Research Article

Qingwei Zhou, Yongjun Qi*, Hailin Tang, and Peng Wu

# Machine learning-based processing of unbalanced data sets for computer algorithms

https://doi.org/10.1515/comp-2022-0273
received January 13, 2023; accepted March 16, 2023

**Abstract:** The rapid development of technology allows people to obtain a large amount of data, which contains important information and various noises. How to obtain useful knowledge from data is the most important thing at this stage of machine learning (ML). The problem of unbalanced classification is currently an important topic in the field of data mining and ML. At present, this problem has attracted more and more attention and is a relatively new challenge for academia and industry. The problem of unbalanced classification involves classifying data when there is insufficient data or severe category distribution deviations. Due to the inherent complexity of unbalanced data sets, more new algorithms and tools are needed to effectively convert a large amount of raw data into useful information and knowledge. Unbalanced data set is a special case of classification problem, in which the distribution between classes is uneven, and it is difficult to classify data accurately. This article mainly introduces the research on the processing method of computer algorithms based on the processing method of unbalanced data sets based on ML, aiming to provide some ideas and directions for the processing of computer algorithms based on unbalanced data sets based on ML. This article proposes a research strategy for processing unbalanced data sets based on ML, including data preprocessing, decision tree data classification algorithm, and C4.5 algorithm, which are used to conduct research experiments on processing methods for unbalanced data sets based on ML. The experimental results in this article show that the accuracy rate of the decision tree C4.5 algorithm based on ML is 94.80%, which can be better used for processing unbalanced data sets based on ML.

**Keywords:** machine learning, unbalanced data sets, classification algorithm, data processing, computer algorithm, decision tree classification

## 1 Introduction

For decades, artificial intelligence has been a hot topic in the information field. Since the 1950s, people have been looking forward to the development of a revolutionary software that has multiple functions such as thinking and logical judgment, voice and image processing, and providing expert opinions. Intelligent systems, such as machine learning (ML), face recognition, video surveillance, voice image processing software, etc., have been widely used in the current society. A large part of ML technology is derived from statistical methods and to a large extent depends on the efficiency of numerical algorithms, using more and more powerful computing platforms and the availability of the world's huge data sets. In addition, because the results of ML have been easily made available to the public through various methods (such as cloud networks, etc.), people's interest in ML will continue to rise sharply, which will further contribute to society, economy, and science. In the 1950s, ML began to emerge. With nearly 70 years of development, ML has become the most cutting-edge research field in artificial intelligence.

An unbalanced data set means that the proportion of samples in the majority class is much greater than the proportion of samples in the minority class. From daily life to national security, from enterprise information processing to government decision support systems, from micro data analysis to macro scale, etc., in the practical

* **Corresponding author: Yongjun Qi,** Faculty of Megadata and Computing, Guangdong Baiyun University, Guangzhou 510450, Guangdong, China; School of Information and Communication Technology, Mongolian University of Science and Technology, Bayanzurkh District, 13341, Ulaanbaatar, Mongolia, e-mail: qyj200702022@baiyunu.edu.cn
**Qingwei Zhou:** School of Information and Engineering, Sichuan Tourism University, Chengdu 610000, Sichuan, China, e-mail: QingweiZhou@sctu.edu.cn
**Hailin Tang:** Faculty of Megadata and Computing, Guangdong Baiyun University, Guangzhou 510450, Guangdong, China; School of Information and Communication Technology, Mongolian University of Science and Technology, Bayanzurkh District, 13341, Ulaanbaatar, Mongolia, e-mail: linht88@163.com
**Peng Wu:** School of Information and Engineering, Sichuan Tourism University, Chengdu 610000, Sichuan, China, e-mail: 0001524@sctu.edu.cn

applications of these data mining, unbalanced data problems are common, such as big data analysis, text mining, natural disaster prediction, transaction fraud detection, satellite radar image target recognition, biological anomaly classification, and computer-aided medical diagnosis and treatment. In the 2005 ICDM (IEEE International Conference on Data Mining) international conference, the problem of unbalanced data mining was listed as one of the ten challenging problems in the field of data mining. And an unbalanced data set also has a great influence on the processing method of computer algorithms. A new challenge, whether it is in academia or in the industry, has attracted widespread attention.

Vollant et al. explored a new procedure for developing models in the context of passive scalar Large Eddy Simulation (LES), relying on the combination of best estimator theory and ML algorithms. The concept of the best estimator can determine the most accurate parameter set to be used when deriving the model, and then, the model itself can be defined by training an artificial neural network (ANN) in a database, which is filtered from direct numerical simulation (DNS) results inferred. This process produces a sub-grid scale model that shows good structural performance, which allows the execution of LES very close to the filtered DNS results. However, the first process cannot control the performance of the model, so when the process configuration is different from the training database, the model may fail. So Vollant A proposes another procedure in which model functional forms are applied, and ANN is only used to define model coefficients. The training step is a dual objective optimization to control structural and functional performance. This research is not comprehensive enough and lacks persuasiveness [1]. Hunt et al. found that ML algorithms based on deep neural networks (NN) have achieved remarkable results and have been widely used in different fields. On the other hand, with the growth of cloud services, several machine learning as a service (MLaaS) has emerged, which can be trained and deployed on the cloud provider's infrastructure. However, ML algorithms require access to raw data that is usually sensitive to privacy and may create potential security and privacy risks. In order to solve this problem, Hunt T proposed CryptoDL, which is a framework that develops new technologies to provide solutions for applying deep neural network algorithms to encrypted data. Hunt T provides a theoretical basis for the realization of deep neural network algorithms in the encryption domain and develops a technology that uses NN within the practical limits of current homomorphic encryption schemes, demonstrates the applicability of CryptoDL proposed using a large number of data sets, and evaluates its performance. This research lacks experimental data support and is weak in scientificity [2]. Li et al. apply ML technology with genetic algorithm to determine the polarization force field parameters, using only the initial quantum mechanics (QM) calculation from MP2/6-31 G(d, p) molecular clusters Data; DFMP2(fc)/jul-cc-pVDZ and DFMP2(fc)/jul-cc-pVTZ levels are used to predict the experimental condensation phase properties (i.e. density and heat of vaporization). The training data set obtained by QM calculation and the optimized force field model achieve excellent consistency. The compensation factor is introduced in the ML process to compensate the difference between the energy calculated by the QM and the energy generated by the optimized force field, while maintaining the local shape of the QM energy surface. The steps and procedures of this method are more complicated and not practical [3]. The use of deep learning technology can optimize the processing of unbalanced data sets, but it lacks accurate classification of unbalanced data sets.

