Journal of Network Intelligence ©2024 ISSN 2414-8105
Taiwan Ubiquitous Information Volume **9**, Number **3**, August **2024**

Unbalanced Big Data Classification based on Population Intelligence Optimized Machine Learning

An-Qing Zhu*

Management School,
South China Business College Guangdong, University of Foreign Studies,
Guangzhou 510000, P. R. China
204932@gwng.edu.cn

Hong-Yuan Li

School of Architectural Engineering,
Guangzhou City Construction College, Guangzhou, 510925, P. R. China
87651566@qq.com

Rui-Hui Wu

Faculty of Entrepreneurship and Business,
Universiti Malaysia Kelantan, Kuantan 16100, Malaysia
a20e0276f@siswa.umk.edu.my

*Corresponding author: An-Qing Zhu

Received July 21, 2023, revised November 19, 2023, accepted February 1, 2024.

Abstract. *The number of majority class samples in unbalanced data far exceeds the number of minority classes, making the data appear unbalanced. How to improve the recognition rate of minority classes in unbalanced data has become an urgent problem in machine learning. The Random Forest model is a machine learning algorithm with good robustness, but it has the problem of poor recognition rate when classifying unbalanced data. To solve this problem, this work proposes a weighted Random Forest model based on population intelligence optimisation. Firstly, each decision tree is multiplied by a weight proportional to its training accuracy at the time of voting. Secondly, to address the problem of difficult parameter selection, a whale swarm algorithm is adopted to iteratively optimise the parameters affecting the new model, so as to select the required parameters for the model. In addition, the inter-individual distance calculation and body movement rules of the WOA were improved in order to enhance the local search capability of the WOA and its convergence speed. The simulation was validated by Python software on four standard data sets from the UCI database. The simulation results show that the proposed model has high minority class recognition performance in unbalanced data classification. Compared with the standard Random Forest, the recall of the proposed model is improved by 11% at an unbalanced degree of 5%.*

Keywords: Big data; Random Forests; Whale swarm algorithm; Decision trees; Imbalanced data

1. **Introduction.** Data mining techniques are becoming an active and important research topic to be addressed today [1,2]. The goal of data mining is to be able to extract from a large amount of random data the relatively unknown and usable information that is meaningful to humans. In general, data mining can be seen as an integration of multiple fields of convergence [3,4,5]. There are many disciplines where data mining will highlight its strengths. For example, it is used particularly extensively in machine learning [6], in

the field of artificial intelligence [7], and in database technology [8], and has also yielded remarkable results.

The main techniques in the field of data mining contain three directions [8]: classification of data, clustering of data, and association rules for data. The main research in this work is classification mining. Classification techniques focus on the construction of classifiers from the dataset being trained. Among the many classification techniques, the one with a relatively uncomplicated single classifier structure is the decision tree algorithm [9]. However, big data presents characteristics such as large number and imbalance, i.e. the number of samples of one class is much more than the number of samples of another class. How to mine valuable data from the data is what we should focus on. Currently, there are many problems that need to be solved in data mining, such as multi-class classification problems, dichotomous classification problems, misclassification costs, class overlap, meaningless disjoint and imbalanced data. Classification mining problems related to unbalanced data deserve even more attention.

Many real-life applications are filled with a variety of categories of unbalanced data. In case data for disease diagnosis prediction [10], for example, the number of samples for important and rare diseases is much smaller than the number of samples for normal and common diseases. In the Internet sample data for intrusion detection [11], the number of regular samples is much larger than the number of samples we need for intrusions. Since the number of majority class samples is much higher than the number of minority class samples, applying conventional classifiers directly to these sample class imbalance scenarios would lead to unsatisfactory classification results. Therefore, it is an important issue how to effectively identify the minority class in category imbalanced data. Various solutions have been proposed in order to solve the problems related to data imbalance [12,13]. However, the multi-category imbalance problem is difficult to solve.

To address these issues, two categories can be distinguished according to their treatment: the first is external or data-level methods [14], where the data are pre-processed to rebalance the class distribution in order to reduce the heterogeneous distribution effect in the classification process. The second category is internal or algorithm-level approaches. Machine learning as well as other algorithms, such as association classification algorithms [15] and K-nearest neighbour algorithms [16], also play an important role in unbalanced data classification.

There is now a growing awareness of the limitations and drawbacks of single classifiers in practice. The classification performance of single classifiers is gradually failing to meet the demanding requirements of today's data information classification. As a result, a cascade approach to single classifiers has emerged. Cascading methods use two or more classifiers to train on data information, and finally combine the outputs of multiple classifiers to obtain the final output. Random Forests are the product of this environment.

Although Random Forest models [17] have been used on such a large scale in numerous fields, their voting selection mechanism can lead to some decision trees with lower training accuracy having the same voting power, thus reducing the voting accuracy. Moreover, the number of decision trees and other parameters selected in a Random Forest model usually have a large impact on the final classification results. These problems can seriously affect the accuracy of unbalanced data classification, therefore, this work makes a series of improvements to the traditional Random Forest algorithm and achieves good results.

1.1. **Related Work.**

Methods for classifying non-equilibrium data can be divided into two categories: methods based on pre-processing of the dataset, and methods based on the algorithm level.

