classifier system based on Bagging [36]. They develop samples balancing methods to obtain different balanced data sets, and use those to construct multiple classifier systems to get a high-performance classifier. They introduce two samples balancing methods, one uses a clustering algorithm to gather instances of majority class into some clusters and combines them with all minority class samples to generate many balanced data sets, and another splits majority class samples according to the number of minority class instances in order to ensure the number of majority class instances of new generated data sets is the same with that of minority class instances.

SMOTEBoost is a famous modified algorithm based on Adaboost, it integrates SMOTE with Adaboost, and uses SMOTE to construct a new balanced data set to train classifier at each time in Adaboost to increase misclassification samples' weights in next iteration [37].

The other way is to union ensemble learning technologies with algorithm-level methods. Guo *et al*. adopt Adaboost and feature selection on the basis of binary particle swarm optimization to resolve the multiclass imbalanced classification

problem [38]. They regard Adaboost as a component of binary particle swarm optimization which implements feature selection, and employ Adaboost to construct multiple classifier system by new training samples.

Li *et al*. develop a self-adapted ensemble classification algorithm which uses ensemble learning, data-level methods and algorithm-level methods simultaneously to solve multiclass imbalanced classification problem specifically [39]. They apply feature selection on original data, and take resampling strategy based on ensemble learning to construct multiple classifier systems. Then they use some alternate methods to realize the proposed algorithm, such as employing Fast Correlation Based Filter or binary particle swarm optimization to realize feature selection, and applying Adaboost, under-sampling balanced ensemble, or over-sampling balanced ensemble to realize resampling.

Except for those mentioned algorithms, RUSBoost, EUSBoost, EasyEnsemble and BalanceCascade etc. have been widely used in resolving imbalanced classification problems [40]–[42]. The applications of ensemble learning technologies can improve original algorithms' adaptability and robustness, and they have become a hot direction of imbalanced classification researches. However, ensemble learning methods have an embarrassing drawback, i.e. they are very time-consuming, especially Boosting and Adaboost. Besides, the development of big data has made high dimensional data become the main data type, which leads to a difficult situation for current ensemble learning methods.

III. GU-MOACOFS

A. FRAMEWORK OF GU-MOACOFS

The framework of the proposed GU-MOACOFS is described in Fig. 1.

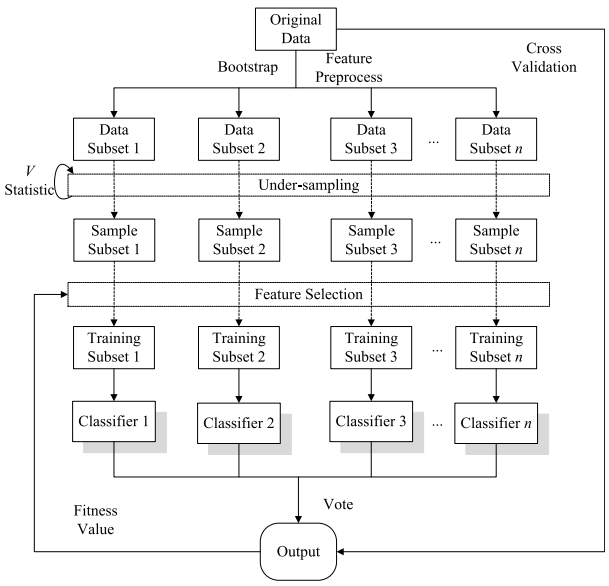


FIGURE 1. Framework of GU-MOACOFS.

Now we explain the reasons why GU-MOACOFS looks like that.

- There are some general ensemble learning frameworks such as Bagging, Boosting, and Adaboost, etc. We choose Bagging because of the following factors. First, Bagging has a better performance than other frameworks under same time [43]. Second, Bagging is an inherent parallel framework which is easy to be deployed. Besides, it has less time costs than other iterative methods, and it is more popular than iterative based ensembles in applications [44].
- Any sampling method has its own principle. For example, RUS implements random under-sampling until the number of instances of majority class is the same as that of minority class. We propose V-statistic to measure the distribution of under-sampling data by genetic algorithm, and it can improve the efficiency of sampling compared with other evolutionary under-sampling approaches combining with classifiers. Besides, GU-MOACOFS applies SU to abandon noise and useless features to reduce time costs when data dimensionality is very high.
- Multiobjective ant colony optimization is a discrete optimization algorithm which is more fit for feature selection compared with continuous optimization algorithms such as genetic algorithm or particle swarm optimization. There is more than one objective that we want to optimize in imbalanced classification problems, and they may conflict with each other. It is better to use multiobjective optimization technology to get better solutions than single objective optimization methods. So GU-MOACOFS employs multiobjective ant colony optimization to implement feature selection.
- GU-MOACOFS applies sampling first and then feature selection, so multiobjective ant colony optimization can try its best to make use of distribution information in different sampling subsets.

B. V-STATISTIC

Some indicators are used to measure the complexity of imbalanced data, and the most popular indicator is imbalanced ratio (IR) which is defined as the ratio between the number of majority class samples and the number of minority class samples, and it is shown in eq. (1).

$$IR = \frac{N_{\text{major}}}{N_{\text{minor}}} \tag{1}$$

where N_{major} denotes the number of majority class samples, and N_{minor} denotes the number of minority class samples.

But it is not enough to use IR to measure the complexity and distribution of imbalanced data. Fig. 2 shows two different distributions under the same IR.

We can find that though they have the same IR, the distribution of subfigure (a) is simpler than subfigure (b) which has a worse boundary. Moreover, the classifier trained by samples

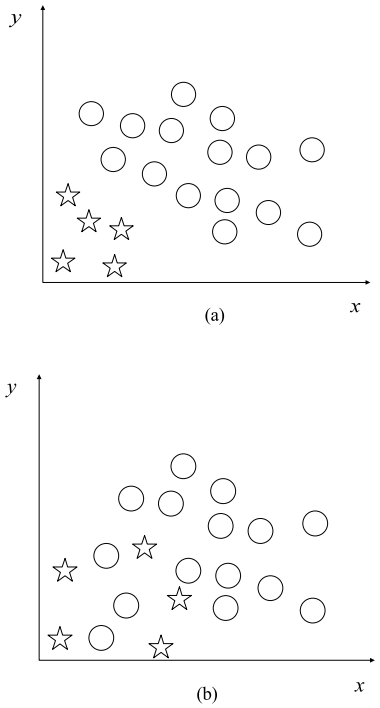


FIGURE 2. Two different distributions under the same IR. (a) Simple distribution situation. (b) Complex distribution situation.

in subfigure (a) will have a better performance. However, it cannot be measured only by IR.

Ho and Base summarize 12 indications that measure the complexity of imbalanced data, and they divide them into three categories, i.e. measures of overlap individual feature values, measures of separability of classes and measures of geometry, topology and density of manifolds [45]. Luengo *et al*. make some comparison experiments between those 12 measures, and they conclude that maximum Fisher's discriminant has a better ability [14]. The maximum Fisher's discriminant is defined as eq. (2).

$$fd(i) = \frac{(\mu_{i1} - \mu_{i2})^2}{\delta_{i1}^2 + \delta_{i2}^2}$$
$$MF = \max(fd) \tag{2}$$

where $\mu_{i1}, \mu_{i2}, \delta_{i1}, \delta_{i2}$ are the means and variances of the two classes. $fd(i)$ denotes the fisher's discriminant value of feature i, and MF is the maximum fd over all the features.

In order to reflect the distribution of imbalanced data comprehensively, we combine those two measures to propose V-statistic which is defined to be eq. (3).

$$V = MF/IR \tag{3}$$

It means the distribution of data is better if the value of V-statistic is higher.

C. GENETIC ALGORITHM BASED UNDER-SAMPLING

It is an important method to use an evolutionary algorithm to resample data. In traditional researches, it often combines with classifiers and adopts classification performance as optimization objectives. That way has some disadvantages which we want to solve. First, its results have a strong correlation with classifiers, and it does not make use of samples' information. Second, it may be time-consuming because of using classifiers. GU-MOACOFS performs genetic algorithm-based under-sampling (GAUS) which is independent of classifier and uses V-statistic as its optimization objective to resolve conventional algorithms' disadvantages and improve the efficiency of resampling.

