

A New Random Forest Applied to Heavy Metal Risk Assessment

Ziyan Yu¹, Cong Zhang^{1,*}, Naixue Xiong² and Fang Chen¹

¹Wuhan Polytechnic University, Department of Mathematics and Computer Science, Wuhan, 430023, China ²Northeastern State University, Department of Mathematics and Computer Science, Tahlequah, OK, 74464, USA

*Corresponding Author: Cong Zhang. Email: hb_wh_zc@163.com Received: 04 March 2021; Accepted: 30 April 2021

Abstract: As soil heavy metal pollution is increasing year by year, the risk assessment of soil heavy metal pollution is gradually gaining attention. Soil heavy metal datasets are usually imbalanced datasets in which most of the samples are safe samples that are not contaminated with heavy metals. Random Forest (RF) has strong generalization ability and is not easy to overfit. In this paper, we improve the Bagging algorithm and simple voting method of RF. A W-RF algorithm based on adaptive Bagging and weighted voting is proposed to improve the classification performance of RF on imbalanced datasets. Adaptive Bagging enables trees in RF to learn information from the positive samples, and weighted voting method enables trees with superior performance to have higher voting weights. Experiments were conducted using G-mean, recall and F1-score to set weights, and the results obtained were better than RF. Risk assessment experiments were conducted using W-RF on the heavy metal dataset from agricultural fields around Wuhan. The experimental results show that the RW-RF algorithm, which use recall to calculate the classifier weights, has the best classification performance. At the end of this paper, we optimized the hyperparameters of the RW-RF algorithm by a Bayesian optimization algorithm. We use G-mean as the objective function to obtain the optimal hyperparameter combination within the number of iterations.

Keywords: Random forest; imbalanced data; Bayesian optimization; risk assessment

1 Introduction

The accumulation of heavy metals in agricultural land will not only affect the quality of crops, but also the quality of life of the surrounding residents. Soil is a non-renewable resource, and monitoring heavy metal content in soil and conducting risk assessment is a very important part of land management work. Soil heavy metal contamination risk assessment is similar to credit assessment and disease detection in that it involves identifying hazardous samples from a large number of samples. Because relying on domain experts for this work yields inefficient and not entirely reliable results, many scholars have resorted to machine learning techniques to improve efficiency. We propose to apply RF to soil heavy metal contamination risk assessment. RF was first proposed by Breiman in 2001 [1], an integrated learning method based on the Bagging algorithm [2]. RF has been applied in various industries, such as Rodriguez-Galiano et al. [3]



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

who applied RF to land cover classification. Feng et al. [4] used RF extracts inundated regions in the spectraltexture feature space and performs well in urban inundation mapping. RF can also be applied to problems such as outlier detection [5]. Because the sample set and feature set of the input base learner are different and the correlation between the base learners is low, RF has a strong generalization capability.

Although RF performs well on most classification or regression problems, its hyperparameters are always a challenge to tune. Hyperparameters are parameters in a model that are used to define model properties or define the model training process, such as the learning rate in a neural network model. When the learning rate is small, the convergence rate may be too slow and the time cost may increase. A high learning rate can lead to poor results. Bayesian optimization (BO) [6] can handle optimization problems with black-box functions, so it is widely used for hyperparameter tuning of integrated learning methods. At the end of this paper, we performed hyperparameter tuning of the RF using BO.

The rest of this paper is organized as follows: In Sections 2 and 3 describe the RF and Bayes Optimization algorithm. Section 4 present the principles and steps of the W-RF. While the analysis of experiments and obtained results are provided in Section 5. We conclude with final remarks in Section 6.

2 Random Forest

2.1 Bagging

Breiman [7] proposed the Bagging algorithm in 1996. The algorithm is based on Bootstrap implementing replacement sampling on the dataset to get several subsets, and use these subsets to learn several base learners, and finally output the result by voting method or averaging method. For weak learning algorithms, Bagging can improve the prediction accuracy, but it has no significant effect for strong learning algorithms and sometimes even decreases the accuracy of prediction. In 2003, Brylla et al. [8] proposed Attribute Bagging and demonstrated that applying it to a base learner sensitive to attribute increase or decrease can improve the accuracy of the model. In 2009, Paleologo et al. [9] proposed a subagging method for the high imbalance of credit assessment dataset and experimentally confirmed that this subagging can improve the performance of the base classifier.