The innovations of this article are proposing (1) an unbalanced data set training algorithm that maximizes $F1$ value for processing of unbalanced data sets based on ML; (2) an unbalanced data set based on NIBoost algorithm Classification methods that are used to process unbalanced data sets based on ML.

# 2 Strategies for processing methods of unbalanced data sets based on ML

## 2.1 Unbalanced data set classification problem

At present, various fields such as industry, commerce, and scientific research are applying the theories and methods of data mining to solve various problems. However, with the deepening of data problems, how to classify imbalanced data sets has become another urgent need to solve the problem. Unbalanced data set, that is, the amount of data of one type in the sample data set, is much larger than that of other types. For example, when the data type is only divided into two types, the type with a small number of samples is a positive type (minority type), and the number of samples is a negative type: class (majority class) [4]. The traditional classification method is inaccurate for the positive class when the data class distribution is unbalanced; if the imbalance is serious, the effect is worse; but the accuracy of the negative class classification is very high [5]. In life, the problem of unbalanced data can

be seen everywhere, such as lottery winning, computer firewall blocking Trojans, weather forecasting, and elevator failure detection. They all have one thing in common, that is, taking positive information as the focus of attention [6].

The following two aspects are mainly studied for the classification of imbalanced data sets:

(1) Data preprocessing: reconstruct the unbalanced data set by sampling and other methods, reduce the unbalance degree of the unbalanced data set, and meet the data requirements of traditional classification algorithms [7].

Sampling algorithms are popular in the fields of data mining and statistics. In data mining, sampling is a commonly used analysis method in the data preprocessing stage, mainly to avoid many problems caused by excessive data in the context of big data. The sampling method selects key research objects. On the one hand, it compresses the original data set, and on the other hand, it reduces the burden of excessive data volume for subsequent research. It is often used for extraction research before the algorithm or selected after processing the data set. Typical samples are used for final analysis and improvement [8]. If the sample is representative, it should have roughly the same characteristics as the original data set. If the average value of the data object is the feature to be studied, and the value of the sample is similar to the average value of the original data set, the sample is representative. Since sampling is a statistical process, the representation of specific samples is different; therefore, the best sampling plan is to select a representative sample to ensure a good probability [9].

(2) Algorithm: By integrating, increasing probability factors, or improving traditional classification algorithm, focus on the classification of positive classes in unbalanced data sets to improve the performance of positive class classification [10].

The classification of unbalanced data sets poses a challenge to prediction modeling, because most ML algorithms used for classification are designed based on the assumption that the number of samples in each category is equal. Classification of unbalanced data sets can improve the ability to solve practical problems.

## 2.2 Unbalanced data set classification algorithm

### 2.2.1 Decision tree data classification algorithm

There are many kinds of ML algorithms, and the decision tree algorithm is one of its classic algorithms. Decision tree is an important part of data mining classification methods, a kind of ML technology, mainly used to explore internal data rules and predictive analysis of new samples [11]. Because its basic algorithm is relatively mature, it has been adopted by various intelligent decision-making systems very early. Because the decision tree algorithm has good data analysis effects, and the decision tree can provide simple and intuitive result analysis, it has been widely used in the field of data mining [12].

Decision tree algorithm combines the classification process of tree shape and adaptation problem. There is a shared attribute in the specified tree; that is, each layer corresponds to a classification attribute [13]. The nodes in the layer have different attribute values, and the corresponding data in the attribute values are stored in the nodes. Each node stores the probability distribution of different types of label attributes on the branch line. Suppose that the current node is $v$, the training data set to $v$ is $L$, there are $k$ different class labels $C_i(i = 1, 2,\ldots,k)$, let $C_{i,L}$ be the tuple set of class label $C_i$ in $L$, $|L|$ and $|C_{i,L}|$ are the number of tuples, and a certain division will $L$ is divided into y subsets $\{L_1, L_2,\ldots,L_y\}$ [14,15].

#### 2.2.1.1 Information gain

The information gained on a certain division attribute $N$ is defined as the difference between the amount of information (entropy) needed to identify tuples before division and the amount of information needed to identify tuples after division on attribute $N$ [16]. There is the following relationship:

$$\text{InfoGain}(N) = \text{Info}(L) - \text{Info}_N(L). \tag{1}$$

In formula (1), $\text{Info}(L)$ represents the initial information entropy of the unbalanced data set.

$$\text{Info}(L) = -\sum_{i=1}^{k} p_i \log_2(p_i), \tag{2}$$

$$p_i = P(t \in C_i | \forall t \in L), \tag{3}$$

$$\text{Info}_N(D) = \sum_{j=1}^{y} \frac{|L_j|}{|L|} \times \text{Info}(L_j). \tag{4}$$

#### 2.2.1.2 Gain rate

The information gain metric tends to use the attribute partition with more branches, and the gain rate metric is adopted, which uses the split information value to normalize the information gain [17]. The definition formula of split information is as follows:

$$\text{SplitInfo}_N = -\sum_{j=1}^{y} \frac{|L_j|}{|L|} \times \log_2\left(\frac{|L_j|}{|L|}\right). \tag{5}$$

The gain rate is defined as:

$$\text{GainRatio}(N) = \frac{\text{InfoGain}(N)}{\text{SplitInfo}(N)}. \tag{6}$$

In formula (6), GainRatio($N$) represents the information gain rate of the unbalanced data set on the specific partition attribute $N$.

#### 2.2.1.3 Gini indicator

The Gini index is the measurement criterion used in the CART algorithm. The Gini index measures the impurity of the data partition or the training tuple set L [18]. Its definition formula is:

$$\text{Gini}(L) = 1 - \sum_{i=1}^{k} p_i^2, \tag{7}$$

$$p_i = p(t \in C_i | \forall t \in L). \tag{8}$$

The Gini index considers the binary division of each attribute. Assuming that $L$ is divided into $L_1$ and $L_2$ for a certain binary division of attribute $N$ on $L$, the Gini index of this division is defined as:

$$\text{Gini}_N(L) = \frac{|L_1|}{|L|}\text{Gini}(L_1) + \frac{|L_2|}{|L|}\text{Gini}(L_2). \tag{9}$$

The decrease in impurity due to this division is defined as:

$$\Delta\text{Gini}(N) = \text{Gini}(L) - \text{Gini}_N(L). \tag{10}$$

Each time the attribute that can maximize the reduction of impurity is selected as the split attribute, this attribute and its split subset (discrete value attribute) or split point (continuous value attribute) together form the split criterion [19,20].