Algorithm 1 shows how GAUS works.

Algorithm 1 Pseudo Code of GAUS

01. Input: Imbalanced Data
02. Output: Rebalanced Data
03. BEGIN
04. Number the instances of majority class, transform them into chromosomes, and initialize chromosomes
05. WHILE (Not meet stopping criteria)
06. Crossover operation between chromosomes
07. Mutation operation among new chromosomes
08. Union selected instances of majority class with all instances of minority class
09. Calculate fitness values of chromosomes by V-statistic
10. Choose elite chromosomes based on their fitness values
11. END WHILE
12. END

There are many crossover operations in genetic algorithms, such as single-point crossover, multiple-point crossover and uniform crossover etc. We use single-point crossover familiarly in this paper.

D. MULTIOBJECTIVE ANT COLONY OPTIMIZATION

It is a very common preprocessing way to apply feature selection to find relevant features in order to improve model's interpretation, reduce time-consuming and storage of data, and improve classification performance. Fig. 3 illustrates this observation. We randomly select a data set and plot its sample distribution on three dimensions space, and then remove the noise feature. We can find that it is easier to detect boundary after removing noise feature. Though conventional feature selection algorithms work well in balanced data, they may deteriorate in imbalanced data and resulting in a higher error rate [29]. In order to utilize the advantage of feature selection and avoid the effect of imbalanced data, we employ multiobjective ant colony optimization to implement feature selection.

Feature selection is a subset problem in math, and multi-objective ant colony optimization has a better performance in resolving subset problems. Cao *et al*. propose a graph-based ant system which constructs a structure graph composed of a directed graph and some mappings which map the problems onto directed graphs [46]. Then they apply a method called equivalent routes pheromone strengthening policy to update the pheromone matrix. However, it is only used

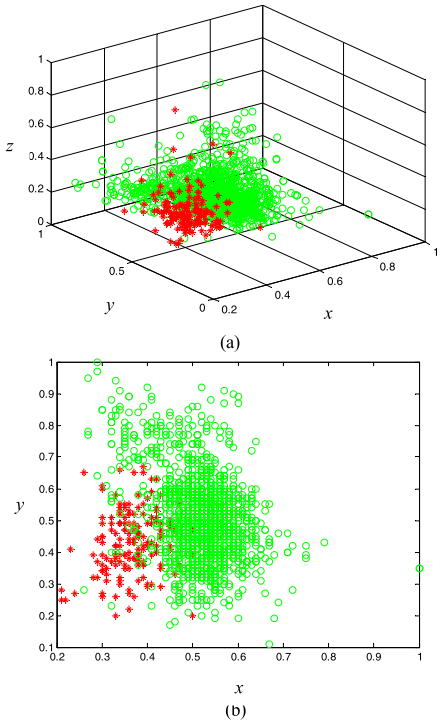


FIGURE 3. Sample distribution before and after removing noise feature. (a) Before feature selection. (b) After feature selection.

Algorithm 2 Pseudo Code of Feature Selection by Multiobjective Ant Colony Optimization

01. Input: Data sets
02. Output: Pareto solutions (feature subsets)
03. BEGIN
04. Initialize Pareto archive, pheromone matrices and heuristic information matrix
05.WHILE (Not meet stopping criteria)
06. FOR each ant
07. Construct solution based on transition probability function and heuristic information
08. Reconstruct training data sets based on selected features and train classifiers to obtain fitness values
09. END FOR
10. Update Pareto archive according to fitness values of solutions
11. Choose solutions from Pareto archive to update pheromone matrices
12. END WHILE
13. END

for single objective optimization problems. Based on reference [46], we make it being fit for multiobjective optimization problems. Then we employ it to choose feature subsets.

Algorithm 2 shows the feature selection process of the modified multiobjective ant colony optimization.

In step 7, the transition probability is given by eq. (4).

$$p_{ij}^{k} = \begin{cases} \dfrac{[\tau_{ij}]^{\alpha} \cdot [\eta_i]^{\beta}}{\sum_{e_{bj} \notin visit_k} [\tau_{bj}]^{\alpha} \cdot [\eta_b]^{\beta}} & e_{ij} \notin visit_k \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where p_{ij}^{k} denotes the probability that ant k travels from node d_j to node d_{j+1} by edge e_{ij}. τ_{ij} is the pheromone value of edge e_{ij} at current iteration, η_i a statistic expectation heuristic information of selecting node i. $visit_k$ denotes the edges which are visited by ant k. αandβ are constants to control the relative importance of the pheromone versus the heuristic information.

We use Fisher discriminant as the heuristic information of multiobjective ant colony optimization. As GU-MOACOFS has n sample subsets because of using Bagging method, we calculate Fisher discriminant value of each feature in every sample subset, then aggregate them to generate heuristic information, which is given by eq. (5).

$$sf(i) = \sum_{r=1}^{nf} f_r(i) \tag{5}$$

In order to get better quality of Pareto solutions, we set more than one pheromone matrix. And weight product method is applied to aggregate values of pheromone matrices to calculate transition probability. Supposing there are two optimization objectives, Eq. (6) shows us how to obtain transition probability.

$$\tau_{ij} = (\tau_{ij}^{1})^{(1-\lambda)} \cdot (\tau_{ij}^{2})^{\lambda} \tag{6}$$

where λ is a weight parameter and its range is [0, 1].

After obtaining new Pareto archive, pheromone matrices must be updated for further evolution by solutions from Pareto archive. The way to update edge $tabu^t$ at time t is described in eq. (7).

$$\tau_{ij}(t) = \begin{cases} (1-\rho)\tau_{ij}(t-1) + \Delta'\left(tabu^t\right) & e_{ij} \in \Psi\left(tabu^t\right) \\ (1-\rho)\tau_{ij}(t-1) & otherwise \end{cases} \tag{7}$$

where ρ is evaporation rate, $\Psi\left(tabu^t\right)$ is the equivalent routes of edges $tabu^t$, $\Delta'\left(tabu^t\right)$ is the increasing pheromone value [46]. We use fitness values of selected solutions to update pheromone matrices to improve the algorithm's ability to find better results. $\Delta'\left(tabu^t\right)$ is calculated by eq. (8).

$$\Delta'\left(tabu^t\right) = \left(\sum_{h=1}^{m} f_h(tabu^t)\right)/(Q * m) \tag{8}$$

where m is the number of objectives, $f_h(tabu^t)$ denotes the fitness value of edge in hth objective, and Q is a constant value.

E. PREPROCESSING AND DESCRIPTION OF PROPOSED METHOD

In this part, we describe the preprocessing steps of the proposed algorithm when the dimension of data is high. Then we give the pseudo code of GU-MOACOFS and its complexity.

Our method may have a shortcoming when the dimension of data is very high. As we apply V-statistic as an optimization objective, it is very time-consuming when dimensionality is high. So we employ SU which is an indicator to measure feature correlations based on entropy. SU is effective in feature selection for large scale data sets [47]. Supposing x and y are two variables, and their uncertain is given by eqs. (9) and (10).

$$H(x) = -\sum P(x_i)\log_2(P(x_i)) \tag{9}$$

$$H(x|y) = -\sum_i P(y_i)\sum_j P(x_i|y_j)\log_2(P(x_i|y_j)) \tag{10}$$

where $P(x_i)$ is the prior probability over all values of x, and $P(x_i|y_j)$ is the posterior probability of x. Based on eqs. (9) and (10), we can get information gain defined in eq. (11).

$$G(x|y) = H(x) - H(x|y) \tag{11}$$

it gives the entropy loss of x when y is considered. Apparently, x contains more information if G is higher. To compare each combination of x, y meaningfully, the values in eq. (11) have to be normalized as eq. (12).

$$S(x, y) = \frac{2G(x|y)}{H(x) + H(y)} \tag{12}$$

The range of S is between 0 and 1. $S = 1$ means that x and y are fully correlated, while $S = 0$ implies that x and y are independent.