2.2 RSM and CART Decision Tree

Random Subspace Method (RSM) [10] is a classification algorithm proposed by Ho in 1998. RSM implements random sampling of the feature set to obtain several feature subsets, and trains several base classifiers through the feature subsets. Finally, the classification results are output by voting method.

Classification regression tree (CART) [11] can deal with both classification and regression problems. If the goal is to generate a regression tree, the feature with the smallest squared error value is selected as the splitting node in the CART partitioning process. If the goal is to generate a classification tree, the feature with the smallest Gini index is selected as the segmentation node in the CART partitioning process.

2.3 RF for Classification

When using RF for classification, the number of trees in the forest needs to be defined. Suppose the training set contains n input features. RF performs replacement sampling on the training set to obtain T sub-training sets, and then generates T trees using the sub-training sets. To construct the decision tree, m features are randomly selected from the feature set of the sub-training set using the random subspace method. The optimal feature from the m features is selected as the split node using the minimum Gini index as the division criterion. Finally, RF integrates the classification results of all decision trees by the simple voting method and output them.

Suppose the category space of the dataset is $\omega_j = \{\omega_1, ..., \omega_c\}$, *c* is the number of categories, $h_i(x)$ represents the *i*-th base classifier, the number of base classifiers is *T*, and H(x) is the classification result of RF. The classification function of the base classifier is Eq. (1):

$$v_{i,j} = \begin{cases} 1, & \text{if } h_i(x) = \omega_j \\ 0, & \text{else} \end{cases}$$
(1)

Simple voting methods include majority voting and plurality voting. Majority voting [12] means that when the number of votes for a category exceeds more than half of the number of base classifiers, the category is output as the classification result. If there is no category with more than half of the votes, the classification is rejected.

$$H(x) = \begin{cases} \omega_{j}, & \text{if } \sum_{i=1}^{T} v_{i,j} > \frac{1}{2} \sum_{k=1}^{c} \sum_{i=1}^{T} v_{i,k} \\ \text{reject}, & \text{else} \end{cases}$$
(2)

where $v_{i,j}$ represents the number of votes obtained by category ω_j .

Plurality voting [12] means that the category with the highest number of votes is used as the classification result, without considering whether the number of votes exceeds half of the number of base classifiers. If more than one category receives the highest number of votes, one of the categories is randomly selected for output.

$$H(x) = \omega_{\underset{j}{\arg\max}} \sum_{i=1}^{T} v_{i,j}$$
(3)

where $v_{i,j}$ represents the number of votes obtained by category ω_j .

3 Bayesian Optimization

BO is a model-based sequential optimization algorithm (SMBO). Its iterative approach is to complete the first evaluation, then find the next evaluation point for the second evaluation, and so on. When using BO for hyperparametric optimization, the objective function needs to be given as f(x). If f(x) represents the generalization error of the machine learning model, then according to the objective function, the optimization problem to be solved can be expressed as follows.

$$x' = \arg\min_{x \in X} f(x) \tag{4}$$

X is the given hyperparameter search space; x' is a certain combination of hyperparameters in the hyperparameter search space that minimizes f(x); f(x) is the objective function. The objective function can be the generalization error of the model, or the mean square error or accuracy, etc.

The process of BO implementation is:

a. Randomly collect a number of points from the hyperparametric search space and construct a dataset based on the values of the objective function corresponding to these points.

b. Select a probability model to proxy the distribution of this dataset over the objective function. The evaluation point for the next iteration can be determined from the probability distribution. A Gaussian process (GP) is usually used as a probabilistic surrogate model.

c. The next evaluation point is found by maximizing the acquisition function.

d. Combine the evaluation points and corresponding objective function values with the previously constructed dataset, update the probabilistic surrogate model and repeat the above steps. Stop the optimization when the preset number of iterations is reached.