The decision tree is a decision analysis method that calculates the probability that the expected value of the net present value is greater than or equal to zero, evaluates the project risk, and judges its feasibility by constructing a decision tree based on the known probability of occurrence of various situations. The decision tree can realize the classification of unbalanced data sets by analyzing the probability of different data sets.

### 2.2.2 Several main algorithms of decision trees

ID3.0 algorithm is the first decision tree algorithm. The algorithm requires eigenvalues to be discrete eigenvalues. The ID3.0 algorithm cannot handle continuous variables. In response to this problem, Quinlan extended the ID3.0 algorithm and proposed C4.5 [21]. The algorithm uses the percentage of information gain as the criterion for selecting variables. The C4.5 algorithm has its own advantages in dealing with missing values and uncertainties [22].

ID3.0 algorithm uses information gain as the selection criterion of sample features. The basic principle of the algorithm is as follows: calculate the information gain of all attributes on the root node, select the variable with the highest value as the segmentation variable, and then create industries based on the different values of the segmentation variable. This method is used to conduct retrospective research on subsequent nodes until the development of each review process is over, and all sample feature values are obtained [23].

The C4.5 algorithm is developed on the ID3.0 algorithm and is an extension and extension of the ID3.0 algorithm. C4.5 algorithm uses an information gain rate to select partition features, which overcomes the shortcomings of information gain selection, but the information gain rate has a preference for attributes with a small number of values.

### 2.2.3 C4.5 algorithm

The C4.5 algorithm is an improvement of the ID3.0 algorithm. It is based on the theory of information entropy. It takes the attribute with the largest information gain rate in the test sample as the test attribute and keeps judging the sample set until a tree is formed: complete decision tree. The advantage of this algorithm is that the generated classification rules are easy to understand and accurate. It can handle both discrete data and continuous data. It also introduces tree pruning technology to improve the accuracy of classification [24].

The algorithm assumes that the training data set $D$ is divided into $k$ categories: $D_1$, $D_2$,…, $D_k$. The total number of observations in the data set is $d$, $d_i$ is the number of observations in $D_i$, the probability that the sample belongs to the $i$th category is:

$$p_i = \frac{d_i}{d}, \quad (i = 1, 2, \ldots, k). \tag{11}$$

Information expectations for data set classification：

$$I = -\sum_{i=1}^{k} p_i \log_2 p_i. \tag{12}$$

Let $A$ be a certain attribute of the data set, and the value is $a_1$, $a_2$,…, $a_h$. The expected $I(A = a_j, j = 1, 2, \ldots, h)$ for each value is:

$$I(A = a_j) = -\sum_{i=1}^{k} p_{ij} \log_2 p_{ij}. \tag{13}$$

$d_j$ is the number of observations of $A = a_j$, and the information entropy of attribute $A$ is calculated as follows:

$$\text{Ent}(A) : p_i = \frac{d_j}{d}, \tag{14}$$

$$\text{Ent}(A) = \sum_{j=1}^{m} p_j * I(A = a_j). \tag{15}$$

The information gain of attribute $A$ is:

$$\text{Gain}(A) = \text{Ent}(A) - I. \tag{16}$$

The larger the Gain($A$) is, the larger the amount of classification information occupied by the attribute $A$ in the classification process [25]. If the maximum information gain is used as the criterion, it is more inclined to choose more classified attributes, so the information gain rate is used as the criterion for selecting nodes [26]. The information gain rate of attribute $A$ is:

$$\text{Gain}(A) - \text{Ratio}(A) = \frac{\text{Gain}(A)}{I(A = a_j)}. \tag{17}$$

The method part of this article uses the above method to study the processing method of an unbalanced data set based on ML, and the specific process is shown in Figure 1.

In Figure 1, the process of processing unbalanced data sets by ML is described. After data preprocessing, multiple decision tree algorithms are used to classify the unbalanced data sets.

# 3 Experiment on processing method of unbalanced data set based on ML

## 3.1 Use imbalanced data set training algorithm to maximize *F*1 value

Traditional classification algorithms cannot be directly applied to the problem of unbalanced sample classification. For classic unbalanced sample classification algorithms, the general idea is very intuitive, and most of them are directly applied to the data set, whether it is directly changing the samples in the data set or the weight distribution of the data set is to add or delete various samples in the data set through a special method. The classic unbalanced sample classification algorithms mainly include ID3 algorithm and naive Bayesian classifier. Finally, it is to make the original unbalanced data set achieve a relatively balanced state and finally can apply traditional classification methods to Solve the original problem [27].