If there are many dimensions, we can apply SU to measure the correlations of features to abandon noise and redundant features firstly. By this way, we can reduce the time costs of genetic algorithm and multiobjective ant colony optimization at the same time. As it is clear that it will cost more time to select 20 features from 2000 features than from 100 features. Eq. (12) can only give the SU of one feature against a single target class label. We use eq. (13) to measure the weight of feature f_i over two class labels.

$$FS(f_i, c) = \frac{S(f_i|c)}{\sum_j S(f_j|c)} \tag{13}$$

where $\forall j,\ i \neq j$. If feature f_i is strongly correlated with class c_i, then its FS will have the greatest value for all $S(f_j|c)$.

Now, we can give the pseudo code of GU-MOACOFS in algorithm 3.

Now, we analyze the complexity of GU-MOACOFS. Supposing the number of data subsets is n, original data dimension is C, GAUS iteration number is ite_g, GAUS population size is N_g, MOACO iteration number is ite_m, MOACO population size is N_m. The time of Bootstrap sampling is $O(n)$, the overall complexity of crossover and mutation operation of GAUS is $O(N\_g)$, the evaluation time of GAUS for each solution by V-statistic is $O(C)$, so the overall time of GAUS is $O(n{\times}ite\_g{\times}N\_g{\times}C)$. In MOACO, the most time-consuming part is searching for solutions by ants and its complexity is $O(ite\_m{\times}N\_m{\times}C^2)$. The overall time of GU-MOACOFS is $O(n{\times}ite\_g{\times}N\_g{\times}C + ite\_m{\times}N\_m{\times}C^2)$.

Algorithm 3 Pseudo Code of GU-MOACOFS

01. BEGIN
02. Initialize parameters and set maximum iteration value *iter*
03. IF dimension of original data is higher than predefined value *nf*
04. Obtain every feature's correlation by eq. (13), sort them in descending order, and select former *nf* features to construct new input data
05. ENF IF
06. Resample data by Bootstrap to generate n groups of data subsets
07. FOR each data subsets
08. Adopt algorithm 1 to implement under-sampling to generate corresponding sample subsets
09. END FOR
10. Employ algorithm 2 to get feature subsets
11. Select one feature subset based on user's preference
12. END

IV. EXPETIMENTS AND ANALYSIS

A. DATASETS AND MEASURES

In this part, we use fourteen data sets which include ten low dimensional data sets and four high dimensional data sets to make comparison experiments. Four low dimensional data sets come from website http://www.keel.es/datasets.php, three high dimensional data sets, i.e. DLBCL, CNS and COLON, come from reference [30], and the last dimensional data set comes from reference [15]. Table 1 gives the characteristics of fourteen data sets, and the last column denotes the number of features which are selected by our algorithm and compared algorithms. We have mentioned that we will use SU to implement feature selection preprocessing steps in order to

TABLE 1. Characteristics of experiment datasets.

Name	Instances	Features	IR	Selected Features
GLASS4	214	9	15.4615	**5**
ECOLI01VS5	240	6	11.0000	**4**
ECOLI067VS35	222	7	9.0909	**4**
ECOLI0146VS5	280	6	13.0000	**4**
YEAST2VS8	482	8	23.1000	**5**
ECOLI0347VS56	257	7	9.2800	**5**
VEHICLE0	846	18	3.2513	**15**
ECOLI01VS235	244	7	9.1667	**5**
YEAST05679VS4	528	8	9.3529	**5**
YEAST4	1484	8	11.0000	**4**
DLBCL	59	7129	1.5000	**14**
CENTRAL NERVOUS(CNS)	60	7129	1.8571	**15**
GLI85	85	22283	2.2692	**10**
COLON	62	2000	1.8182	**20**

improve our method's efficiency, and we set the number of selected features by SU as 100 for all high dimensional data sets.

Indicator is very important for measuring the performance of algorithms in imbalanced data. For balanced binary classification data, we often use the classification success rate (also known as overall accuracy) which is the total classification accuracy of two classes. But it is not enough for imbalanced problems. Supposing there are 90 negative samples and 10 positive samples, a classifier's classification success rate will be 90% though it misclassifies all positive samples. So we must employ some more effective indicators. Table 2 gives the confusion matrix for two-class classification problem, and the positive class is minority class and negative class is majority class.

TABLE 2. Confusion matrix for two-class problem.

		Predicted	
		Positive	Negtive
True	Positive	TP(True Positive)	FN (False Negative)
	Negtive	FP (False Positive)	TN (True Negative)

TP (FP) is the number of positive (negative) instances classified correctly, and FN (TN) is the number of positive (negative) instances classified wrongly.

As we usually care about positive instances classification results, we can use F indicator which is given as eq. (14) to measure classifier's performance.

$$F_\beta = \frac{(1+\beta^2)\cdot TP_{rate}\cdot PP_{value}}{\beta^2\cdot PP_{value}+TP_{rate}} \tag{14}$$

where $TP_{rate} = \frac{TP}{TP+FN}$, $PP_{value} = \frac{TP}{TP+FP}$, TP_{rate} is also called recall, and PP_{value} is usually called precision. F indicator is a comprehensive measure for recall and precision, and β is a coefficient which is used to adjust the important degree of recall compared with precision. Usually, we set $\beta = 1$ (also known as F_1 indicator) and it means that recall and precision are important equally.

The area under ROC curve (*AUC*) is also a popular indicator used to measure the classification performance in imbalanced data. And it can be obtained by eq. (15).

$$AUC = \frac{1+TP_{rate}-FP_{rate}}{2} \tag{15}$$

where $FP_{rate} = \frac{FP}{TN+FP}$. The range of *AUC* is [0.5, 1], and the classifier is better if *AUC* is higher. If *AUC* value is equal to 0.5, it means the classifier is a random classifier.

Besides, we often adopt geometric mean (*Gmean*) to get classifier's performance, and it is given by eq. (16).

$$Gmean = \sqrt{\frac{TP}{TP+FN}\times\frac{TN}{TN+FP}} \tag{16}$$

Gmean is the geometric mean of the accuracies of two classes, and it attempts to maximize them while obtaining good balance.

At last, we will still use classification success rate to evaluate the results as it can give us the overall performance evaluation of classifier in a sort of way. It is computed as eq. (17).

$$Acc = \frac{TP+TN}{TP+FN+TN+FP} \tag{17}$$

B. VALIDATION OF GENETIC ALGORITHM UNDER-SAMPLING

First, we make some tests to evaluate the performance of the proposed genetic algorithm based under-sampling (GAUS). We use ROS, RUS and SMOTE as compared algorithms. The parameters of GAUS are set as follows, maximum number of iterations ite_g =200, population size N_g =80, crossover percentage is 0.7, mutation percentage is 0.3, and mutation rate is 0.25. We implement twenty times fivefold cross tests. In order to make a fair comparison between algorithms, we employ feature selection by SU before running each algorithm in four high dimensional data sets. Table 3 to table 6 give the F_1, *Gmean*, *AUC* and *Acc* results of four compared algorithms.

We analyze the performance of four compared algorithms from low to high dimensional data aspects. We count the number of best results provided by four algorithms in ten low dimensional data sets firstly. From table 3, we can find that GAUS, SMOTE and RUS get seven, four and one best values separately. GAUS obtains seven best results, SMOTE provides three highest values, and ROS and RUS obtain one maximum separately in table 4. In *AUC* experiments, GAUS performs better in eight data sets, ROS is better on GLASS4 and RUS is the best in YEAST2VS8. In *Acc* results, GAUS and SMOTE perform better in six data sets simultaneously although RUS is better than others in YEAST2VS8. From statistic results, we can find that random sampling methods (ROS and RUS) are worse than sampling methods based on criteria (GAUS and SMOTE) as random algorithms are uncertain inherently. Besides, it is indicated that RUS is better than ROS though ROS is over fitting easily and RUS may lose information. And it also means under-sampling is more effective than over-sampling in a way. At last, it is clear that GAUS performs better than SMOTE in most cases as SMOTE only uses the distance information of training samples, while GAUS adopts V-statistic as criterion which considers feature's classification ability and sample's quantity at the same time. So GAUS can provide sampling subsets which have a better classification performance.