The data-level approach focuses on equalising the data using resampling techniques, mainly consisting of oversampling and undersampling. Oversampling balances the categories by adding a small number of class samples, but does not add any valid new information. In contrast to oversampling, the undersampling method balances the data by reducing the number of samples in the majority class, but inevitably loses some valid information. In addition to the most basic random undersampling and random oversampling, the Synthetic Minority Over-Sampling Technique (SMOTE) is a more common method that reduces the degree of imbalance in a dataset by interpolating between sample points to generate a minority class of data. Fernández et al. [18] proposed an improved SMOTE algorithm that removes noisy data and discrete points to obtain higher data classification performance. Jiang et al. [19] proposed a majority weighted minority oversampling technique that effectively selects minority classes of samples that are not easily learned and assigns them appropriate weights. However, the training computation of the classifier will be greatly increased if the amount of data is large, and important samples are easily lost. Thus the ability to handle unbalanced data by data-level methods is limited.

Random Forest is an integrated learning algorithm that classifies classes of test samples by combining the classification results of several single classifiers. The algorithm has better classification results and generalisation ability compared to individual classifiers. Currently, researchers have proposed optimisation and improvement methods in different stages of Random Forests. In the data processing stage, Luo et al. [20] proposed a probabilistic Random Forest model, which improves the classification accuracy of the integrated classifier by treating features and labels as probability distribution functions. In the feature selection phase, Zhou et al. [21] proposed a feature cost-sensitive Random Forest model, which prefers features with high cost, i.e., features with higher classification accuracy, when selecting features. In the integrated voting stage, Adnan and Islamet [22] combined weighted Random Forest and balanced Random Forest, which can improve the recognition accuracy of a few classes while ensuring the overall recognition accuracy. In the final sample classification of Random Forests, Charbuty and Abdulazeez [23] proposed decision trees that make the classification results more inclined towards high classification accuracy. However, while these methods can have high overall recognition accuracy when the data is highly unbalanced, the recognition accuracy for the minority classes is not satisfactory.

1.2. **Motivation and contribution.** The above analysis shows that the existing improved Random Forest models tend to favour the majority class decision trees in order to maximise the classification accuracy, thus ignoring the results of the minority class votes. Moreover, the number of decision trees and other parameters in the Random Forest model usually have a greater impact on the final classification results. Therefore, to address the above problems, a weighted Random Forest model based on population intelligence optimization is proposed in this work.

The main innovations and contributions of this work include:

(1) A new model, the accuracy-weighted Random Forest model, is conceived to address the drawbacks of traditional Random Forest models in terms of decision tree voting mechanisms. The key is to add the concept of weights to the traditional model so that the decision tree can take full advantage of its own strengths when voting on the classification results.

(2) As a new type of swarm intelligence optimization algorithm, the Whale-swarm optimization algorithm (WOA) has the features of setting fewer parameters, simple computational process and easy implementation [24], and has shown better performance in global and local parameter search and optimization solution tasks. To enhance the recognition

performance of the Random Forest approach for a few classes, the suggested technique is then parameter optimized using WOA on top of the weighted model.

(3) The inter-individual distance calculation and body movement rules of WOA were improved in order to enhance the local search capability of WOA and its convergence speed.

2. Principles of Random Forests.

2.1. Bagging algorithm.

A Random Forest's primary concept is to create a number of decision trees that are randomly unconnected and may each be trained separately using training samples. Each decision tree, which has its own vote, classifies fresh samples as they are introduced and communicates the results of the classification to the Random Forest.

The final categorization outcome of the sample is determined by the Random Forest using the grouping with the most votes. If the same training samples with the same attributes were used for each decision tree, the results would be identical and the voting mechanism would be meaningless. Therefore, in order to ensure that each decision tree is different, we need to obtain different training samples in some way, and the samples cannot be identical in terms of attributes, so we need to select a random number of attributes each time to generate the decision tree. All decision trees generated in this way would then be full of randomness and would allow different voting results for the samples.

In order to ensure that each decision tree can produce different classification results independently, the training samples and decision attributes used in generating each decision tree should also be different. The decision characteristics can be generated by choosing a random subspace algorithm, while the training samples are produced using the Bagging algorithm [25], which is a component of the Random Forest algorithm.

During the random sampling process, the weak classifier will have some samples that are not drawn, these samples are out-of-bag samples [26]. Considering that there are N total training samples, all samples are sampled with the same probability, so that the probability of each sample being drawn is $1/N$. All weak classifiers will be randomly drawn N times, and the probability that a sample is not drawn in any of the N times is P.

$$P = \left(1 - \frac{1}{N}\right)^n \tag{1}$$

When N in the above equation tends to infinity, the probability that a sample is never drawn is $1/e$, which is approximately 0.368, where e denotes the natural logarithm. About 36.8% of the total training samples are out-of-bag samples.

2.2. Stochastic subspace algorithms.

Another key to forming a Random Forest model is the random subspace algorithm.

The randomised subspace algorithm is characterised by randomly selecting a subset of features to train weak classifiers that are independent of each other. This is followed by a voting process to determine the final classification result of the samples to be tested. Similar to the Bagging algorithm, the random subspace algorithm also constructs different weak classifiers by sampling [27]. and by taking a voting mechanism to constitute a strong classifier after independent training. The difference between the two, however, is that one samples the sample space with a put-back to obtain training samples, while the other samples the features without a put-back. The advantages and disadvantages of the two will be compared in the following two aspects.

One of the parameters in the random subspace algorithm that needs to be taken manually is the number of randomly selected features at a time, m. The value of m has a very significant impact on the algorithm. If m is chosen too small, the classification accuracy of the weak classifier may be significantly reduced, thus affecting the classification accuracy of the strong classifier. However, if m is chosen too large, it will not be able to avoid the generation of redundant attributes. Although the impact of redundant attributes on the correct rate of the algorithm is reduced at the time of voting, the prolonged training time due to redundant attributes is indeed unavoidable. Therefore, it is also important to choose an appropriate value of m. The m value is chosen based on the total number of attributes M.

$$m = \lfloor \log_2(M+1) \rfloor , \tag{2}$$

where the symbol $\lfloor \cdot \rfloor$ indicates rounding down.