In this way, the minimum of the complex nonconvex objective function can be found with fewer evaluation steps, but more computations are required at each step to determine the next sampling point. The probabilistic surrogate model and acquisition function are the two most important parts of BO. It is called BO because the Bayesian formulas are required in the optimization process. The Bayesian formula is as follows.

$$p(f|D) = \frac{p(D|f)p(f)}{p(D)}$$
(5)

f is the objective function; $D = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ is the dataset consisting of the evaluation point x_i and the corresponding objective function value $y_i = f(x_i), i = 1, 2, ..., n; p(D|f)$ is the likelihood distribution of y; p(f) is the prior probability distribution of f; p(D) is the marginal likelihood distribution; p(f|D) is the posterior probability distribution of objective function f. The posterior probability distribution describes the confidence level of the unknown objective function [13], after correcting the prior by the observed dataset.

4 Improved Random Forest

Although RF has good classification performance because of the increased sample perturbation by Bagging, it still suffers from the following shortcomings.

a. When there is a class imbalance problem in the input training set, the sub-training set obtained by the Bagging algorithm may contain few or no positive samples. This may lead to under-learning of positive samples by RF.

b. Regardless of the good or bad classification performance of the base classifier, the simple voting method gives all base classifiers equal voting weights, which limits the classification performance of RF.

4.1 Adaptive Bagging

Chen et al. [14] proposed Balanced Random Forest (BRF) and Weighted Random Forest (WRF) to deal with imbalanced data. BRF [14] use stratified bootstrap and down-sampling method to learn imbalanced data. Although down-sampling of negative samples results in the same class priority, it reduces the information that can be learned for each tree in the RF. WRF [14] calculates the Gini index and voting results based on the weights of classes, giving higher weights to positive samples. However, if the weights are not appropriate, the classification results will be greatly affected. To overcome the above problems, an adaptive Bagging algorithm is proposed in this paper. The steps for implementing the adaptive Bagging algorithm are as follows.

a. Calculate *pos_num*, the number of positive samples in the input training set. Calculate *ratio*, the percentage of positive samples in the total samples.

b. Draw bootstrap samples from the input training data set to build a sub-training set.

c. If the *ratio* is less than 0.1, go to step 4, otherwise go to step 5.

d. Calculate the number of positive samples in the sub-training set. If the number is less than *pos_num*, bootstrap sampling is performed on the positive samples in the input training set so that the number of sub-training sets is equal to *pos_num*.

e. Generating base classifiers using sub-training sets.

Adaptive Bagging does not limit the number of positive samples in each sub-training set, but ensures that each base classifier can learn a certain number of positive sample information.

4.2 Weighted Voting

The second deficiency of RF can be improved by changing the simple voting method to a weighted voting method. Higher weights are assigned to base classifiers with better classification performance and lower weights are assigned to base classifiers with poorer classification performance. Currently, many scholars have demonstrated that the weighted voting method can improve the classification performance of RF. For example, Yan et al. [15] derived error rate bounds for the confidence voting method and introduced a new weight assignment method to improve the classification accuracy of the algorithm. Zhang et al. [16] uses the posterior probability of the Bayesian equation to express the credibility. The larger the value of posterior probability, the higher the credibility of the base classifier and the larger the weight. He [17], based on Zhang et al. [16], proposed to use the out-of-bag (OOB) accuracy of RF instead of the posterior probability value as the credibility, and obtained a new voting weight calculation formula. It improves the classification accuracy of RF and reduces the computational complexity.

In this paper, we propose to assign weights based on the classification performance metrics of the tree in RF. The metrics for classification performance include accuracy, precision, recall, F1-score, G-mean, etc. In order to select the most suitable metric to be used for weighted voting, we conducted experiments under different metrics. The selected metrics are G-mean, F1-score and recall. Eq. (6) shows the weight function.

$$weight(i) = \frac{metric(i)}{\sum_{i=1}^{T} metric(i)}$$
(6)

T is the number of base classifiers; metric(i) is the performance metric of the *i* base classifier. When the metric is G-mean, it is called G-mean weighted Random Forest (GW-RF). When the metric is F1-score, it is called F1-score weighted Random Forest (FW-RF). When the metric is recall, it is called recall weighted Random Forest (RW-RF).