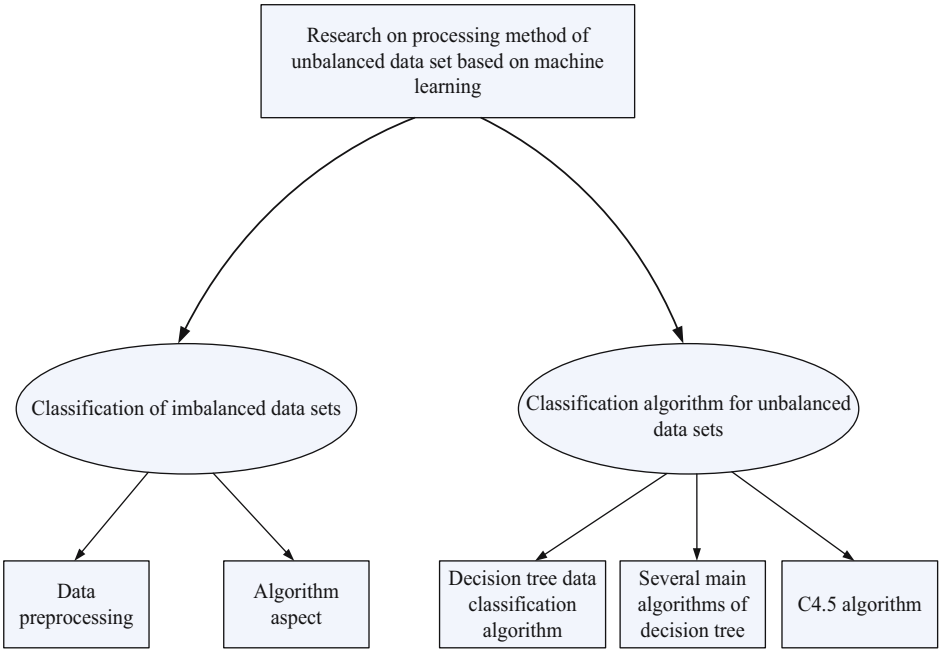


**Figure 1:** Part of the technical process of this method.

As we all know, the basic idea of traditional classification methods is to summarize the consistency hypothesis of the training sample space, which is to induce bias, so that the samples that have not appeared in the entire sample space can be predicted. For different classifiers, it is the VC dimension (Vapnik Chervonenkis Dimension) function they use, and the bias conditions are different [28]. Unfortunately, the classic imbalanced classification algorithm happens to be at odds with the idea of traditional ML. Because the classic imbalanced classification algorithm mostly solves the imbalance problem by changing the original sample distribution, while the traditional ML algorithm is based on training data and real data. It is based on independent and identical distribution. If the distribution of the training data is changed, it is entirely possible to have an unknown effect on the results [29]. Although this kind of influence on some discriminant models is not up to the level that can destroy the effect of the entire model, it is certain that it will definitely affect the decision-making process of the model [30]. Moreover, it is usually impossible to judge whether this effect is biased toward the good direction or the bad direction for the final prediction process of the real space, especially for some algorithms that involve random processes, such as the SMOAT algorithm, which changes the distribution more seriously. Even if the cross-validation method is used to train the model, the average classification accuracy or classification $F1$ value in multiple trials will fluctuate relatively greatly. Therefore, in order to solve the above problems, this article designed a method to directly train the model by targeting the evaluation criteria and achieved results that are equivalent to or better than the classic imbalance classification algorithm on most data sets.

## 3.2 Classification method of unbalanced data set based on NIBoost algorithm

Most of the traditional classification algorithms assume that the misclassification cost is the same and the ultimate goal is to improve the classification accuracy of the classifier. Therefore, when dealing with the classification problem of unbalanced data sets, they usually divide the minority class samples into the majority class and then improve the classification accuracy of the classifier. However, in the classification of unbalanced data sets, the correct classification of minority samples is often more important than the majority of samples. Sensitive cost learning is based on the above theory and gives a small number of samples of higher cost misclassification, which are not correctly classified. Based on the cost-sensitive theory, this article combines the ideas of RareBoost algorithm and GMBoost algorithm and proposes an unbalanced NIBoost data classification algorithm idea and oversampling technology. This article introduces the basic principles of the NIBoost algorithm and analyzes the classification efficiency and accuracy of the algorithm through experiments.

### 3.2.1 Basic principles and algorithms of NIBoost

This module proposes an unbalanced data classification algorithm that combines cost-sensitive ideas with NIBoost oversampling technology. The basic idea is to first use the geometric mean of the large-order sample error rate and the decimal-order sample error rate and use the GMBoost algorithm as the classification criterion; second, integrate the NKSMOTE algorithm proposed in Chapter 3 into each repetition, that is, adding a few samples Balance the data set and training the classifier in the data set; then different weight adjustments are performed according to the sample classification results and the original classification criteria. Therefore, the algorithm is divided into the following four stages:

1) Initialize the weights of the samples in the unbalanced data set $S$.
2) Call the oversampling algorithm (NKSMOTE) and add a minority class and original data set to form a training set.
3) Train a weak classifier, calculate the $e_t$ and other correlation values of the classifier, and update the weights of samples whose classification results are positive and those whose classification results are negative.
4) Calculate the correlation value $e_t$ of the classifier, when $e_t$ is less than or equal to 0.5, repeat the second and third stages, and terminate the cycle if $e_t$ is greater than 0.5.

**Table 1:** Experimental steps in this article

| Experimental research on the processing method of unbalanced data set based on ML | | | | | | | |
|---|---|---|---|---|---|---|---|
| 3.1 | Training algorithm for imbalanced data sets by maximizing $F1$ value | | | 3.2 | Classification method of unbalanced data set based on NIBoost algorithm | | |
| 1 | VC dimensional function | 2 | Smoat algorithm | 1 | Basic principle and algorithm of NIBoost | 2 | Weight update |

**Table 2:** Comparison results of algorithm maximization *F*1 value

| Data set | IB | *K*-means | DSIB | NCut | McIB | Comparison results |
|---|---|---|---|---|---|---|
| Abalone5 | 0.4127 | 0.4231 | 0.4629 | 0.4336 | 0.4813 | Better |
| Balance scale | 0.6726 | 0.6309 | 0.6931 | 0.6412 | 0.6746 | Better |
| Iris | 0.8307 | 0.8049 | 0.8647 | 0.5679 | 0.7431 | Quite |
| Wine | 0.7521 | 0.7122 | 0.7540 | 0.5927 | 0.7963 | Better |
| Yeast | 0.4628 | 0.4116 | 0.4567 | 0.4231 | 0.4971 | Quite |

### 3.2.2 Weight update

The algorithm in this section mainly updates the weights of the samples to make the classifier pay more attention to the samples that are easily misclassified. The NIBoost algorithm contains two weight update steps. The first time was because the oversampling algorithm was incorporated in the iterative process, which changed the number of samples in the unbalanced data set, so it was necessary to assign weights to the synthetic minority samples, and the weights of the original samples had to be changed. The second time is after the weak classifier is called, the sample weights need to be updated according to the classification results, so that the samples that are difficult to classify will get more attention in the next iteration.

This part of the experiment proposes that the above steps are used for the research experiment on the processing method of the unbalanced data set based on ML. The specific process is shown in Table 1.

## 4 Processing methods of unbalanced data sets based on ML

### 4.1 Experimental analysis of unbalanced data set based on maximizing *F*1 value algorithm

(1) A balanced data set (such as the Iris data set) is added to this experiment. We hope that this algorithm can not only handle unbalanced data but also apply to