2.3. **Voting strategy.** Like the Bagging algorithm and the random subspace algorithm, the Random Forest algorithm also uses a voting strategy, and its voting process is expressed as follows:

$$f_{RF}(x) = \arg \max_{i=1,2,\dots,c} \{ I(f_i^{\text{tree}}(x) = i) \}, \tag{3}$$

where $f_{RF}(x)$ denotes the classification result of the Random Forest model for the sample x to be tested, $I(\cdot)$ denotes the number of expressions satisfying the brackets, $f_i^{\text{tree}}(x) = i$ denotes the output category of the l-th decision tree as I, and c denotes the number of categories in the whole Random Forest.

A decision tree is the fundamental building block of a Random Forest. Let there be a total of m categories $C_i (i = 1, 2, \dots, m)$ in the sample set S. s_i is the number of samples belonging to C_i, then the sample expectation entropy is calculated as shown as follow:

$$I(s_1, s_2, \dots, s_m) = -\sum_{i=1}^{m} \frac{s_i}{s} \log_2 \frac{s_i}{s}, \tag{4}$$

where s indicates the total number of samples and s_i indicates the number of samples belonging to the category C_i.

For a single feature A of the sample, the expected entropy of the sample feature is calculated as shown as follow:

$$E(A) = \sum_{j=1}^{k} \frac{s_{1j} + s_{2j} + \cdots + s_{mj}}{s} I(s_{1j}, s_{2j}, \dots, s_{mj}), \tag{5}$$

where k denotes the total number of sample features and s_{ij} denotes the features of the j-th dimension of the samples belonging to the category i.

$$I(s_{1j}, s_{2j}, \dots, s_{mj}) = -\sum_{i=1}^{m} \frac{s_{ij}}{s_j} \log_2 \frac{s_{ij}}{s_j} \tag{6}$$

From Equations (4) and (5) the entropy gain of the feature A is calculated as shown as follow:

$$\text{Gain}(A) = I(s_1, s_2, \dots, s_m) - E(A) \tag{7}$$

The entropy gain rate $\text{Gain}'(A)$ is calculated as shown as follow:

$$\text{Gain}'(A) = \frac{\text{Gain}(A)}{\text{splitInfo}(s)} \tag{8}$$

$$splitInfo(s) = \sum_{i=1}^{m} \frac{s_i}{|s|} \times \log_2 \frac{s_i}{|s|} \tag{9}$$

The Random Forest consists of several decision tree structures, and the main process of its classification is shown in Figure 1.

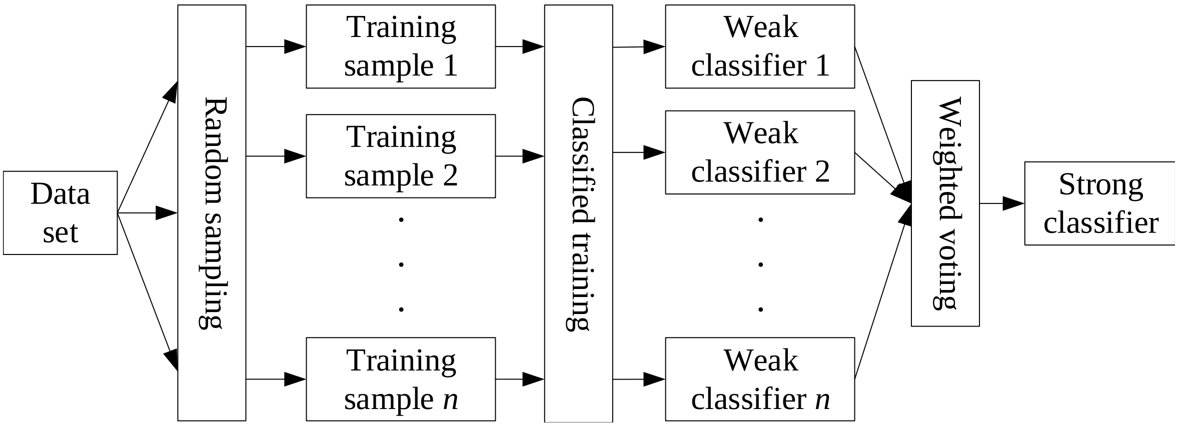

Figure 1. Flow of the Random Forest Model

Let a Random Forest consist of k decision trees $h_1(x), h_2(x), \ldots, h_k(x)$. For any two features X and Y of the sample, the expression of the edge function is shown as follow:

$$ma(X,Y) = av_k(I(h_k(X) = Y)) - \max_{j \neq Y} av_k(I(h_k(X) = j)), \tag{10}$$

where $av_k(\cdot)$ denotes the mean value. The value of $ma(X,Y)$ is proportional to the effect of feature extraction.

3. Weighted Random Forest model based on WOA.

3.1. **Whale swarm optimisation algorithm.** The WOA treats the optimization iterations similarly to the whale hunting movement process [27], where the WOA hunting process importantly consists of prey search, envelope predation and bubble attack.

The WOA search path update methodology is shown as follow:

$$\vec{D} = \left| \vec{C}\vec{X}^*(t) - \vec{X}(t) \right| \tag{11}$$

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \cdot \vec{D}, \tag{12}$$

where $\vec{X}^*$ is the optimal solution coordinate, $\vec{X}$ is the current solution coordinate and $||$ is the absolute value symbol.