This paper studies the binary classification problem, Eq. (7) shows the voting function:

$$H(x) = sign\left(\sum_{i=1}^{T} weight(i)h_i(x)\right)$$
(7)

weight(i) is the weight of the *i* base classifier; $sign(x) = \begin{cases} 1, & x \ge 0 \\ -1, & x < 0 \end{cases}$.

4.3 W-RF Algorithm

In binary classification problems such as risk assessment and disease diagnosis, the input datasets are usually imbalanced datasets with less than 10% of the total number of positive samples. The W-RF algorithm uses adaptive Bagging and weighted voting methods to better learn imbalanced data. The pseudo-code of the W-RF algorithm is described below.

Algorithm 1. W-RF

Input: training data set *S*; test data set *D*; $y = \{-1, 1\}$; the number of features *d*; the number of CART trees *T*.

Output: Classification result H(x)

()

1: for i = 1, 2, ..., T do

2: calculate the number of positive samples *pos_num* in S and *ratio*.

3: draw bootstrap samples from S to build a sub-training set S_i .

4: if $ratio \le 0.1$, then go to step 4; else go to step 5.

5: calculate the number of positive samples pos_num_i in S_i .

6: if $pos_num_i < pos_num$, then do bootstrap sample again for the positive samples in S, so that $pos_num_i = pos_num$.

7: randomly extract $k \ (k \le d)$ features from S_i to form a feature subset S'_i .

8: use S_i' to train a base classifier $h_i(x)$.

9: end for

10: combine $h_1(x), h_2(x), \dots, h_i(x)$ to ensemble classifier $E = \{h_1(x), \dots, h_i(x)\}$

11: use test data set D to calculate the metric(i) of each base tree, and use metric(i) to calculate weight(i) with Eq. (6).

$$weight(i) = \frac{metric(i)}{\sum_{i=1}^{T} metric(i)}$$
(6)

12: use Eq. (7) to calculate the final classification result.

$$H(x) = sign\left(\sum_{i=1}^{T} weight(i)h_i(x)\right)$$
(7)

5 Experiments

5.1 Dataset

The dataset 1,2 shown in the Tab. 1 are downloaded from the OODS library [18]. The Cardio dataset is a fetal electrocardiogram dataset edited by a professional obstetrician, with 176 positive samples and 1655 negative samples. The Ionosphere dataset is an ionospheric dataset with two categories, good and bad. Good category has 225 samples and bad category has 126 samples. The Diabetes dataset is downloaded from the UCI machine learning repository [19]. It has 268 positive samples and 500 negative samples.

Sequence name	Dataset	Number of cases	Number of variables	%Positive class
1	Cardio	1831	21	9.6%
2	Ionosphere	351	33	35.9%
3	Diabetes	768	8	34.9%
4	Heavy metal	1161	12	7.5%

 Table 1: Dataset summary

The Heavy metal dataset comes from Wuhan Academy of Agricultural Sciences, which contains sampling information and measurement results of eight heavy metals (As, Cd, Cr, Cu, Ni, Pb, Zn, Hg) in agricultural soils around Wuhan. The dataset contains 12 variables: farmland type, altitude, longitude, latitude, and the content of the heavy metals. It has 87 positive samples and 1074 negative samples.

5.2 Performance Measurement

In order to evaluate the performance of RF, the paper use metrics such as classification accuracy, precision, recall, F1-score, and G-mean. All these metrics are calculated through a confusion matrix.