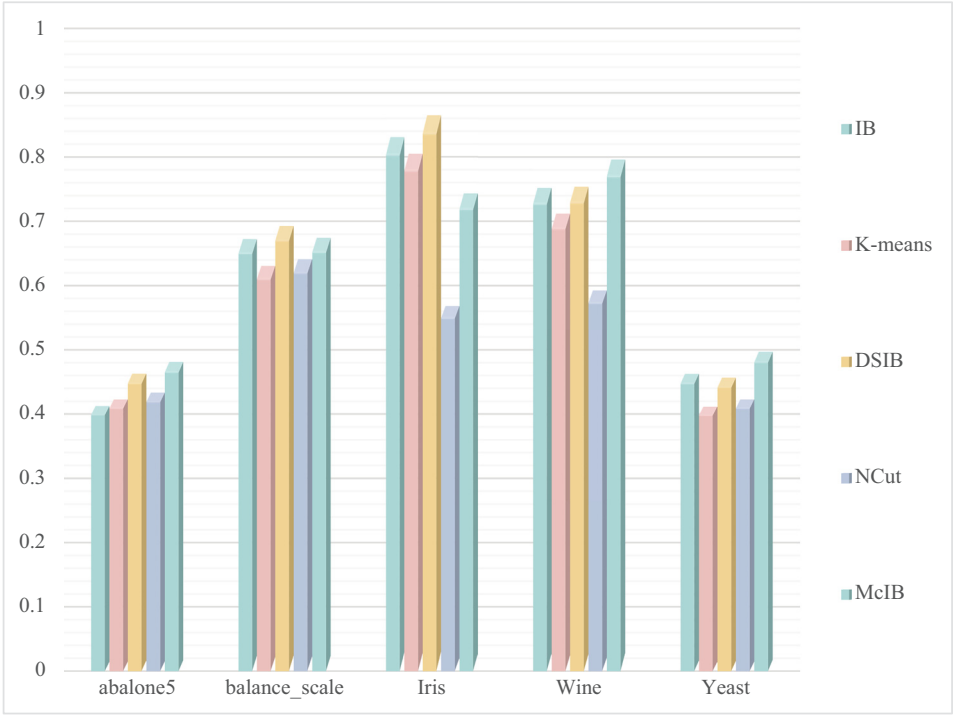


**Figure 2:** Comparison result of algorithm maximizing *F*1 value.

**Table 3:** Performance comparison between the algorithm to maximize *F*1 value and the other three algorithms

| Evaluating indicator | WELM | ADASYN-SVM | ID 3.0 | Maximize *F*1 value |
|---|---|---|---|---|
| *G*-Mean | 0.8612 | 0.7435 | 0.9027 | 0.9874 |
| *F*-Measure | 0.5637 | 0.6248 | 0.7639 | 0.9546 |
| AUC | 0.6742 | 0.7231 | 0.8459 | 0.9673 |

balanced data. The statistical algorithm maximizes the comparison result of the *F*1 value and gives the comparison result of the average performance, as shown in Table 2 and Figure 2. Among them, "Better" means that the algorithm for maximizing *F*1 value is better than other comparison algorithms in the performance of the table; "Quite" means that the algorithm for maximizing *F*1 value does not show obvious advantages compared with other algorithm, and the performance is similar.

It can be seen from the chart that the overall performance of the algorithm for maximizing *F*1 value on unbalanced data is better than other clustering analysis methods that are not specifically for unbalanced data sets; compared with the original IB algorithm and *K*-means algorithm, the maximum *F*1 value is the algorithm can also effectively process balanced data sets. Compared with the DSIB algorithm, the algorithm for maximizing the *F*1 value will not have large fluctuations, and the overall performance is relatively stable.

(2) When the proportion of the marked target domain data is 15%, the first batch of data is taken as the source domain data, and the second to tenth batches of data are respectively used as the target domain data. The classification performance of the *F*1 value is compared with the other three algorithms, including weighted extreme learning machine (WELM), support vector machine with ADASYN sampling strategy (ADASYN-SVM), and ID 3.0 algorithm. Statistically sort the specific results and draw them into charts, as shown in Table 3 and Figure 3.

It can be seen from the chart that the TWELM algorithm proposed in this article has better classification results than other unbalanced learning algorithms in the three evaluation indicators of *G*-mean, *F*-measure, and AUC, and the performance is more stable. The algorithm that maximizes the *F*1 value can get the best results most of the time. Compared with the two unbalanced classification algorithms, the proposed maximum *F*1 value algorithm can make full use of the source domain information to solve the target domain classification problem. Compared with the other three algorithms, the algorithm to maximize *F*1 value can get more stable results. In the experiment, the first batch is used as the source domain data. As the sensor drifts, subsequent batches will contain more knowledge information that is different from the first batch. Therefore, the classification performance of the other three comparison algorithms will be relatively poor. However, the algorithm that maximizes the *F*1 value can always maintain good classification accuracy.

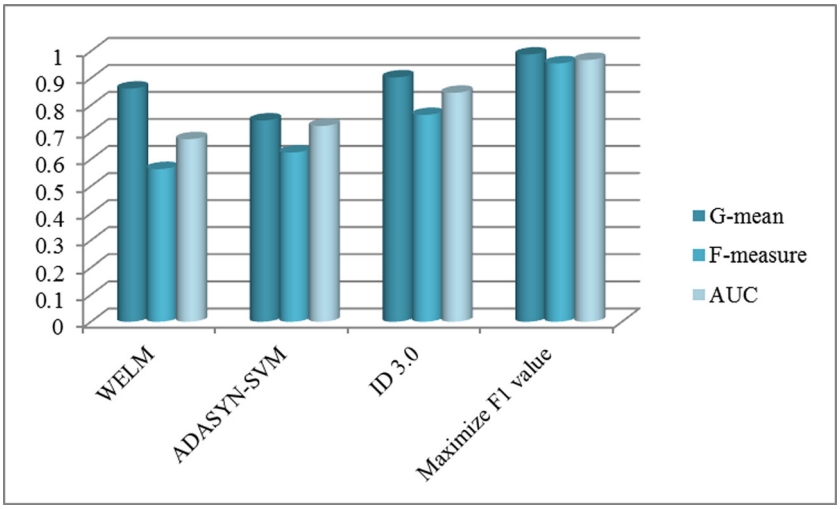


**Figure 3:** Performance comparison between the algorithm to maximize *F*1 value and the other three algorithms.

**Table 4:** Comparison of experimental results

| | Bagging (%) | AdaBoost (%) | ID 3.0 (%) | C4.5 (%) |
|---|---|---|---|---|
| Piam_Indians-Gm | 76.31 | 62.24 | 78.64 | 89.42 |
| Haberman-Gm | 67.42 | 71.09 | 76.41 | 91.07 |
| Transfusion-Gm | 59.67 | 63.25 | 71.13 | 88.64 |
| Yeast5-Gm | 76.13 | 82.27 | 86.42 | 94.75 |
| Abalone15-Gm | 83.15 | 86.74 | 81.04 | 90.46 |
| Abalone19-Gm | 65.34 | 72.29 | 78.47 | 93.06 |

## 4.2 Experimental analysis of unbalanced data set based on C4.5 algorithm

(1) When the ID 3.0 algorithm processes imbalanced data sets, the classification accuracy of positive samples is low. This is because the classification hyperplane is biased toward minority samples when the classifier processes imbalanced data sets, which is easy to misclassify minority samples as the majority sample. Although the negative samples have a relatively high accuracy rate, the Gm value is relatively low. The decision tree C4.5 algorithm is compared with ID 3.0 algorithm, AdaBoost algorithm, and Bagging algorithm. The experimental results are shown in Table 4 and Figure 4.