The coefficient vectors $\vec{A}$ and $\vec{C}$ are calculated as follow:

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \tag{13}$$

$$\vec{C} = 2\vec{r}, \tag{14}$$

where $\vec{r}$ is a random vector of $[0,1]$.

The wrap-around convergence is shown in Figure 2.

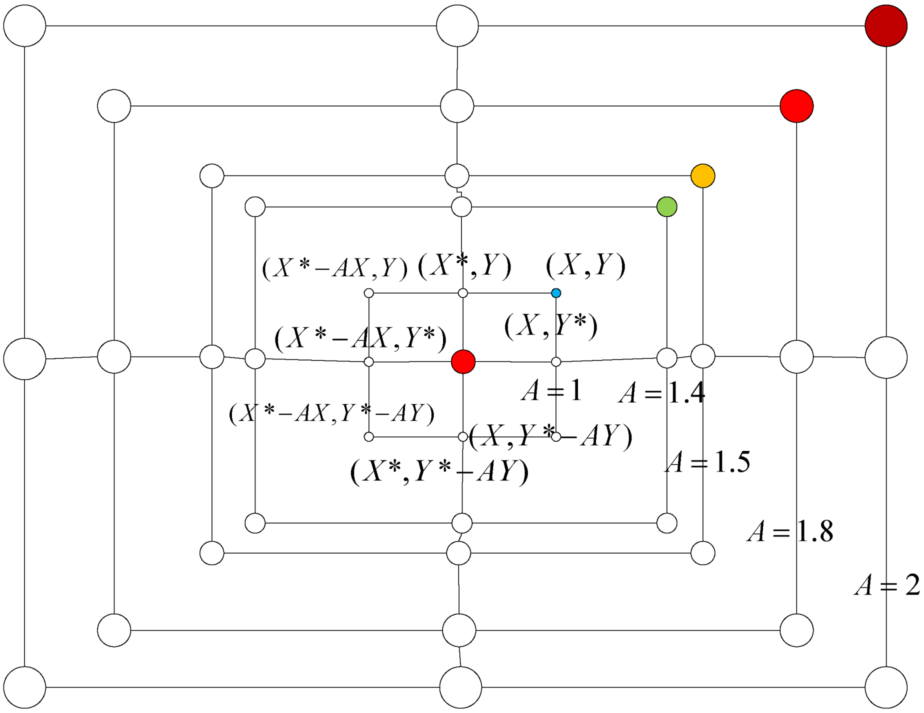

Figure 2. Wrap-around convergence approach

Let the coordinates of a humpback whale be $\vec{X}_{rand}$ and the mathematical expression for the search for prey be as follow:

$$\vec{D} = \left|\vec{C} \cdot \vec{X}_{rand} - \vec{X}\right| \tag{15}$$

$$\vec{X}(t+1) = \vec{X}_{rand} - \vec{A} \cdot \vec{D}, \tag{16}$$

The spiral attack method is shown as follow:

$$\vec{X}(t+1) = \vec{D} \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) \tag{17}$$

When a humpback whale finds its prey, it makes either an enveloping predation or a spiral attack, as determined by the probability p of selection.

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} \cdot \vec{D} & \text{if } p < 0.5 \\ \vec{D} \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) & \text{if } p \geq 0.5 \end{cases} \tag{18}$$

By constantly updating $\vec{X}(t+1)$, the hunting movement is carried out according to Equation (15) and the cycle is iterated until the optimal individual is obtained. The spiral update is shown in Figure 3.

3.2. **Weighted voting strategy for Random Forests.** The weighted Random Forest model's crucial component is the division of the training sample into two portions, one of which serves as the training sample for the conventional Random Forest model and is used to train all of the decision trees. After the remainder of the tree has done training itself, it may be utilised as a prediction sample. The decision trees are then evaluated, and their proper classification rates are computed.

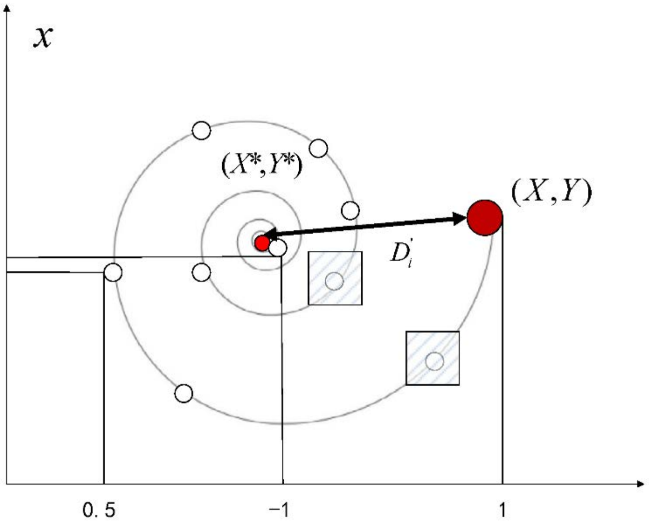

Figure 3. The spiral update approach

$$w_l = \frac{X_l^{\text{correct}}}{X_L}, \quad l = 1, 2, \ldots, L \tag{19}$$

where X_l^{correct} is the number of samples correctly classified by the l-th decision tree out of the L decision trees, X_L is the number of pretest samples, and w_l is the percentage of correctly classified by the l-th decision tree.