Now we take a glance at the results in four high dimensional data sets. GAUS provides all four best values while the precision and recall of ROS and RUS are all zeros. It means that they cannot find minority class samples. In table 4, GAUS get the best results in GLI85 and COLON, SMOTE is better in DLBCL, and RUS performs better in CNS. In *AUC* results, GAUS has a better performance in three data sets except CNS whose highest value is provided by RUS. Besides, the *AUC* of ROS is 0.5 in GLI85, and it indicates that the performance of ROS is equal to that of the random classifier which does

TABLE 3. F_1 results of four compared algorithms.

Data sets	ROS	RUS	SMOTE	GAUS
GLASS4	0.5661±0.2017	0.4949±0.1757	**0.5926±0.1732**	0.5558±0.1141
ECOLI01VS5	0.6314±0.1943	0.6181±0.1980	0.6389±0.1992	**0.6717±0.1475**
ECOLI067VS35	0.6295±0.0831	0.4307±0.1398	0.6587±0.1599	**0.6933±0.1321**
ECOLI0146VS5	0.5256±0.0631	0.4797±0.0559	0.7177±0.2037	**0.7267±0.1862**
YEAST2VS8	0.5077±0.2104	**0.5825±0.2378**	**0.5825±0.2378**	**0.5825±0.2378**
ECOLI0347VS56	0.7302±0.1025	0.6561±0.2537	0.6733±0.1324	**0.8592±0.0950**
VEHICLE0	0.9201±0.0215	0.9329±0.0213	**0.9259±0.0133**	0.9204±0.0139
ECOLI01VS235	0.6641±0.0252	0.6001±0.0809	0.6880±0.1319	**0.7154±0.0825**
YEAST05679VS4	0.4078±0.0642	0.4125±0.5555	0.4647±0.0262	**0.5239±0.0573**
YEAST4	0.2170±0.0783	0.1974±0.0699	**0.3634±0.1331**	0.2564±0.0851
DLBCL	/	/	0.5775±0.1575	**0.6407±0.2025**
CNS	/	/	0.4871±0.2377	**0.5282±0.1627**
GLI85	0.4852±0.1415	0.5219±0.2024	0.4996±0.2042	**0.5572±0.1733**
COLON	0.6721±0.1108	0.6469±0.1108	0.7286±0.1121	**0.7472±0.0769**

TABLE 4. *GMEAN* results of four compared algorithms.

Data sets	ROS	RUS	SMOTE	GAUS
GLASS4	**0.8787±0.1136**	0.8685±0.1101	0.8661±0.1079	0.8335±0.1255
ECOLI01VS5	0.8310±0.1401	0.8796±0.0724	0.8313±0.1638	**0.8909±0.0761**
ECOLI067VS35	0.8356±0.0947	0.7743±0.0578	0.8431±0.0983	**0.8471±0.0960**
ECOLI0146VS5	0.8607±0.1147	0.8661±0.0900	**0.8892±0.1101**	0.8681±0.1184
YEAST2VS8	0.6442±0.1807	**0.6467±0.1827**	**0.6467±0.1827**	**0.6467±0.1827**
ECOLI0347VS56	0.9214±0.0647	0.8828±0.1028	0.8397±0.0967	**0.9382±0.0605**
VEHICLE0	0.9537±0.0145	0.9602±0.0242	**0.9653±0.0167**	0.9591±0.0097
ECOLI01VS235	0.8226±0.0983	0.8563±0.0720	0.8240±0.1173	**0.8588±0.0862**
YEAST05679VS4	0.8109±0.0690	0.8278±0.0675	0.8127±0.0596	**0.8453±0.0422**
YEAST4	0.7519±0.0727	0.7523±0.0794	0.6744±0.0612	**0.7569±0.0610**
DLBCL	0.2767±0.2588	0.1622±0.1280	**0.2794±0.2703**	0.1650±0.1273
CNS	0.3335±0.3051	**0.4581±0.2850**	0.4099±0.1298	0.3308±0.2130
GLI85	0±0	0.1650±0.1289	0.3132±0.1238	**0.5395±0.1309**
COLON	0.7669±0.0918	0.7489±0.0853	0.8081±0.0930	**0.8268±0.0425**

not eliminate drawbacks brought by high dimension and imbalanced distribution. In *Acc* experiments, ROS gets the best result in DLBCL and RUS obtains the highest value in CNS, but GAUS performs better in GLI85 and COLON. We can make a conclusion that random sampling methods are not fit in high dimensional imbalanced data sets, and GAUS has a better comprehensive classification performance than other compared algorithms. It is because that we employ SU to implement feature selection to reduce the effects of irrelevant and noise features, and only GAUS based on V-statistic makes full use of information of training samples to improve classifier's performance, though other three compared algorithms are also running in processed data sets.

Finally, we can see that the results of GAUS in F_1, *Gmean* and *AUC* are better than those in *Acc*, it demonstrates that GAUS has an excellent performance in resolving imbalanced classification problems, and *Acc* indicator is inappropriate to measure imbalanced classification results.

C. VALIDATION OF GU-MOACOFS

In this section, we make exhaust experiments to evaluate the performance of our proposed GU-MOACOFS. We choose six state-of-art algorithms to make comparisons with the proposed method, i.e. ADASYN, RUSBoost, support vector machine recursive feature elimination (SVMRFE), minimum redundancy maximum relevance (MRMR), instance selection feature selection multiobjective evolutionary algorithm (IS+FS-MOEA) [32], and SYMON [30]. ADASYN and RUSBoost are two popular methods for imbalanced classification problems. As GU-MOACOFS employs MOACO to selects feature subsets for improving classification performance, we introduce two excellent feature selection algorithms, SVMRFE and MRMR, to demonstrate the advantages of our algorithm. IS+FS-MOEA is the first method based on the multiobjective evolutionary algorithm for the imbalanced classification problem, and it is realized by non-dominated sorting genetic algorithm II and has an outstanding capability.

TABLE 5. ***AUC*** **results of four compared algorithms.**

Data sets	ROS	RUS	SMOTE	GAUS
GLASS4	**0.8872±0.0993**	0.8784±0.0978	0.8770±0.1230	0.8457±0.1124
ECOLI01VS5	0.8496±0.1390	0.8839±0.0694	0.8521±0.1315	**0.8952±0.0720**
ECOLI067VS35	0.8467±0.0809	0.7858±0.0484	0.8547±0.0849	**0.8596±0.0814**
ECOLI0146VS5	0.8678±0.1072	0.8705±0.0865	**0.8966±0.1205**	0.8775±0.1088
YEAST2VS8	0.7183±0.1121	**0.7215±0.1144**	**0.7215±0.1144**	**0.7215±0.1144**
ECOLI0347VS56	0.9246±0.0614	0.8861±0.1395	0.8514±0.0834	**0.9415±0.0661**
VEHICLE0	0.9541±0.014	0.9608±0.0236	0.9656±0.0165	**0.9593±0.0096**
ECOLI01VS235	0.8401±0.0827	0.8624±0.0662	0.8380±0.1035	**0.8662±0.0780**
YEAST05679VS4	0.8134±0.0687	0.8326±0.0677	0.8180±0.0504	**0.8484±0.0383**
YEAST4	0.7602±0.0653	0.7582±0.0726	0.7192±0.0405	**0.7682±0.0532**
DLBCL	0.4908±0.0896	0.4454±0.1062	0.4854±0.0860	**0.5043±0.0829**
CNS	0.5239±0.1306	**0.6041±0.1153**	0.4887±0.1415	0.5186±0.0731
GLI85	0.5±0	0.5653±0.1460	0.5393±0.1379	**0.6387±0.0965**
COLON	0.7742±0.0896	0.7592±0.0866	0.8300±0.0737	**0.8392±0.0404**