The calculation formula is shown in Eqs. (8)–(12). Tab. 2 shows the confusion matrix. In addition to the above metrics, the receiver operating characteristic (ROC) curve and the area under the ROC curve (AUC) are also used.

$$Accuracy = \frac{TP + TN}{P + N} \tag{8}$$

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

$$F1 - socre = \frac{2 \times Precision \times Recall}{Precision + Recall}$$
(11)

$$G - mean = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}}$$
(12)

 Table 2: Confusion matrix

	Predicted Positive Class	Predicted Negative Class
Actual Positive Class	True Positive (TP)	False Negative (FN)
Actual Negative Class	False Positive (FP)	True Negative (TN)

5.3 Performance Comparison

Use the 5-fold cross-validation method for experiments. The experimental dataset is divided into five sub-data sets. Given that the original dataset is D, the five sub-datasets are D_1 , D_2 , D_3 , D_4 , D_5 . In each training, D_i is selected from the five subsets as the test set, and the other four subsets are combined as the training set, and i = 1, 2, ..., 5. Take the average of accuracy, precision, recall, F1-score and G-mean obtained from 5 training sessions, draw the corresponding ROC curve and calculate the AUC value.

5.3.1 Performance Comparison between W-RF and RF

Tab. 3 compares the performance of different algorithms on the Cardio dataset. It can be observed that in addition to precision, the results of GW-RF are the best on other metrics. Fig. 1 shows the ROC curve and AUC value of the Cardio dataset. For five subsets, comparing the AUC values under several algorithms shows that the AUC values under W-RF are mostly higher than RF.

Algorithms	Accuracy	Precision	Recall	F1-score	G-mean
GW-RF	0.984	0.993	0.836	0.906	0.913
RW-RF	0.983	0.994	0.830	0.901	0.909
FW-RF	0.980	0.993	0.801	0.885	0.894
RF	0.981	1.000	0.807	0.892	0.898

Table 3: Performance comparison on Cardio dataset



Figure 1: ROC curve and AUC value of the Cardio dataset

Tab. 4 compares the performance of different algorithms on the Ionosphere dataset. It can be observed that the metrics results of GW-RF are best. Fig. 2 shows the ROC curve and AUC value of the Ionosphere dataset. It can be found that on the ionospheric dataset 4, the AUC value obtained by the W-RF algorithm is not ideal. But in other subsets, the AUC values obtained are all higher than RF.

Algorithms	Accuracy	Precision	Recall	F1-score	G-mean
GW-RF	0.934	0.936	0.881	0.905	0.921
RW-RF	0.917	0.906	0.865	0.882	0.904
FW-RF	0.920	0.901	0.874	0.885	0.908
RF	0.923	0.923	0.858	0.888	0.907

Table 4: Performance comparison on ionosphere dataset

Tab. 5 shows the performance of different algorithms on the Diabetes dataset. Fig. 3 shows the ROC curve and AUC value of the Diabetes dataset. As can be seen from Tab. 5, the results of RW-RF are the best in most of metrics. Comparing the AUC values of these algorithms, the AUC value of W-RF is mostly higher than that of RF. Analyzing the experimental results, we can conclude that the classification performance of W-RF is better than that of RF in most cases.

5.3.2 Comparison of Risk Assessment Performance

To verify the effectiveness of the W-RF algorithm, the algorithm was compared with RF, K-nearest neighbors (KNN), logistic regression (LR) and support vector machine (SVM) [20] on the heavy metal dataset. The learning rate of LR is set to 0.1, and the number of iterations is set to 500. The number of K nearest neighbor samples of the KNN is set to 5. SVM use the polynomial kernel as the kernel function, and the penalty coefficient C is set to 1. The number of trees for RF and W-RF is set to 50, and the others remain the default.

Tab. 6 compares the performance of different algorithms on the heavy metal dataset. Comparing the accuracy, all the algorithms are above 90% except SVM. On the contrary, when comparing recall, all the algorithms except SVM and RW-RF have less than 50% recall. It shows that the partitioned hyperplane of SVM has certain distinguishing ability for positive samples, but has a large classification errors for negative samples. Regardless of whether the weight is set by G mean or F1 score, the five metrics obtained by W-RF in this experiment are better than RF and other algorithms. Fig. 4 shows the ROC curve and AUC value of the Heavy metal dataset. For five subsets, comparing the AUC values under several algorithms shows that the AUC values under RW-RF are mostly higher than other algorithms.

5.4 Parameter Optimization

After the experiment in Section 5.3.3, it is found that RW-RF has the best performance on the heavy metal dataset. This section uses the BO to complete the hyperparameter tuning of RW-RF. Tab. 7 shows the hyperparameters of RW-RF.