Each decision tree in the Random Forest has to be multiplied by its associated weight when it is voted on, with w serving as the appropriate weight of the decision tree. The decision trees and the output of the Random Forest model are shown below:

$$f_l^{\text{tree}}(x) = i \tag{20}$$

$$f_{WRF}(x) = \arg \max_{i=1,2,\ldots,e} \left\{ \sum_{l \in L} w_l \right\} \tag{21}$$

3.3. **Basic process of the WOA-based weighted Random Forest model.** The main steps taken in this paper to optimize the weighted Random Forest with WOA are as follows:

Step 1: Determine the initial parameters, randomly set the pruning threshold ε, the number of decision trees L, the predicted sample rate X and the number of random feature attributes m;

Step 2: Sampling the data to be classified in the UCI dataset using the Bootstrap algorithm, generating L training sets at random, and selecting a random sample of predicted trials in each training set;

Step 3: The remaining samples of each training set are used to generate the decision tree separately. During the generation process, m attributes are selected from all feature attributes as decision attributes for the current node before each attribute selection;

Step 4: When the number of samples contained within a node is less than the threshold ε, the node is taken as a leaf node and its target attribute is returned as the classification result of this decision tree, otherwise return to step;

Step 5: When all decision trees have been generated, pretest each decision tree and record its correctness;

Step 6: Calculate the classification results of the weighted Random Forest model using Equation (21);

Step 7: The classification results were used as accuracy values and the WOA algorithm was used to iteratively optimize the parameters mentioned in Step 1 above. The Random Forest model is generated according to the final optimized parameters.

3.4. **Calculation of inter-individual whale distances.** The calculation of inter-individual whale distances is used to find the guiding individual for each whale [29], so calculating the effective distance between whales can greatly improve the effectiveness of the algorithm.

The purpose of this paper is to calculate the number of elements that need to be added or removed to transform from one individual to another, and to define this number as the distance between these two individuals. The pseudo-code for calculating this distance is shown in Algorithm 1.

Algorithm 1 Pseudo-code for calculating the distance between whale individuals

```
Input: individual whales $\Omega_1$, $\Omega_2$
Output: distance between individual whales.
begin
    Dist = 0, uIndex = 1, vIndex = 1.
    Sort($\Omega_1$), Sort($\Omega_2$)        ▷ Sort the individual elements of the whale
while $uIndex \neq \Omega_1$.length and $vIndex \neq \Omega_2$.length do
    if $\Omega_1[uIndex] < \Omega_2[vIndex]$ then
        $uIndex++$, $Dist++$
    else if $\Omega_1[uIndex] = \Omega_2[vIndex]$ then
        $uIndex++$, $vIndex++$
    else
        $vIndex++$, $Dist++$
    end if
end while
return Dist
end
```

3.5. **Individual whale movement rule.** The individual whale movement rule is designed to guide the movement of the whale by the ”nearest and better whale” to achieve a closer approach to the optimal solution and to obtain the expected next generation of whales.

However, due to the special discrete encoding of individual whales, the individual iteration of the WOA algorithm is not applicable to the weights in Random Forests. Therefore, the individual movement rules of the WOA algorithm are improved to ensure that individual whales move towards their ”nearest and better whale” in order to address the individual encoding characteristics of weights in Random Forests. Two parameters are introduced, the bootstrap probability λ_r and the variance probability λ_o. The pseudo-code for the individual movement rule for whales is shown in Algorithm 2.

4. Experimental results and analysis.

4.1. **Experimental platform and dataset.** The experimental platform is based on Windows and uses the Python language to build the original algorithm model of the Random Forest and incorporates the improved WOA algorithm.

The dataset was selected from the publicly available dataset UC. The UCI database is a database for machine learning proposed by the University of Califonia Irvine, and

Algorithm 2 Pseudo-code for calculating the distance between whale individuals

```
Input: individual whale $\Omega$, the nearest and better whale $Y$
Output: expected next generation Whale $X$
begin
  Generate an empty set $O$;
for Index=1 to $\Omega$.length do
    if $Y$ does not exist in $\Omega$[Index] then
        Add $\Omega$[Index] to the set $O$;
    end if
end for
for Index=1 to $Y$.length do
    if $\Omega$ exists in $Y$[Index] then
        $X$[Index]=$Y$[Index];
    else if Rnd (0,1) $< \lambda_r$ then
        $X$[Index]=$Y$[Index];
    else
        $X$[Index] $\rightarrow O$;
        Remove $X$[Index] from $O$;
    end if
    if Rnd (0,1) $< \lambda_o$ then
        Randomly initialize $X$[Index];
    end if
end for
return $X$
end
```

this database currently has 488 datasets. The UCI dataset is a commonly used standard test dataset. In this paper, four commonly used datasets were selected from the UCI, namely Abalone, Cancer, Letter and Yeast, with Abalone and Cancer having higher feature strength, Letter having average feature strength and Yeast having lower feature strength. The details of these four datasets are shown in Table 1.

Table 1. The information of UCI data sets.

Data set name	**Sample size**	**Number of features**	**Number of classes**	**Unevenness**
Abalone	231	13	3	32.05%
Cancer	593	9	2	35.14%
Letter	406	2	2	25.87%
Yeast	814	8	2	35.02%

The experimental hardware configuration is shown in Table 2. The parameters related to the algorithm in the experiments are set as shown in Table 3.

4.2. **Classification performance of samples with different degrees of imbalance.** To verify the classification performance of the WOA+ Random Forest algorithm for samples with different degrees of equilibrium.

The number of positive and negative samples were differentially selected from Table 1, and different subsets of positive and negative sample ratios were constructed for classification training to validate the classification accuracy and RMSE performance of the WOA+

Table 2. Main computer hardware configuration.

Configuration	**Parameters**
Processor	Inter(R) Core(TM) i3 M 370 @2.4GHz 2.39GHz
Memory	8 G
Hard disk capacity	1 TB
System type	64-bit operating systems
Operating systems	Microsoft Window 10

Table 3. Parameter information.