TABLE 6. ***ACC*** **results of four compared algorithms.**

Data sets	ROS	RUS	SMOTE	GAUS
GLASS4	0.9116±0.0303	0.8651±0.0624	**0.9255±0.0255**	0.9116±0.0303
ECOLI01VS5	0.9333±0.0271	0.9125±0.0401	**0.9375±0.0147**	0.9333±0.0271
ECOLI067VS35	0.9240±0.0202	0.8126±0.0948	0.9375±0.0095	**0.9465±0.0119**
ECOLI0146VS5	0.8892±0.0233	0.8607±0.0196	0.9428±0.0196	**0.9500±0.0387**
YEAST2VS8	0.9689±0.0242	**0.9752±0.0208**	**0.9752±0.0208**	**0.9752±0.0208**
ECOLI0347VS56	0.9414±0.0196	0.8707±0.1097	0.9295±0.0383	**0.9726±0.0176**
VEHICLE0	0.9657±0.0098	0.9716±0.0088	**0.9669±0.0088**	0.9645±0.0094
ECOLI01VS235	0.9102±0.0398	0.8612±0.0566	**0.9184±0.0456**	**0.9184±0.0456**
YEAST05679VS4	0.8117±0.0568	0.8059±0.0536	0.8631±0.0308	**0.8840±0.0321**
YEAST4	0.8532±0.0210	0.8283±0.0288	**0.9482±0.0173**	0.8855±0.0192
DLBCL	**0.5454±0.1175**	0.4424±0.1013	0.4955±0.1689	0.5136±0.2336
CNS	0.5333±0.0745	**0.6333±0.0745**	0.4500±0.1828	0.4333±0.1490
GLI85	0.3294±0.1220	0.4000±0.2544	0.4706±0.1715	**0.5177±0.1578**
COLON	0.7795±0.0739	0.7487±0.0460	0.7667±0.1067	**0.8128±0.0401**

SYMON is a powerful feature selection algorithm for high dimensional imbalanced classification.

The parameters of GU-MOACOFS are set as follows, GAUS's parameters are unchanged, the number of data subsets generated by Boostrap is 21 (i.e. $n = 21$), maximum number of iterations of MOACO $ite\_m = 200$, pheromone importance degree $\alpha = 1$, heuristic information importance degree $\beta = 1$, the number of solutions in Pareto archive is 60, MOACO population size is $N\_m = 30$, pheromone evaporation rate $\rho = 0.1$, weight parameter $\lambda = 0.5$. SVM is used as classifier, and it uses RBF kernel, $\sigma = 0.4$ and $C = 100$. The parameters of other algorithms are set the same as those in their original papers. Table 7 to table 10 give the F_1, *Gmean*, *AUC* and *Acc* results of seven compared algorithms.

We still compare those algorithms in the opinion of low and high dimensional data. We analyze the results in ten low dimensional data sets firstly. In F_1 results, GU-MOACOFS provides best values in eight data sets while SYMON only performs better in ECOLI0146VS5 and YEAST2VS8. The results of SVMRFE and MRMR in three data sets do not exist, which means their precision or recall values are zero and they cannot find minority class samples. In *Gmean* experiments, GU-MOACOFS gets the best solutions in eight data sets, while IS+FS-MOEA obtains better outcomes in ECOLI0347VS56 and ECOLI01VS235. In *AUC* results, GU-MOACOFS performs better in eight data sets except the previous two data sets where IS+FS-MOEA also provides the best solutions. It indicates that SVMRFE and MRMR cannot resolve imbalanced classification problems as their results are 0.5 in YEAST05679VS4 and YEAST4. In table 10, GU-MOACOFS is better in five data sets, SYMON performs best on three data sets, and SVMRFE outstands in two data sets. From previous statistic results, we can find that conventional feature selection algorithms, i.e. SVMRFE and MRMR, are not suitable for imbalanced data classification

TABLE 7. F_1 results of four compared algorithms.

Data sets	ADASYN	RUSBOOST	SVMRFE	MRMR	IS+FS-MOEA	SYMON	GU-MOACOFS
GLASS4	0.7072±0.1908	0.5997±0.2428	/	/	0.7255±0.0750	0.7933±0.1251	**0.8102±0.1283**
ECOLI01VS5	0.7981±0.1833	0.8278±0.1135	0.7238±0.0599	0.7732±0.0672	0.7975±0.1522	0.8967±0.1007	**0.9152±0.0444**
ECOLI067VS35	0.7029±0.1002	0.6096±0.1137	0.7661±0.0809	0.7534±0.0827	0.7819±0.1449	0.8025±0.0850	**0.8306±0.0553**
ECOLI0146VS5	0.5963±0.1183	0.6117±0.1029	0.7239±0.0681	0.7767±0.0812	0.7289±0.1979	**0.8918±0.1043**	0.8592±0.0951
YEAST2VS8	0.3082±0.2921	0.4538±0.2205	0.6736±0.0829	0.6925±0.0881	0.5733±0.1588	**0.7267±0.1862**	0.6804±0.1478
ECOLI0347VS56	0.7311±0.1675	0.7695±0.0850	0.7897±0.0582	0.7233±0.0573	0.9044±0.0928	0.8095±0.1429	**0.9481±0.0485**
VEHICLE0	0.9195±0.0216	0.8348±0.0501	0.9325±0.0115	0.9319±0.0131	0.9075±0.0469	0.9401±0.0284	**0.9429±0.0209**
ECOLI01VS235	0.7056±0.0796	0.7621±0.1177	0.7233±0.0758	0.6982±0.0740	0.7982±0.1610	0.8406±0.1468	**0.8512±0.1192**
YEAST05679VS4	0.5069±0.0899	0.4823±0.0681	/	/	0.5456±0.1168	0.5035±0.1374	**0.5755±0.1212**
YEAST4	0.3187±0.0462	0.3061±0.1186	/	/	0.3228±0.0933	0.3413±0.0430	**0.4473±0.1083**
DLBCL	0.5374±0.1294	0.5333±0.1960	0.4796±0.0599	0.5241±0.0674	0.8232±0.1040	0.8272±0.0856	**0.9256±0.0059**
CNS	0.4824±0.1737	0.4833±0.1628	0.5351±0.0425	0.4406±0.0835	0.8123±0.1696	0.8499±0.0621	**0.8853±0.0573**
GLI85	0.5431±0.1535	0.7293±0.1456	0.4268±0.1090	0.6686±0.0806	0.9448±0.0719	0.9017±0.0641	**0.9765±0.0122**
COLON	0.7551±0.1678	0.6351±0.1975	0.7144±0.0771	0.6158±0.0713	0.9034±0.0620	0.7816±0.1394	**0.9664±0.0162**

TABLE 8. *GMEAN* results of four compared algorithms.

Data sets	ADASYN	RUSBOOST	SVMRFE	MRMR	IS+FS-MOEA	SYMON	GU-MOACOFS
GLASS4	0.8974±0.1006	0.8446±0.1894	0.2310±0.3163	0.2310±0.3163	0.9482±0.0384	0.9609±0.0600	**0.9848±0.0144**
ECOLI01VS5	0.9364±0.0586	0.8738±0.0827	0.8292±0.0468	0.8590±0.0513	0.9771±0.0159	0.9188±0.0942	**0.9804±0.0180**
ECOLI067VS35	0.9180±0.1002	0.8274±0.0905	0.8239±0.0548	0.8204±0.0617	0.8832±0.1088	0.8544±0.1033	**0.9371±0.0596**
ECOLI0146VS5	0.8940±0.1053	0.8628±0.0852	0.8210±0.0539	0.8518±0.0585	0.9170±0.0539	0.9008±0.0935	**0.9693±0.0126**
YEAST2VS8	0.7560±0.1580	0.7716±0.1548	0.7407±0.0737	0.7478±0.0715	0.6357±0.1255	0.7616±0.1579	**0.8354±0.1100**
ECOLI0347VS56	0.9245±0.0542	0.9237±0.0569	0.8534±0.0501	0.7867±0.0442	**0.9844±0.0149**	0.8494±0.1063	0.9044±0.0928
VEHICLE0	0.9583±0.0169	0.9191±0.0235	0.9535±0.0092	0.9536±0.0105	0.9643±0.0130	0.9710±0.0150	**0.9797±0.0085**
ECOLI01VS235	0.8897±0.1099	0.8688±0.0748	0.8284±0.0558	0.7781±0.0588	**0.9724±0.0224**	0.8808±0.1092	0.9428±0.0436
YEAST05679VS4	0.8141±0.0538	0.8042±0.0780	0±0	0±0	0.8246±0.0292	0.7893±0.0978	**0.8284±0.1004**
YEAST4	0.8578±0.0369	0.7902±0.0597	0±0	0±0	0.8667±0.0591	0.8514±0.0840	**0.8910±0.0733**
DLBCL	0.5072±0.1376	0.5646±0.1601	0.4545±0.0816	0.5510±0.0702	0.7450±0.1422	0.8356±0.0609	**0.9110±0.0158**
CNS	0.3692±0.1679	0.5797±0.1374	0.5721±0.0217	0.4955±0.0391	0.7059±0.1822	0.8755±0.0555	**0.9251±0.0169**
GLI85	0.1447±0.3237	0.8135±0.1178	0.6238±0.0993	0.7534±0.0702	0.9642±0.0497	0.9494±0.0370	**0.9765±0.0122**
COLON	0.8116±0.1221	0.7309±0.1650	0.7856±0.0625	0.7042±0.0524	0.8876±0.0658	0.8332±0.1108	**0.9677±0.0244**