It can be seen from Table 4 and Figure 4 that the algorithm proposed in this article has better Gm values on the six unbalanced data sets than the other three algorithms. Specifically, on the Piam_Indians, Abalone15, and Abalone19 data sets, the Gm value obtained by the C4.5 algorithm is higher than the Gm value obtained by the ID 3.0 algorithm, AdaBoost algorithm, and Bagging algorithm.

(2) Taking into account the limitations of computer and R statistical software, the sample size of the six data

**Table 5:** Accuracy of positive samples

| | Bagging (%) | AdaBoost (%) | ID 3.0 (%) | C4.5 (%) |
|---|---|---|---|---|
| Data set 1 | 78.46 | 72.98 | 69.45 | 89.96 |
| Data set 2 | 74.31 | 78.12 | 71.53 | 92.08 |
| Data set 3 | 79.61 | 82.11 | 76.26 | 95.16 |
| Data set 4 | 69.42 | 76.43 | 77.42 | 96.03 |
| Data set 5 | 82.06 | 80.15 | 80.67 | 98.45 |
| Data set 6 | 75.63 | 77.49 | 85.14 | 97.12 |
| Average | 76.58 | 77.88 | 76.75 | 94.80 |

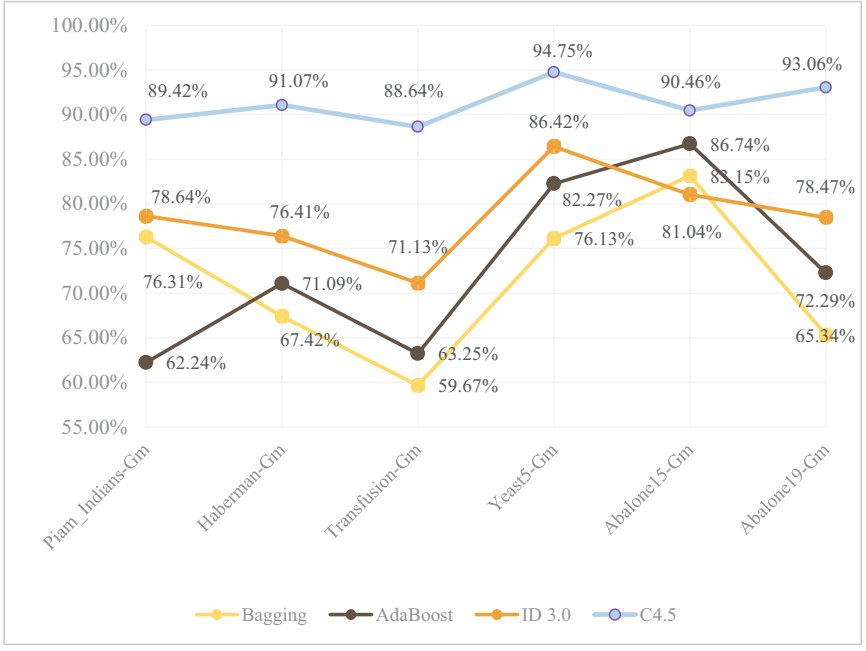


**Figure 4:** Comparison of experimental results.

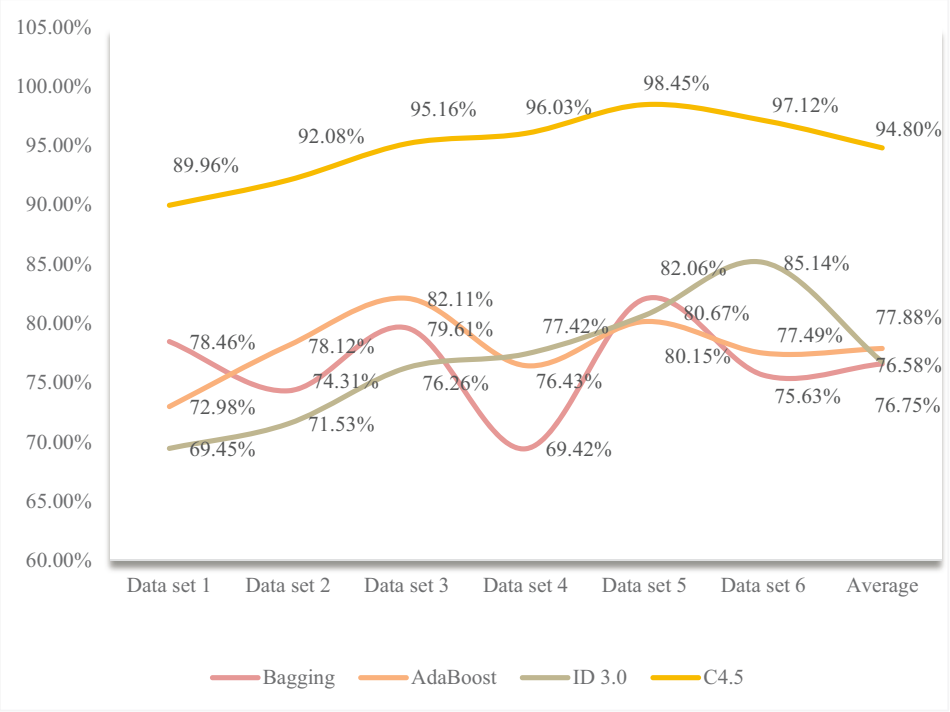


**Figure 5:** Accuracy of positive samples.

**Table 6:** Experimental results of unbalanced data set

| | Abalone19 | Glass4 | Pima | Haberman |
|---|---|---|---|---|
| Boosting | 0.4631 | 0.5139 | 0.6142 | 0.6511 |
| RUSBoost | 0.4507 | 0.5674 | 0.6325 | 0.6742 |
| Adaboost | 0.4829 | 0.5912 | 0.6746 | 0.7031 |
| NIBoost | 0.5261 | 0.6341 | 0.6957 | 0.8542 |

sets used for modeling is chosen to be about 5000. To judge the quality of a model, it is usually measured by the accuracy of the model. Statistical analysis of the experimental results obtained by using C4.5 algorithm, ID 3.0 algorithm, AdaBoost algorithm, and Bagging algorithm on R on the six data sets, and the results obtained by modeling on Clementine