Configuration	**Parameters**
ntree	100
ω	0.85
λ_r	0.5
λ_o	0.005
$\|\Omega\|$	10

Random Forest algorithm for the 4-class sample set, as shown in Table 4 and Table 5. The highest classification accuracy was obtained for all four sample sets with a positive to negative sample ratio of 3:1, while the lowest classification accuracy was obtained for a positive to negative sample ratio of 6:1. The cross-sectional comparison revealed that the WOA+Random Forest algorithm had the highest classification accuracy for the Yeast sample set and the worst for the Letter sample set.

Table 4. Classification accuracy of different samples.

Category	**Positive and negative sample proportions**	**Classification accuracy**		
		Minimum value	**Average**	**Maximum value**
Abalone	3:1	0.9186	0.9192	0.9197
	4:1	0.9146	0.9149	0.9155
	5:1	0.9078	0.9083	0.9088
	6:1	0.9025	0.903	0.9038
Cancer	3:1	0.9271	0.928	0.9293
	4:1	0.9235	0.9244	0.925
	5:1	0.9183	0.9188	0.9192
	6:1	0.9134	0.9136	0.914
Letter	3:1	0.9132	0.9135	0.9139
	4:1	0.9095	0.9098	0.9104
	5:1	0.9054	0.9058	0.906
	6:1	0.9012	0.9014	0.9017
Yeast	3:1	0.9356	0.9359	0.9362
	4:1	0.9328	0.933	0.9332
	5:1	0.9301	0.9304	0.9306
	6:1	0.9284	0.9287	0.9289

As can be seen in Table 5, the degree of non-equilibrium of the samples has a significant impact on the stability of the data classification. When the ratio of positive to negative samples was 3:1, the RMSE values for classification of all four categories of samples were small. When the difference between positive and negative samples is larger, the RMSE values gradually increase. When the ratio of positive to negative samples was 6:1, the RMSE values increased significantly compared to the 5:1 ratio. This indicates that for 4 classes of samples, the classification algorithm of WOA+Random Forest has a significant stability perturbation in its algorithm classification when the ratio of positive to negative samples is greater than or equal to 6:1.

Table 5. Classification RMSE values of different samples

Category	Positive and negative sample proportions	RMSE Mean
Abalone	3:1	4.06E-02
	4:1	4.09E-02
	5:1	5.64E-02
	6:1	7.52E-02
Cancer	3:1	4.32E-02
	4:1	4.50E-02
	5:1	5.92E-02
	6:1	7.09E-02
Letter	3:1	4.13E-02
	4:1	4.37E-02
	5:1	5.63E-02
	6:1	6.89E-02
Yeast	3:1	3.26E-02
	4:1	3.40E-02
	5:1	4.00E-02
	6:1	6.10E-02

4.3. **Optimization effect of WOA.** In order to verify the optimization performance of WOA for Random Forest classification, the Random Forest algorithm and WOA + Random Forest algorithm were used for classification simulation respectively, and the number of samples tested was 200 for each of the four types, with a positive to negative sample ratio of 4:1. The comparison results are shown in Table 6.

Table 6. Classification recall rate and F1 value of two algorithms

Data sets	Models	Recall rate	F1 value
Abalone	Random Forest	0.7896	0.7844
	WOA + Random Forest	0.8311	0.8123
Cancer	Random Forest	0.7564	0.7463
	WOA + Random Forest	0.8365	0.8171
Letter	Random Forest	0.7501	0.7219
	WOA + Random Forest	0.8321	0.8085
Yeast	Random Forest	0.7911	0.7823
	WOA + Random Forest	0.8509	0.8404

A comparison of the 2 algorithms revealed that the recall and F1 values for the classification of unbalanced data differed significantly between the 2 algorithms, with the WOA-optimised Random Forest achieving higher performance. a comparison of the 4-class sample set revealed that both the Random Forest algorithm and the WOA+Random Forest algorithm achieved the highest recall and F1 values for the classification of the Yeast sample, and the lowest recall and F1 values for the classification of the Letter sample, which indicates that the Random Forest algorithm is most applicable in the classification of the Yeast unbalanced sample set.

4.4. **Comparison of the classification performance of different algorithms.** In order to further verify the effectiveness of the WOA+Random Forest model proposed in this paper in dealing with unbalanced datasets, the Abalone with high feature strength and the Yeast dataset with low feature strength were processed so that they presented five different balance degrees. SVM [30], KNN [31], Random Forest and WOA+ Random Forest were selected for comparison experiments respectively, and the results are shown in Figure 4 and Figure 5.