problems, and ADASYN and RUSBoost can settle drawbacks brought by imbalanced distribution at a certain extent. It is better to combine instance selection and feature selection than only use feature selection as IS+FS-MOEA is superior to SYMON in ten data sets. In the end, GU-MOACOFS is superior to other six algorithms in ten data sets in the opinion of four indicators, and it indicates that employing ensemble learning, sampling policy and feature selection at the same time can achieve a better result, and ensemble learning can further promote algorithm's performance.

Now, we see the results given by seven algorithms in four high dimensional data sets. It is easy to find out that GU-MOACOFS is the best algorithm in all four testing data sets. IS+FS-MOEA is better than SYMON in GLI85 and COLON while SYMON is better in DLBCL and CNS. Besides, conventional sampling methods, ADASYN and RUSBoost, and feature selection algorithms, SVMRFE and MRMR, cannot handle high dimensional imbalanced data classification problems effectively. As high dimension property and imbalanced distribution interact with each other, and the combination of them brings more difficulties. Sampling policy only resolves imbalanced problems and feature selection merely settles high dimension difficulties, so they may not account for high dimension and imbalanced characteristics at the same time. Experiments show that SYMON performance is not superior to IS+FS-MOEA significantly though SYMON is designed for solving high dimensional imbalanced classification problems. With the results in ten low dimensional data sets, we can make a conclusion that adopting instance selection and feature selection

TABLE 9. ***AUC*** **results of four compared algorithms.**

Data sets	ADASYN	RUSBOOST	SVMRFE	MRMR	IS+FS-MOEA	SYMON	GU-MOACOFS
GLASS4	0.9023±0.0948	0.8594±0.1345	0.5617±0.0959	0.5643±0.0936	0.9496±0.0368	0.9624±0.0568	**0.9850±0.0142**
ECOLI01VS5	0.9393±0.0550	0.8827±0.0747	0.8480±0.0385	0.8741±0.0441	0.9775±0.0154	0.9252±0.0842	**0.9812±0.0164**
ECOLI067VS35	0.9203±0.0562	0.8396±0.0769	0.8472±0.0451	0.8442±0.0484	0.8962±0.1004	0.8693±0.0875	**0.9399±0.0562**
ECOLI0146VS5	0.8985±0.0989	0.8713±0.0784	0.8438±0.0430	0.8681±0.0473	0.9213±0.0448	0.9100±0.0825	**0.9705±0.0202**
YEAST2VS8	0.7706±0.1490	0.7981±0.1267	0.7834±0.0537	0.7907±0.0541	0.7083±0.0833	0.8000±0.1264	**0.8525±0.0940**
ECOLI0347VS56	0.9254±0.0542	0.9247±0.0560	0.8701±0.0402	0.8147±0.0343	**0.9846±0.0147**	0.8646±0.0914	0.9756±0.0186
VEHICLE0	0.9585±0.0168	0.9193±0.0236	0.9541±0.0089	0.9542±0.0102	0.9646±0.0128	0.9711±0.0149	**0.9798±0.0085**
ECOLI01VS235	0.8984±0.0933	0.8772±0.0688	0.8470±0.0461	0.8070±0.0463	**0.9730±0.0218**	0.8957±0.1001	0.9460±0.0493
YEAST05679VS4	0.8155±0.0523	0.8108±0.0658	0.5±0	0.5±0	0.8278±0.0298	0.7997±0.0865	**0.8375±0.0941**
YEAST4	0.8588±0.0361	0.7955±0.0543	0.5±0	0.5±0	0.8676±0.0587	0.8560±0.0777	**0.8948±0.0686**
DLBCL	0.5727±0.1217	0.5971±0.1587	0.5283±0.0779	0.5765±0.0687	0.8341±0.1578	0.8489±0.0519	**0.9151±0.0144**
CNS	0.5891±0.1274	0.6014±0.1309	0.5901±0.0295	0.5689±0.0319	0.8164±0.1880	0.8811±0.0537	**0.9280±0.0157**
GLI85	0.5452±0.1012	0.8235±0.1116	0.6426±0.0087	0.7701±0.0630	0.9658±0.0467	0.9512±0.0353	**0.9765±0.0122**
COLON	0.8245±0.1093	0.7498±0.1521	0.7978±0.0561	0.7245±0.0478	0.9289±0.0354	0.8482±0.0945	**0.9691±0.0226**

TABLE 10. ***ACC*** **results of four compared algorithms.**

Data sets	ADASYN	RUSBOOST	SVMRFE	MRMR	IS+FS-MOEA	SYMON	GU-MOACOFS
GLASS4	0.9298±0.0406	0.9114±0.0553	0.9391±0.0217	0.9439±0.0132	0.9393±0.0206	0.9718±0.0197	**0.9721±0.0255**
ECOLI01VS5	0.9625±0.0309	0.9625±0.0309	0.9571±0.0095	0.9644±0.0129	0.9583±0.0295	0.9792±0.0208	**0.9917±0.0014**
ECOLI067VS35	0.9281±0.0242	0.9147±0.0293	0.9562±0.0167	0.9539±0.0159	0.8888±0.0993	**0.9686±0.0124**	0.9640±0.0199
ECOLI0146VS5	0.9214±0.0324	0.9321±0.0149	0.9653±0.0094	0.9718±0.0096	0.9393±0.0505	**0.9857±0.0140**	0.9786±0.0149
YEAST2VS8	0.8366±0.0839	0.9355±0.0372	0.9771±0.0060	0.9794±0.0075	0.9792±0.0074	**0.9855±0.0093**	0.9731±0.0118
ECOLI0347VS56	0.9151±0.0482	0.9419±0.0273	0.9620±0.0110	0.9529±0.0147	0.9729±0.0259	0.9686±0.0224	**0.9882±0.0105**
VEHICLE0	0.9610±0.0089	0.9172±0.0152	0.9682±0.0052	0.9678±0.0058	0.9539±0.0210	0.9717±0.0134	**0.9728±0.0067**
ECOLI01VS235	0.9219±0.0088	0.9589±0.0143	0.9490±0.0121	0.9492±0.0096	0.9508±0.0399	0.9714±0.0310	**0.9755±0.0171**
YEAST05679VS4	0.8257±0.0313	0.8599±0.0305	**0.9027±0.0122**	0.9014±0.0116	0.8784±0.0548	0.8748±0.0311	0.8845±0.0337
YEAST4	0.8638±0.0140	0.8694±0.0213	**0.9663±0.0042**	0.9640±0.0047	0.8761±0.0205	0.8794±0.0215	0.9461±0.0114
DLBCL	0.4985±0.1389	0.5727±0.1374	0.4924±0.0607	0.5668±0.0716	0.8181±0.1578	0.8329±0.0650	**0.9334±0.0289**
CNS	0.5394±0.1503	0.6000±0.1294	0.5958±0.0344	0.5396±0.0251	0.7889±0.1279	0.8710±0.0534	**0.9292±0.0250**
GLI85	0.4706±0.2121	0.8259±0.1080	0.6353±0.0824	0.7784±0.0580	0.9634±0.0481	0.9412±0.0416	**0.9765±0.0122**
COLON	0.8406±0.0949	0.7664±0.1337	0.8032±0.0570	0.7201±0.0488	0.9060±0.0858	0.8579±0.0877	**0.9667±0.0256**

simultaneously can obtain a universal fine performance, and ensemble learning can further promote algorithm's ability. So GU-MOACOFS could resolve high dimensional imbalanced data classification problems more effectively.

