Advances in Computer, Signals and Systems (2023)
Clausius Scientific Press, Canada

DOI: 10.23977/acss.2023.070610
ISSN 2371-8838 Vol. 7 Num. 6

Concrete Slump Prediction Based on Hybrid Optimization XGBoost Algorithm

Wanli Xiong, Yi Wang*, Junping Wu, Zhichao Hu, Bilian Li

School of Computer Science and Engineering, Sichuan University of Science & Engineering, Zigong, Sichuan, 643000, China
**Corresponding author*

Keywords: Concrete, slump, XGBoost algorithm, particle swarm, grid search

Abstract: In this study, a hybrid optimization XGBoost model was used to predict the slump of concrete. This optimization model combines grid search and particle swarm optimization (PSO) algorithm. The grid search is used to determine the maximum depth and the number of trees in XGBoost, while the particle swarm optimization optimizes other floating-point hyperparameter ranges to improve the predictive accuracy of the model. The factors influencing the slump of concrete include water, cement, fine aggregate, coarse aggregate, and water reducer, which are represented by seven parameters. The model performs excellently in both the training and testing sets, with a coefficient of determination (R2) exceeding 0.97. In conclusion, this study demonstrates that the hybrid optimization of the XGBoost model using grid search and particle swarm optimization algorithm can accurately predict the slump of concrete, which is of significant importance for controlling and optimizing the concrete production process.

1. Introduction

Concrete slump prediction and mixture ratio optimization are crucial steps in the concrete preparation process, as they directly impact the quality and performance of concrete. Therefore, the development of effective prediction models and optimization strategies to enhance prediction accuracy and batching efficiency has garnered significant attention. By training on existing datasets using such prediction models, the slump can be predicted for different concrete mixture ratios, enabling the production of concrete that is better suited to meet specific requirements.

Numerous scholars have proposed their own methods in this regard. Ji Tao et al. proposed an artificial neural network (ANN)-based model for predicting concrete strength and slump. The calculation models for average paste thickness and equivalent water-cement ratio can be obtained by reverse extrapolating the two prediction models [1]. Yeh et al. simulated the slump of self-consolidating concrete (SCC) using an artificial neural network and validated the developed model through response tracking plots. Their study explored the complex nonlinear relationship between concrete components and slump behavior, concluding that response tracking plots can be used for this purpose [2]. Moayedi et al. utilized the ant lion optimizer (ALO) to fine-tune neural networks in the field of concrete slump prediction, and their model performed well in approximating concrete slump [3]. Hamed Safayenikoo et al. employed vortex search algorithm

(VSA), multi-verse optimizer (MVO), and shuffled complex evolution algorithm (SCE) to optimize the configuration of a multi-layer perceptron (MLP) neural network, achieving a 33% reduction in prediction error [4].

To address this problem, we selected multiple models such as XGBoost and random forest to predict and compare their performance using evaluation metrics such as coefficient of determination and root mean square error. These machine learning models have been widely applied in various prediction problems and have demonstrated superior predictive capabilities in many practical applications. However, there is still room for improvement in their application to mixture ratio optimization. After conducting multiple experiments, we found that XGBoost outperformed the random forest model in terms of accuracy, but there was still room for improvement. To further enhance the predictive accuracy of the model, we introduced a hybrid algorithm combining search algorithms and particle swarm optimization.

This study considers the key factors in concrete mixture ratios, including water, cement, fine aggregate, coarse aggregate, slag, fly ash, and water reducer, which have significant influence on the accuracy and practicality of the prediction model. Comparative experiments revealed that the combination of XGBoost model and hybrid algorithm optimization exhibited significant advantages over other methods in terms of both prediction accuracy and mixture ratio optimization in concrete.

The aim of this paper is to compare multiple models through experimental analysis and identify the most effective model, ultimately proving the superiority of XGBoost. Finally, by combining XGBoost with the hybrid algorithm, we achieve precise prediction of concrete slump and optimization of the mixture ratio to enhance the quality and performance of concrete. The goal is to provide a more accurate and optimized strategy for concrete preparation, with the hope of widespread application in engineering.

2. Data Collection and Statistical Analysis of Data

2.1 Data Collection

The data for this experiment is obtained from reference [5], which consists of over 1000 data sets. Each data set includes 7 concrete mixture ratios and the corresponding slump values of the concrete produced using those ratios. The 7 components of the mixture are water, cement, fine aggregate, coarse aggregate, slag, fly ash, and water reducer. However, a significant portion of the data has missing values for the slump. Therefore, the data sets with missing slump values were discarded, and we retained the remaining 295 data sets with non-empty slump values. The sample data is shown in Table 1.

Table 1: Construction of a dataset for predicting porosity models.

Water (kg/m^3)	Cement (kg/m^3)	Fine Aggregate (kg/m^3)	Coarse Aggregate (kg/m^3)	Slag (kg/m^3)	Fly ash (kg/m^3)	SP (kg/m^3)	Slump (kg/m^3)
185	225	900	746	135	90	5.85	28
167	318.8	900	814	81.4	49.8	5.31	25.5
167	131.3	900	772	188.6	130.2	5.31	27.3
180	238.4	900	745	81.4	130.2	5.31	25
180	211.6	900	771	188.6	49.8	5.31	27.5
...	...	...	...	...	...	...	...
169.6	109.4	940	799	157.1	108.5	4.58	27.5
182.9	198.7	940	770	67.9	108.5	4.58	28
182.9	176.3	940	791	157.1	41.5	3.67	26

2.2 Statistical Analysis of Data

The 295 data sets were subjected to statistical analysis to calculate the mean, standard deviation, minimum value, and maximum value for each variable, as shown in Table 1. Additionally, the Pearson correlation coefficients were calculated to explore the linear relationships between variables, as shown in Table 2 and Figure 1.