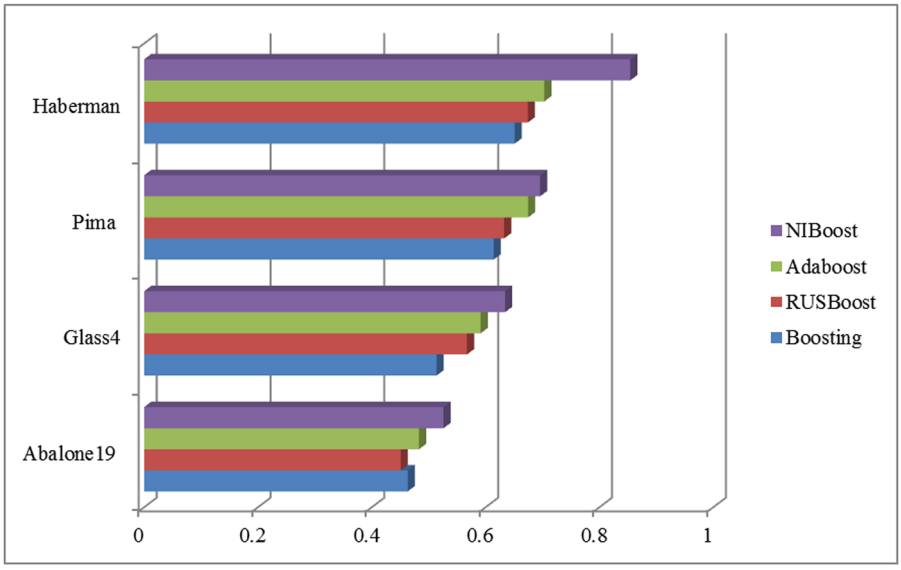


**Figure 6:** Experimental results of unbalanced data set.

software using decision tree algorithm, as shown in Table 5 and Figure 5.

It can be seen from the chart data that the accuracy of the positive sample classification is generally very high. The average positive sample classification accuracy of the Bagging algorithm is as high as 76.58%, and the ID 3.0 algorithm and the AdaBoost algorithm have an average positive sample classification accuracy of 76.75 and 77.88%, respectively, and the C4.5 algorithm gets 94.80% accuracy.

### 4.3 Experimental analysis of unbalanced data set based on NIBoost Algorithm

Using sample subset sampling, a sparse sampling technique is used to process the experimental data set, and then for each selected different data set, the unbalanced sample data set is classified by the integrated learning algorithm. The performance of the data sampling algorithm and classifier in the unbalanced data set is measured according to the evaluation index of the classification on the unbalanced data set. The classic NIBoost algorithm is compared in the field of sample data sampling. The algorithm is run 200 times on the data set in the experiment and averaged, and the corresponding values are obtained, as shown in Table 6 and Figure 6.

The AUC results of the data set used in this experiment are compared with the results of the previous algorithms. The integrated learning algorithm optimized for sample subsets used in the experiment is improved compared with the previous algorithms. Comparing the data in each row of the table, it can be concluded that the NIBoost algorithm is 0.0623 higher than the RUSBoost algorithm on the Pima data set. There is a significant improvement in the Haberman data set. On the Glass4 data set, the NIBoost algorithm in this article is 0.0667 higher than the RUSBoost algorithm. Using the NIBoost algorithm proposed in this article on the Abalone19 data set is 0.063 higher than the Boosting algorithm. In the above unbalanced data set, the NIBoost algorithm used in this article is basically better than Adaboost and random sparse sampling integrated learning algorithm.

## 5 Conclusions

In recent decades, with the gradual transformation of mechanical learning from academic research to application, some problems have never been encountered. Among them, the unbalanced data set is an unprecedented problem. Imbalanced classification of data sets refers to the classification problem when the number of samples of different categories in the data set is different. The key to solving this kind of problem is to measure the processing method of a computer algorithm.

Improving the classification accuracy of the classifier is the goal of traditional classification algorithm. Therefore, when using this algorithm to deal with imbalanced data sets, it is easier to divide the minority order samples into the majority order samples, and the accuracy of the minority order classification is not high, but these minority samples have greater value and importance than the majority samples. At the same time, the practice has proved that the cost of different sample types is basically different after misclassification.

This article mainly introduces the development environment and overall design of the unbalanced data set classification system. The designed unbalanced data set classification system mainly includes oversampling algorithm, classification algorithm, and parallel algorithm. The oversampling algorithm mainly realizes all the oversampling algorithms used in this article and can choose different traditional classification algorithms for classification. The further research work on classifier ensemble is to select the classifiers first and select the classifiers with good classification effects and great differences through the selection strategy. How to combine the selection strategy with the processing of unbalanced data sets needs further research.

**Funding information:** This work was supported by Key R&D projects of Sichuan Science and Technology Plan, No. 2022YFG0323; Key Research Project of Guangdong Baiyun College, No. 2022BYKYZ02; Key Research Platform of Guangdong Province, No. 2022GCZX009; Special project in key fields of colleges and universities in Guangdong province, No. 2020ZDZX3009.

**Author contributions:** Qingwei Zhou and Yongjun Qi designed and performed the experiment and prepared this manuscript. Hailin Tan and Peng Wu helped to do the experiment. All coauthors contributed to manuscript editing. All authors have read and agreed to the published version of the manuscript.

**Conflict of interest:** The authors state that this article has no conflict of interest.

**Ethical approval:** This article does not contain any studies with human participants performed by any of the authors.

**Data availability statement:** Data sharing is not applicable to this article as no new data were created or analyzed in this study.

## References


[1] A. Vollant, G. Balarac, and C. Corre, "Subgrid-scale scalar flux modelling based on optimal estimation theory and machine-learning procedures," *J. Turbul.*, vol. 18, no. 9, pp. 1–25, 2017.

[2] T. Hunt, C. Song, R. Shokri, V. Shmatikov and E. Witchel, "Privacy-preserving machine learning as a service," *Proc. Priv. Enhancing Technol.*, vol. 2018, no. 3, pp. 123–142, 2018.

[3] Y. Li, H. Li, F. C. Pickard, B. Narayanan, F. Sen, M. K. Y. Chan, et al. "Machine learning force field parameters from Ab initio data," *J. Chem. Theory Comput.*, vol. 13, no. 9. pp. 4492–4503, 2017.

[4] A. Karpatne, Z. Jiang, R. R. Vatsavai, S. Shekhar and V. Kumar, "Monitoring land-cover changes: A machine-learning perspective," *IEEE Geosci. Remote. Sens. Mag.*, vol. 4, no. 2. pp. 8–21, 2016.

[5] P. Plawiak, T. Sosnicki, M. Niedzwiecki, Z. Tabor, and K. Rzecki, "Hand body language gesture recognition based on signals from specialized glove and machine learning algorithms," *IEEE Trans. Ind. Inform.*, vol. 12, no. 3. pp. 1104–1113, 2016.