At last, we talk about an interesting finding. In the results of *Gmean* and *AUC*, we mark three values generated by GU-MOACOFS through underline, and they are the outcomes in two low dimensional data sets, i.e. ECOLI0347VS56 and YEAST05679VS4. We focus on them as they are worse than the results provided by GAUS in table 4 and table 5. It indicates that though GU-MOACOFS performs better in most cases, feature selection may lead to the deterioration of the algorithm in some data sets. A possible reason is that features have a high relevance in those data sets, and removing features may result in a worse situation. So we have to be careful about using feature selection in some situations.

V. CONCLUSIONS

Imbalanced classification problems widely exist in real-world engineering applications, and conventional algorithms' precondition is balanced data. Imbalanced distribution may undermine the performance of previous methods; thus, we propose a new method called GU-MOACOFS which combines ensemble learning, evolutionary under-sampling and multiobjective feature selection for resolving imbalanced classification problems. It employs Bootstrap to generate some data subsets based on the Bagging framework,

implements under-sampling on data subsets to produce sampling subsets by genetic algorithm based on proposed V-statistic, and uses F_1 and *Gmean* indicator as objectives of multiobjective ant colony optimization to select feature subset to improve the performance of classifiers. Experiments show GU-MOACOFS can resolve imbalanced classification problem efficiently, especially in high dimensional data sets. Besides, some conclusions are made as follows.

Firstly, V-statistic for the genetic algorithm can better measure the complexity of data distribution and make use of data information to resample more discriminant sampling subsets.

Secondly, high-dimensional imbalanced data has high dimensionality and imbalanced property at the same time, which leads to a more complex situation of classification. The traditional sampling method or feature selection algorithm cannot handle it completely. Hence, it is more reasonable to combine resampling and feature selection to perform classification in high-dimensional imbalanced data.

Thirdly, it is demonstrated again that ensemble learning can further improve algorithm's performance and robustness.

Finally, though feature selection can promote the ability of classification algorithm in most cases, it may not work well because of strong correlation between features in some data sets. Therefore, we should be more careful when using feature selection.

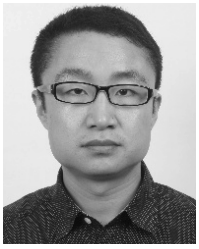
The proposed GU-MOACOFS is more a framework than a pure algorithm. Its components can be replaced by other alternative methods according to user's demands. For example, the genetic algorithm can be replaced with RUS or SMOTE; also, MOACO can be replaced with other feature selection methods. Thus, GU-MOACOFS can be regarded as a flexible prototype of the imbalanced classification framework. Besides, we acknowledge that our proposed method is time-consuming compared with other algorithms in this paper, but it is the cost of its better performance, which is consistent with conclusion of thesis ''No free lunch''. In the future, we will use GU-MOACOFS to resolve the problem of some real-world applications and further promote its performance.

REFERENCES

[1] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, ''Feature selection: A data perspective,'' *ACM Comput. Surv.*, vol. 50, no. 6, p. 94, 2018.
[2] R. Sharma and A. S. Bist, ''Machine learning: A survey,'' *Int. J. Eng. Sci. Res. Technol.*, vol. 4, no. 3, pp. 708–716, 2015.
[3] A. Kilani, A. Ben Hamida, and H. Hamam, ''Artificial intelligence review,'' in *Encyclopedia of Information Science and Technology*, M. Khosrow-Pour, Ed., 4th ed. Hershey, PA, USA: IGI Global, 2018, pp. 106–119. doi: 10.4018/978-1-5225-2255-3.ch010.
[4] A. Fernández, S. D. Río, N. V. Chawla, and F. Herrera, ''An insight into imbalanced big data classification: Outcomes and challenges,'' *Complex Intell. Syst.*, vol. 3, no. 2, pp. 105–120, 2017.
[5] B. Krawczyk, ''Learning from imbalanced data: Open challenges and future directions,'' *Prog. Artif. Intell.*, vol. 5, no. 4, pp. 221–232, 2016.
[6] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, ''Learning from class-imbalanced data: Review of methods and applications,'' *Expert Syst. Appl.*, vol. 73, pp. 220–239, May 2017.
[7] P. Branco, L. Torgo, and R. P. Ribeiro, ''A survey of predictive modeling on imbalanced domains,'' *ACM Comput. Surv.*, vol. 49, no. 2, p. 31, 2016.
[8] L. Yi, D. Xing-Chun, C. Jian-jun, Z. Xing, and S. Yu-ling, ''A method for entity resolution in high dimensional data using ensemble classifiers,'' *Math. Problems Eng.*, vol. 2017, Feb. 2017, Art. no. 4953280.
[9] H. He and E. A. Garcia, ''Learning from imbalanced data,'' *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.
[10] J. Ha and J.-S. Lee, ''A new under-sampling method using genetic algorithm for imbalanced data classification,'' in *Proc. 10th Int. Conf. Ubiquitous Inf. Manage. Commun.*, 2016, p. 95.
[11] J. Pérez-Rodríguez, A. G. Arroyo-Peña, and N. García-Pedrajas, ''Simultaneous instance and feature selection and weighting using evolutionary computation: Proposal and study,'' *Appl. Soft Comput.*, vol. 37, pp. 416–443, Dec. 2015.
[12] Y. Qian, Y. Liang, M. Li, G. Feng, and X. Shi, ''A resampling ensemble algorithm for classification of imbalance problems,'' *Neurocomputing*, vol. 143, pp. 57–67, Nov. 2014.
[13] B. W. Yap, K. A. Rani, H. A. A. Rahman, S. Fong, Z. Khairudin, and N. N. Abdullah, ''An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets,'' in *Proc. 1st Int. Conf. Adv. Data Inf. Eng.*, Singapore, 2013, pp. 13–22.
[14] J. Luengo, A. Fernández, S. García, and F. Herrera, ''Addressing data complexity for imbalanced data sets: Analysis of SMOTE-based oversampling and evolutionary undersampling,'' *Soft Comput.*, vol. 15, no. 10, pp. 1909–1936, 2011.
[15] H. Yu, J. Ni, and J. Zhao, ''ACOSampling: An ant colony optimization-based undersampling method for classifying imbalanced DNA microarray data,'' *Neurocomputing*, vol. 101, no. 3, pp. 309–318, 2013.
[16] B. Krawczyk, M. Galar, Łukasz Jeleń, and F. Herrera, ''Evolutionary undersampling boosting for imbalanced classification of breast cancer malignancy,'' *Appl. Soft Comput.*, vol. 38, pp. 714–726, Jan. 2016.
[17] I. Triguero, M. Galar, S. Vluymans, C. Cornelis, H. Bustince, F. Herrera, and Y. Saeys, ''Evolutionary undersampling for imbalanced big data classification,'' in *Proc. IEEE Congr. Evol. Comput. (CEC)*, May 2015, pp. 715–722.
[18] A. Moreo, A. Esuli, and F. Sebastiani, ''Distributional random oversampling for imbalanced text classification,'' in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Pisa, Italy, 2016, pp. 805–808.
[19] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, ''SMOTE: Synthetic minority over-sampling technique,'' *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 321–357, 2002.
[20] F. Charte, A. J. Rivera, M. J. D. Jesus, and F. Herrera, ''MLSMOTE: Approaching imbalanced multilabel learning through synthetic instance generation,'' *Knowl.-Based Syst.*, vol. 89, pp. 385–397, Nov. 2015.
[21] H. B. He, Y. Bai, E. A. Garcia, and S. Li, ''ADASYN: Adaptive synthetic sampling approach for imbalanced learning,'' in *Proc. IEEE World Congr. Comput. Intell.*, Hong Kong, China, Jun. 2008, pp. 1322–1328.
[22] E. Ramentol, I. Gondres, S. Lajes, R. Bello, Y. Caballero, C. Cornelis, and F. Herrera, ''Fuzzy-rough imbalanced learning for the diagnosis of High Voltage Circuit Breaker maintenance: The SMOTE-FRST-2T algorithm,'' *Eng. Appl. Artif. Intell.*, vol. 48, pp. 134–139, Feb. 2016.
[23] A. K. I. Hassan and A. Abraham, ''Modeling insurance fraud detection using imbalanced data classification,'' *Adv. Nature Biologically Inspired Comput.*, vol. 419, pp. 117–127, 2016.
[24] B. Krawczyk and G. Schaefer, ''Effective imbalanced classification of breast thermogram features,'' in *Proc. Pattern Recognit. Mach. Intell.*, 2015, pp. 535–544.
[25] F. Cheng, J. Zhang, and C. Wen, ''Cost-sensitive large margin distribution machine for classification of imbalanced data,'' *Pattern Recognit. Lett.*, vol. 80, pp. 107–112, Sep. 2016.
[26] S. Ali, A. Majid, S. G. Javed, and M. Sattar, ''Can-CSC-GBE: Developing cost-sensitive classifier with GentleBoost ensemble for breast cancer classification using protein amino acids and imbalanced data,'' *Comput. Biol. Med.*, vol. 73, pp. 38–46, Jun. 2016.
[27] L. Mena and J. A. Gonzalez, ''Symbolic one-class learning from imbalanced datasets: Application in medical diagnosis,'' *Int. J. Artif. Intell. Tools*, vol. 18, no. 2, pp. 273–309, 2009.
[28] Z.-S. Pan, B. Chen, Z. M. Miao, and G.-Q. Ni, ''Overview of study on one-class classifier,'' *Acta Electron. Sinica*, vol. 37, no. 11, pp. 2496–2503, 2009.
[29] L. Yin, Y. Ge, K. Xiao, X. Wang, and X. Quan, ''Feature selection for high-dimensional imbalanced data,'' *Neurocomputing*, vol. 105, no. 3, pp. 3–11, 2013.
[30] A. Moayedikia, K.-L. Ong, Y. L. Boo, W. G. Yeoh, and R. Jensen, ''Feature selection for high dimensional imbalanced class data using harmony search,'' *Eng. Appl. Artif. Intell.*, vol. 57, pp. 38–49, Jan. 2017.
[31] L.-M. Du, Y. Xu, and H. Zhu, ''Feature selection for multi-class imbalanced data sets based on genetic algorithm,'' *Ann. Data Sci.*, vol. 2, no. 3, pp. 293–300, 2015.