This paper uses *BayesianOptimization* function from the *bayesian-optimization* library [21] to search which hyperparameter combination yields the best performance results. This function uses GP as probabilistic surrogate model, and the acquisition points of the next iteration are obtained by maximizing the confidence interval of GP. We set G-mean as the optimized objective function and define the search space as (*n_estimators, max_features, max_depth*): the number of CART trees range in the integer interval $\{50, 150\}$, and the ranges of *max_features and max_depth* are also integer interval $\{1, 12\}$ and $\{1, 10\}$.



Figure 2: ROC curve of the Ionosphere dataset

Fig. 5 shows the change of G-mean in 30 iterations of BO. The G-mean derived at the 29th iteration is the largest, at the time $n_estimators = 61$, $max_features = 10$, $max_depth = 7$, and the G-mean is 0.981.

		_			
Algorithms	Accuracy	Precision	Recall	F1-score	G-mean
GW-RF	0.725	0.720	0.366	0.481	0.578
RW-RF	0.734	0.719	0.397	0.508	0.600
FW-RF	0.720	0.719	0.332	0.450	0.553
RF	0.691	0.646	0.253	0.363	0.483

 Table 5: Performance comparison on diabetes dataset

 Table 6: Performance comparison on heavy metal dataset

Algorithms	Accuracy	Precision	Recall	F1-score	G-mean
GW-RF	0.944	0.860	0.284	0.405	0.502
RW-RF	0.968	1.000	0.571	0.718	0.750
FW-RF	0.955	0.964	0.428	0.584	0.650
RF	0.936	1.000	0.148	0.250	0.370
KNN	0.929	0.503	0.106	0.172	0.284
LR	0.919	0.178	0.155	0.162	0.243
SVM	0.771	0.233	0.503	0.243	0.596

 Table 7: Hyperparameters of RW-RF

Hyperparameter	Description
n_estimators	The number of CART trees
max_features	The maximum number of features used to generate the CART tree
max_depth	Maximum depth of CART tree



Figure 3: ROC curve of the diabetes dataset



Figure 4: ROC curve of the Heavy metal dataset



Figure 5: The change of G-mean in 30 iterations of BO

6 Conclusion and Future Work

Based on the shortcomings of RF in classification performance on imbalanced datasets, this paper proposed a W-RF algorithm combining adaptive Bagging and weighted voting. Adaptive Bagging first calculates the proportions of different types of samples on the input training set, and judges whether the current training set is an imbalanced dataset. Secondly, if it is an imbalanced dataset, calculate the number of positive samples in the sub-training set and compare it with the number before sampling. If the number is less than before, sample the positive samples with replacement again to keep the positive samples at least a certain proportion in the sub-training set. The weighted voting method is to give higher weights to the base classifier with higher performance metrics. The performance metrics we choose includes G-mean, F1-score, and recall.

From the experiments on various datasets, we can conclude that the performance of GW-RF, RW-RF, and FW-RF are better than RF and other classification algorithms. However, we could not determine which of GW-RF, RW-RF or FW-RF is better. For heavy metal risk assessment, RW-RF has the best classification performance. Therefore, we use RW-RF to conduct risk assessment experiments on heavy metal pollution in agricultural fields around Wuhan and obtained the optimal combination of hyperparameters using BO. In future work, consider using deep reinforcement learning methods [22] for heavy metal pollution risk assessment, or study images of heavy metal pollution distribution, and try to make image predictions [23]. In addition, since this paper only studies the binary classification problem, we will consider extending the binary classification problem to the multi-classification problem.