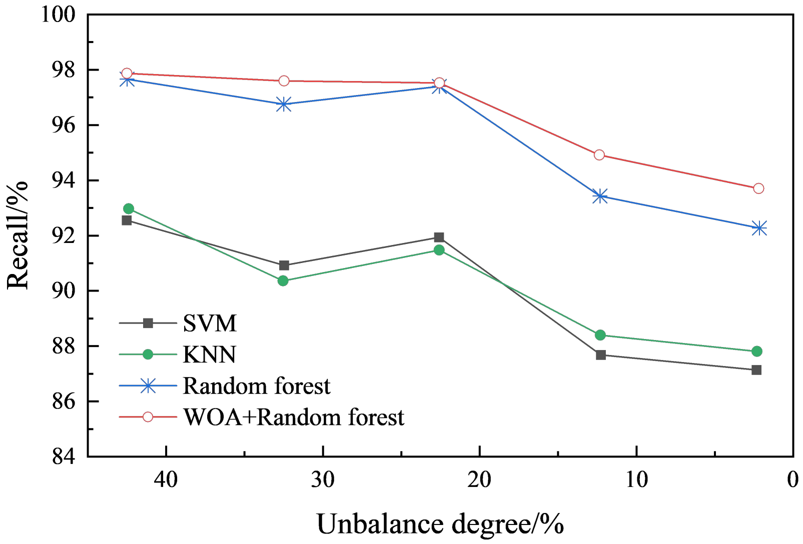

Figure 4. Recall rate of Abalone with different balance degrees

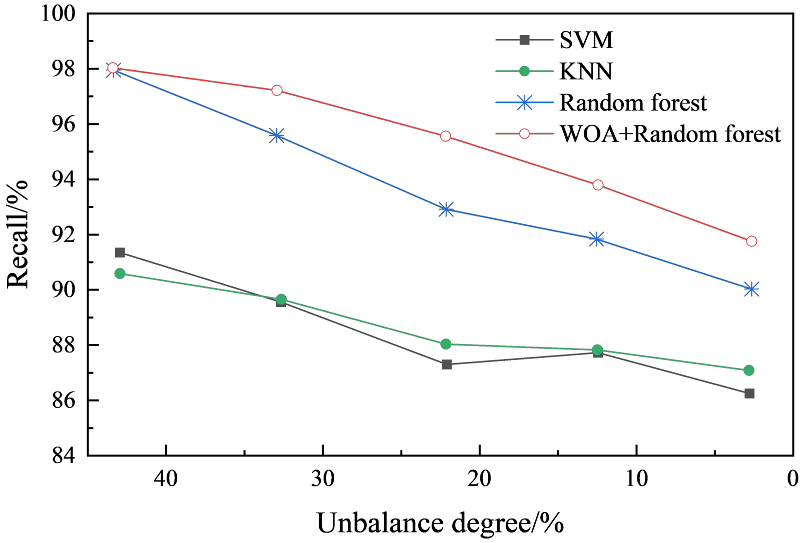

Figure 5. Recall rate of Yeast with different balance degrees

It can be seen that the recognition rates of all four classification models keep decreasing as the proportion of minority classes decreases. Due to the high robustness of the Random Forest, the recognition rates of the Random Forest and WOA+ Random Forest for the minority classes fluctuated less. The recall rates of both were above 93 % on the Abalone dataset, significantly higher than those of SVM and KNN, which did not perform as well. All four classification models showed degradation in recognition rate on the low feature intensity Yeast dataset compared to the high feature intensity Abalone dataset. However, the rate of decline in recall was smoother for Random Forest and WOA+Random Forest than for SVM and KNN, which means that the recognition accuracy for a few classes was higher.

In addition, for both Abalone with high feature strength and YeastYeast with low feature strength, the recall rate of WOA+ Random Forest was higher than that of standard Random Forest under the same non-equilibrium degree condition. At a non-equilibrium degree of 5 %, the recall rate of WOA+ Random Forest improved by 11%, verifying its effectiveness in classifying sub-equilibrium data.

5. Conclusion.

The selection of the number of decision trees and other parameters in a Random Forest model usually also has a large impact on the final classification results. Therefore, a WOA +Random Forest model is proposed in this work. An accuracy-weighted Random Forest model is proposed. The proposed scheme is then parameter optimised using WOA on the basis of the weighted model, thus improving the accuracy of the Random Forest model for minority class recognition. In addition, the inter-individual distance calculation and body movement rules of WOA are improved in order to enhance the local search capability of WOA and its convergence speed. Simulation results show that the recall rate of WOA+Random Forest is improved by 11 % at a non-equilibrium degree of 5 % compared to that of standard Random Forest. The research in this work belongs to the discrete optimisation problem. For discrete optimisation problems, the use of a discrete individual coding scheme can greatly reduce the scope of the search space. Therefore, follow-up research will improve the whale individual coding strategy to narrow the search field and boost the effectiveness of the algorithmic solution.

Funding Statement. This work supported by project of department of education of Guangdong province (No. 2022WZJD013), and Department of Education of Guangdong Province: Research team on integration of data and reality and management innovation (No. 2023WCXTD023).

REFERENCES

[1] T.-Y. Wu, H. Li, and S.-C. Chu, "CPPE: An Improved Phasmatodea Population Evolution Algorithm with Chaotic Maps," *Mathematics*, vol. 11, no. 9, 1977, 2023.

[2] T.-Y. Wu, A. Shao, and J.-S. Pan, "CTOA: Toward a Chaotic-Based Tumbleweed Optimization Algorithm," *Mathematics*, vol. 11, no. 10, 2339, 2023.

[3] F. Zhang, T.-Y. Wu, Y. Wang, R. Xiong, G. Ding, P. Mei, and L. Liu, "Application of Quantum Genetic Optimization of LVQ Neural Network in Smart City Traffic Network Prediction," *IEEE Access*, vol. 8, pp. 104555-104564, 2020.

[4] W. Haoxiang, and S. Smys, "Big data analysis and perturbation using data mining algorithm," *Journal of Soft Computing Paradigm (JSCP)*, vol. 3, no. 01, pp. 19-28, 2021.

[5] V. Plotnikova, M. Dumas, and F. Milani, "Adaptations of data mining methodologies: a systematic literature review," *PeerJ Computer Science*, vol. 6, e267, 2020.

[6] Z. S. Ageed, S. R. Zeebaree, M. M. Sadeeq, S. F. Kak, H. S. Yahia, M. R. Mahmood, and I. M. Ibrahim, "Comprehensive survey of big data mining approaches in cloud systems," *Qubahan Academic Journal*, vol. 1, no. 2, pp. 29-38, 2021.