[32] A. Fernández, M. J. D. Jesus, and F. Herrera, "Addressing overlapping in classification with imbalanced datasets: A first multi-objective approach for feature and instance selection," in *Proc. Intell. Data Eng. Automated Learn.*, 2015, pp. 36–44.
[33] A. Braytee, W. Liu, and P. Kennedy, "A cost-sensitive learning strategy for feature extraction from imbalanced data," in *Proc. Int. Conf. Neural Inf. Process.*, Kyoto, Japan, 2016, pp. 78–86.
[34] W. W. Y. Ng, G. Zeng, J. Zhang, D. S. Yeung, and W. Pedrycz, "Dual autoencoders features for imbalance classification problem," *Pattern Recognit.*, vol. 60, pp. 875–889, Dec. 2016.
[35] B. Wang and J. Pineau, "Online bagging and boosting for imbalanced data streams," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 12, pp. 3353–3366, Dec. 2016.
[36] Z. Sun, Q. Song, X. Zhu, H. Sun, B. Xu, and Y. Zhou, "A novel ensemble method for classifying imbalanced data," *Pattern Recognit.*, vol. 48, no. 5, pp. 1623–1637, 2015.
[37] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTEBoost: Improving prediction of the minority class in boosting," in *Knowledge Discovery in Databases*, vol. 2838. Dubrovnik, Croatia: Springer, 2003, pp. 107–119.
[38] G. Haixiang, L. Yijing, L. Yanan, L. Jinlinga, and L. Xiao, "BPSO-Adaboost-KNN ensemble learning algorithm for multi-class imbalanced data classification," *Eng. Appl. Artif. Intell.*, vol. 49, pp. 176–193, Mar. 2016.
[39] L. Yijing, G. Haixiang, L. Xiao, L. Yanan, and L. Jinling, "Adapted ensemble classification algorithm based on multiple classifier system and feature selection for classifying multi-class imbalanced data," *Knowl.-Based Syst.*, vol. 94, pp. 88–104, Feb. 2015.
[40] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: A hybrid approach to alleviating class imbalance," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 1, pp. 185–197, Jan. 2010.
[41] M. Galar, A. Fernández, E. Barrenechea, and F. Herrera, "EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling," *Pattern Recognit.*, vol. 46, no. 12, pp. 3460–3471, Dec. 2013.
[42] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 2, pp. 539–550, Apr. 2009.
[43] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Inf. Sci.*, vol. 250, pp. 113–141, Nov. 2013.
[44] H.-L. Dai, "Imbalanced protein data classification using ensemble FTM-SVM," *IEEE Trans. Nanobiosci.*, vol. 14, no. 4, pp. 350–359, Jun. 2015.
[45] T. K. Ho and M. Basu, "Complexity measures of supervised classification problems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 289–300, Mar. 2002.
[46] J. J. Cao, P. L. Zhang, Y. X. Wang, G. Q. Ren, and J. P. Fu, "Graph-based ant system for subset problems," *J. Syst. Simul.*, vol. 20, no. 22, pp. 6146–6150, 2008.
[47] R. Ruiz, J. C. Riquelme, J. S. Aguilar-Ruiz, and M. García-Torres, "Fast feature selection aimed at high-dimensional data via hybrid-sequential-ranked searches," *Expert Syst. Appl.*, vol. 39, no. 12, pp. 11094–11102, 2012.

YI LIU was born in Bengbu, Anhui, China in 1990. He received the B.S. degree in network engineering from the Xi'an University of Posts & Telecommunications, Xi'an, Shanxi, in 2011, the M.S. degree in computer application technology from the PLA University of Science and Technology, Nanjing, Jiangsu, in 2014, and the Ph.D. degree in software engineering from the Army Engineering University of PLA, Nanjing, in 2018. He is currently an Assistant Researcher with the National Innovation Institute of Defense Technology (NIIDT). His research interests include machine learning, evolutionary algorithms, and data quality.



YANZHEN WANG received the Ph.D. degree in computer science from the National University of Defense Technology (NUDT), China, in 2011. He is currently an Associated Researcher with the Artificial Intelligence Research Center (AIRC), National Innovation Institute of Defense Technology (NIIDT), and the Tianjin Artificial Intelligence Innovation Center (TAIIC). His research interests include robotics software, computer graphics, and virtual reality.

XIAOGUANG REN received the B.S., M.S., and Ph.D. degrees in computer science and technology from the National University of Defense Technology (NUDT), China, in 2008, 2010, and 2014, respectively. He was a Lecturer of computer science and technology with the High Performance Computing Laboratory, National University of Defense Technology (NUDT), until 2018. He is currently an Assistant Research Fellow with the National Innovation Institute of Defense Technology (NIIDT). His research interests include high-performance computing, numerical computation and simulation, robot operation systems.

HAO ZHOU received the B.S., M.S., and Ph.D. degrees in computer science and technology from the National University of Defense Technology, China, in 2009, 2011, and 2016, respectively. He is currently a Research Assistant with the National Innovation Institute of Defense Technology (NIIDT). His research interests include programming language and complier, high-performance computing, artificial intelligence, and robot operation systems.

XINGCHUN DIAO was born in Taixing, Jiangsu, China, in 1964. He received the master's degree from the PLA National University of Defense Technology, Changsha.

He is currently a Researcher and a Ph.D. Supervisor. He has been involved in the research of data quality control, data analysis, and mining for a long time. He is one of the co-sponsors of the Information Quality Research Group, China. He has authored three data quality translations and over 90 academic papers at important conferences and journals. He holds two authorized invention patents and two authorized software copyrights. He holds six invention patents.

Mr. Diao received ten provincial science and technology progress prizes. He enjoys the special allowance experts of the State Council. He has successively presided over a number of important scientific research projects.