Funding Statement: This work was supported in part by the Major Technical Innovation Projects of Hubei Province under Grant 2018ABA099, in part by the National Science Fund for Youth of Hubei Province of China under Grant 2018CFB408, in part by the Natural Science Foundation of Hubei Province of China under Grant 2015CFA061, in part by the National Nature Science Foundation of China under Grant 61272278, and in part by Research on Key Technologies of Intelligent Decision-making for Food Big Data under Grant 2018A01038.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
- [2] A. Alabdulkarim, M. Al-Rodhaan, Y. Tian and A. Al-Dhelaan, "A privacy-preserving algorithm for clinical decision-support systems using random forest," *Computers Materials & Continua*, vol. 58, no. 3, pp. 585– 601, 2019.
- [3] V. F. Rodriguez-Galiano, B. Ghimire and J. Rogan, "An assessment of the effectiveness of a random forest classifier for land-cover classification," *ISPRS Journal of Photogrammetry & Remote Sensing*, vol. 67, no. 1, pp. 93–104, 2012.
- [4] Q. L. Feng, J. T. Liu and J. H. Gong, "Urban flood mapping based on unmanned aerial vehicle remote sensing and random forest classifier—A case of Yuyao," *China Water*, vol. 7, no. 12, pp. 1437, 2015.
- [5] A. F. Oliva, F. M. Pérez, J. V. Berná-Martinez and M. A. Ortega, "Non-deterministic outlier detection method based on the variable precision rough set model," *Computer Systems Science and Engineering*, vol. 34, no. 3, pp. 131–144, 2019.
- [6] B. Shahriari, K. Swersky and Z. Wang, "Taking the human out of the loop: A review of Bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2016.
- [7] L. Breiman, "Bagging predictors," Machine Learning, vol. 24, no. 2, pp. 123–140, 1996.
- [8] R. Brylla, R. Gutierrez-Osunab and F. Queka, "Attribute bagging: Improving accuracy of classifier ensembles by using random feature subsets," *Pattern Recognition*, vol. 36, no. 6, pp. 1291–1302, 2003.
- [9] G. Paleologo, A. Elisseeff and G. Antonini, "Subagging for credit scoring models," European Journal of Operational Research, vol. 201, no. 2, pp. 490–499, 2010.
- [10] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998.
- [11] L. Breiman, J. H. Friedman and R. A. Olshen, "Classification and regression trees (CART)," *Biometrics*, vol. 40, no. 3, pp. 358, 1984.
- [12] Z. H. Zhou, "Ensemble learning," in *Machine Learning*, 1st ed., vol. 8. Beijing, China: Tsinghua University Press, pp.182–183, 2016.
- [13] J. X. Cui and B. Yang, "Survey on Bayesian optimization methodology and applications," *Journal of Software*, vol. 029, no. 010, pp. 3068–3090, 2018.
- [14] C. Chen and L. Breiman, Using random forest to learn imbalanced data. Berkeley: University of California, 2004.
- [15] J. K. Yan, H. Zheng, Y. Wang and L. J. Zeng, "Voting by confidence," *Chinese Journal of Computer*, vol. 28, no. 008, pp. 1308–1313, 2005.
- [16] X. Zhang, M. Q. Zhou and G. H. Geng, "Research on improvement of bagging Chinese text categorization classifier," *Journal of Chinese Computer Systems*, vol. 31, no. 2, pp. 281–284, 2010.
- [17] J. He, "Application of random forest in text classification," M.S. thesis. South China University of Technology, China, 2015.
- [18] S. Rayana, ODDS Library. Stony Brook, NY: Stony Brook University, Department of Computer Science, 2016. [Online]. Available: http://odds.cs.stonybrook.edu.
- [19] D. Dua and C. Graff, UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science, 2019. [Online]. Available: http://archive.ics.uci.edu/ml.
- [20] Z. Wang, R. Jiao and H. Jiang, "Emotion recognition using WT-SVM in human-computer interaction," *Journal of New Media*, vol. 2, no. 3, pp. 121–130, 2020.
- [21] N. Fernando, *Bayesian Optimization: Open Source Constrained Global Optimization Tool for Python*, 2014. [Online]. Available: https://github.com/fmfn/BayesianOptimization.
- [22] W. Fang, L. Pang and W. N. Yi, "Survey on the application of deep reinforcement learning in image processing," *Journal on Artificial Intelligence*, vol. 2, no. 1, pp. 39–58, 2020.
- [23] W. Fang, F. Zhang, Y. Ding and J. Sheng, "A new sequential image prediction method based on LSTM and DCGAN," *Computers Materials & Continua*, vol. 64, no. 1, pp. 217–231, 2020.