[7] C.-M. Chen, L. Chen, W. Gan, L. Qiu, and W. Ding, "Discovering high utility-occupancy patterns from uncertain data," *Information Sciences*, vol. 546, pp. 1208-1229, 2021.

[8] L. Chen, W. Gan, Q. Lin, S. Huang, and C.-M. Chen, "OHUQI: Mining on-shelf high-utility quantitative itemsets," *The Journal of Supercomputing*, vol. 78, no. 6, pp. 8321-8345, 2022.

[9] W. Gan, L. Chen, S. Wan, J. Chen, and C.-M. Chen, "Anomaly Rule Detection in Sequence Data," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1-11, 2022.

[10] F. Martínez-Plumed, L. Contreras-Ochando, C. Ferri, J. Hernández-Orallo, M. Kull, N. Lachiche, M. J. Ramirez-Quintana, and P. Flach, "CRISP-DM twenty years later: from data mining processes to data science trajectories," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 8, pp. 3048-3061, 2019.

[11] H. Thakkar, V. Shah, H. Yagnik, and M. Shah, "Comparative anatomization of data mining and fuzzy logic techniques used in diabetes prognosis," *Clinical eHealth*, vol. 4, pp. 12-23, 2021.

[12] K. G. Al-Hashedi, and P. Magalingam, "Financial fraud detection applying data mining techniques: a comprehensive review from 2009 to 2019," *Computer Science Review*, vol. 40, 100402, 2021.

[13] F. E. Bock, R. C. Aydin, C. J. Cyron, N. Huber, S. R. Kalidindi, and B. Klusemann, "A review of the application of machine learning and data mining approaches in continuum materials mechanics," *Frontiers in Materials*, vol. 6, 110, 2019.

[14] Y. Yun, D. Ma, and M. Yang, "Human-computer interaction-based decision support system with applications in data mining," *Future Generation Computer Systems*, vol. 114, pp. 285-289, 2021.

[15] F. A. Thabtah, and P. I. Cowling, "A greedy classification algorithm based on association rule," *Applied Soft Computing*, vol. 7, no. 3, pp. 1102-1111, 2007.

[16] S. Uddin, I. Haque, H. Lu, M. A. Moni, and E. Gide, "Comparative performance analysis of the K-nearest neighbour (KNN) algorithm and its different variants for disease prediction," *Scientific Reports*, vol. 12, no. 1, pp. 1-11, 2022.

[17] M. Belgiu, and L. Drăguţ, "Random forests in remote sensing: a review of applications and future directions," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 114, pp. 24-31, 2016.

[18] A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla, "SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary," *Journal of Artificial Intelligence Research*, vol. 61, pp. 863-905, 2018.

[19] Z. Jiang, T. Pan, C. Zhang, and J. Yang, "A new oversampling method based on the classification contribution degree," *Symmetry*, vol. 13, no. 2, 194, 2021.

[20] C. Luo, Z. Wang, S. Wang, J. Zhang, and J. Yu, "Locating facial landmarks using probabilistic random forest," *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2324-2328, 2015.

[21] Q. Zhou, H. Zhou, and T. Li, "Cost-sensitive feature selection using random forest: Selecting low-cost subsets of informative features," *Knowledge-based Systems*, vol. 95, pp. 1-11, 2016.

[22] M. N. Adnan, and M. Z. Islam, "Forest PA: Constructing a decision forest by penalizing attributes used in previous trees," *Expert Systems with Applications*, vol. 89, pp. 389-403, 2017.

[23] B. Charbuty, and A. Abdulazeez, "Classification based on decision tree algorithm for machine learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, pp. 20-28, 2021.

[24] X. Xin, Q. Jiang, S. Li, S. Gong, and K. Chen, "Energy-efficient scheduling for a permutation flow shop with variable transportation time using an improved discrete whale swarm optimization," *Journal of Cleaner Production*, vol. 293, 126121, 2021.

[25] R. K. Dhanaraj, V. Ramakrishnan, M. Poongodi, L. Krishnasamy, M. Hamdi, K. Kotecha, and V. Vijayakumar, "Random forest bagging and x means clustered antipattern detection from sql query log for accessing secure mobile data," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1-9, 2021.

[26] M. H. D. M. Ribeiro, and L. dos Santos Coelho, "Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series," *Applied Soft Computing*, vol. 86, 105837, 2020.

[27] S. Talukdar, B. Ghose, Shahfahad, R. Salam, S. Mahato, Q. B. Pham, N. T. T. Linh, R. Costache, and M. Avand, "Flood susceptibility modeling in Teesta River basin, Bangladesh using novel ensembles of bagging algorithms," *Stochastic Environmental Research and Risk Assessment*, vol. 34, pp. 2277-2300, 2020.

[28] C. Zhang, J. Tan, K. Peng, L. Gao, W. Shen, and K. Lian, "A discrete whale swarm algorithm for hybrid flow-shop scheduling problem with limited buffers," *Robotics and Computer-Integrated Manufacturing*, vol. 68, 102081, 2021.

[29] G. Wang, L. Gao, X. Li, P. Li, and M. F. Tasgetiren, "Energy-efficient distributed permutation flow shop scheduling problem using a multi-objective whale swarm algorithm," *Swarm and Evolutionary Computation*, vol. 57, 100716, 2020.

[30] V. K. Chauhan, K. Dahiya, and A. Sharma, "Problem formulations and solvers in linear SVM: a review," *Artificial Intelligence Review*, vol. 52, no. 2, pp. 803-855, 2019.

[31] W. Xing, and Y. Bei, ”Medical health big data classification based on KNN classification algorithm,” *IEEE Access*, vol. 8, pp. 28808-28819, 2019.