[6] W. Yuan, K. S. Chin, M. Hua, G. Dong, and C. Wang, "Shape classification of wear particles by image boundary analysis using machine learning algorithms," *Mech. Syst. Signal. Process*, vol. 72–73, pp. 346–358, 2016.

[7] M. E. Dickson and G. L. W. Perry, "Identifying the controls on coastal cliff landslides using machine-learning approaches," *Environ. Model. & Softw.*, vol. 76, no. Feb, pp. 117–127, 2016.

[8] G. Wang, M. Kalra, and C. G. Orton, "Machine learning will transform radiology significantly within the next 5 years," *Med. Phys.*, vol. 44, no. 6. pp. 2041–2044, 2017.

[9] Y. Huang, C. L. Gutterman, P. Samadi, P. B. Cho, W. Samoud, C. Ware, et al., "Dynamic mitigation of EDFA power excursions with machine learning," *Opt. Express*, vol. 25, no. 3. pp. 2245–2258, 2017.

[10] T. Liu, Y. Yang, G. B. Huang, K. Y. Yong, and Z. Lin, "Driver distraction detection using semi-supervised machine learning," *IEEE Trans. Intell. TransportatiSyst.*, vol. 17, no. 4. pp. 1108–1120, 2016.

[11] E. E. Tripoliti, T. G. Papadopoulos, G. S. Karanasiou, K. K. Naka, and D. I. Fotiadis, "Heart failure: Diagnosis, severity estimation and prediction of adverse events through machine learning techniques," *Computat. Struct. Biotechnol. J.*, vol. 15, no. C. pp. 26–47, 2017.

[12] J. A. Gonzalez, L. A. Cheah, A. M. Gomez, P. D. Green, and E. Holdsworth, "Direct speech reconstruction from articulatory sensor data by machine learning," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 25, no. 12. pp. 2362–2374, 2017.

[13] E. Giacoumidis, A. Matin, J. Wei, N. J. Doran, L. P. Barry, and X. Wang, "Blind nonlinearity equalization by machine-learning-based clustering for single- and multichannel coherent optical OFDM," *J. Lightwave Technol.*, vol. 36, no. 3. pp. 721–727, 2018.

[14] A. Linden and P. R. Yarnold, "Combining machine learning and matching techniques to improve causal inference in program evaluation," *J. Eval. Clin. Pract.*, vol. 22, no. 6. pp. 864–870, 2016.

[15] J. K. Park, B. K. Kwon, J. H. Park, and D. J. Kang, "Machine learning-based imaging system for surface defect inspection," *Int. J. Precis. Eng. Manuf.-Green Technol.*, vol. 3, no. 3. pp. 303–310, 2016.

[16] A. Kashyap, L. Han, R. Yus, J. Sleeman, T. Satyapanich, S. Gandhi, et al., "Robust semantic text similarity using LSA, machine learning, and linguistic resources," *Lang. Resour. Eval.*, vol. 50, no. 1. pp. 125–161, 2016.

[17] L. M. Eerikinen, J. Vanschoren, M. J. Rooijakkers, R. Vullings and R. M. Aarts, "Reduction of false arrhythmia alarms using signal selection and machine learning," *Phys. Meas.*, vol. 37, no. 8. pp. 1204–1216, 2016.

[18] B. Long, K. Yu, and J. Qin, "Data augmentation for unbalanced face recognition training sets," *Neurocomputing*, vol. 235, no. APR.26. pp. 10–14, 2017.

[19] D. Yu and X. Zi-Qiang, "Prediction of damage to insulation joints based on SVM with unbalanced data sets," *Int. J. Multimed. Ubiquitous Eng.*, vol. 11, no. 3. pp. 273–282, 2016.

[20] A. Werner, G. Olaf, G. Asma, K. H. Folkert, K. Zardad and L. Berthold, "Ensemble pruning for glaucoma detection in an unbalanced data set," *Methods Inf. Med.*, vol. 55, no. 6. pp. 557–563, 2016.

[21] Z. Liang, X. Li, and W. Song, "Research on speech emotion recognition algorithm for unbalanced data set," *J. Intell. Fuzzy Syst.*, vol. 5, pp. 1–6, 2020.

[22] L. Sánchez-Guerrero, J. F. González, B. A. González-Beltrán, and S. B. González-Brambila, "Evaluating predictive techniques in educational data mining: An unbalanced data set case of study," *Res. Comput. Sci.*, vol. 148, no. 3. pp. 49–60, 2019.

[23] A. Den Reijer and A. Johansson, "Nowcasting Swedish GDP with a large and unbalanced data set," *Empir. Econ.*, vol. 57, no. 4. pp. 1351–1373, 2019.

[24] R. Jing-Shi, P. Hai-Wei, L. Peng-Yuan, G. Lin-Lin, H. Qi-Long, Z. Zhi-Qiang, et al., "Symmetry theory based classification algorithm in brain computed tomography image database," *J. Med. Imaging Health Inform.*, vol. 6, no. 1. pp. 22–33, 2016.

[25] J. Cao, W. Huang, T. Zhao, J. Wang, and R. Wang, "An enhance excavation equipments classification algorithm based on acoustic spectrum dynamic feature," *Multidimension. Syst. Signal. Process.*, vol. 28, no. 3. pp. 921–943, 2017.

[26] A. Palacios, L. Sanchez, I. Couso, and S. Destercke, "An extension of the FURIA classification algorithm to low quality data through fuzzy rankings and its application to the early diagnosis of dyslexia," *Neurocomputing*, vol. 176, no. Feb. 2, pp. 60–71, 2016.

[27] C. G. Yan, X. D. Wang, X. N. Zuo, and Y. F. Zang, "DPABI: Data processing & analysis for (Resting-State) brain imaging," *Neuroinformatics*, vol. 14, no. 3. pp. 339–351, 2016.

[28] C. Zhu, H. Wang, X. Liu, S. Lei, L. T. Yang, and V. C. M. Leung, "A novel sensory data processing framework to integrate sensor networks with mobile cloud," *IEEE Syst. J.*, vol. 10, no. 3. pp. 1125–1136, 2016.

[29] R. Munro, R. Lang, D. Klaes, G. Poli, C. Retscher, R. Lindstrot, et al., "The GOME-2 instrument on the Metop series of satellites: Instrument design, calibration, and level 1 data processing - An overview," *Atmos. Meas. Tech.*, vol. 9, no. 3. pp. 1279–1301, 2016.

[30] N. Corbin, E. Breton, M. de Mathelin, and Vappou J. "K-space data processing for magnetic resonance elastography (MRE)." *Magnetic Reson. Mater. Phys. Biol. Med.*, vol. 30, no. 2. pp. 1–11, 2017.