There is a significant negative correlation between the variables of water and water reducer. Increasing the amount of water tends to decrease the amount of water reducer used, and vice versa. The correlation coefficient between coarse aggregate and fine aggregate is -0.702177, indicating a significant negative correlation. Increasing the proportion of coarse aggregate leads to a decrease in the proportion of fine aggregate, and vice versa. This is expected since the total aggregate proportion is fixed. The correlations between other variables are relatively low, which may indicate weak associations or non-linear relationships that are not accurately captured by the Pearson correlation coefficient.

There is a noticeable positive correlation between slump and water reducer. Increasing the dosage of water reducer leads to an increase in the slump of the concrete. This could be attributed to the fact that water reducers help improve the workability of the concrete [6].

Table 2: Statistical analysis of the data

Feature Name	Unit of Measurement	Minimum Value	Maximum Value	Average	Standard Deviation
Water	kg/m^3	116.5	255.0	179.92	24.51
Cement	kg/m^3	109.4	532.0	261.47	81.79
Fine Aggregate	kg/m^3	30.0	1293.0	851.23	194.56
Coarse Aggregate	kg/m^3	436.0	1226.0	838.92	153.62
Slag	kg/m^3	0.0	375.0	89.58	74.51
Slag	kg/m^3	0.0	270.0	82.05	52.51
SP	kg/m^3	0.0	14.0	4.57	3.08
Slump	cm	8.5	28.5	22.90	5.55

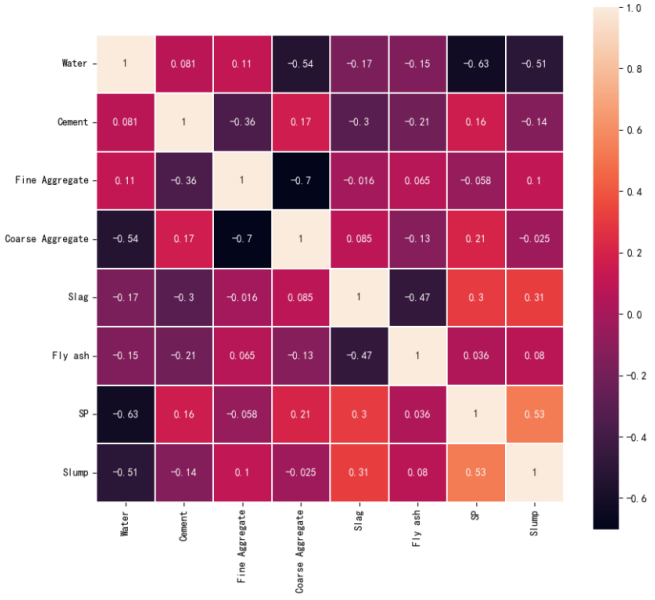

Figure 1: Correlation Coefficients between Variables and Slump

3. Method and Principles

3.1 Overview of XGBoost Model

XGBoost is an efficient machine learning algorithm based on gradient boosting that is designed to tackle large-scale and high-performance machine learning problems. It was developed by Tianqi Chen and his team at the University of Washington in 2014 and has since become an open-source project with widespread applications and high recognition. XGBoost offers the following key features: excellent predictive performance, parallel processing capability, support for various model forms, built-in model validation and early stopping mechanisms, powerful regularization, and the ability to handle sparse data and missing values [7].

XGBoost is an additive model composed of k base models. Assuming that the tree model to be trained in the t-th iteration is denoted as $f_t(x_i)$, we have:

$$\hat{y}_i^t = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{1}$$

Where $\hat{y}_i^t$ is the predicted result of sample i after the t-th iteration, and $\hat{y}_i^{(t-1)}$ is the predicted result of the previous t-1 trees. The objective function of XGBoost can be formulated as:

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{i=1}^{t} \Omega(f_i) \tag{2}$$

The objective function consists of two components: the loss function and regularization. The loss function, $\sum_{i=1}^{n} l(y_i, \hat{y}_i)$, measures the discrepancy between the true value y_i and the predicted value $\hat{y}_i$ for each sample, where n represents the number of samples. The regularization term, $\sum_{i=1}^{t} \Omega(f_i)$ is the sum of complexities across all trees and serves as a regularization term to prevent overfitting. It incorporates $\Omega(f) = \gamma T + \frac{1}{2}\lambda\|\omega\|^2$, γ penalty coefficient for leaf nodes, T as the number of leaf nodes, ω as the leaf weights, and λ as the weight penalty coefficient.

The prediction of the t-th model for the i-th sample, x_i, is given by:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{3}$$

Where $\hat{y}_i^{(t-1)}$ represents the predicted value given by the (t-1)th step model and is a known constant, and $f_t(x_i)$ is the prediction of the new model to be added in this step. The objective function can be written as:

$$\begin{aligned} Obj^{(t)} &= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^{t} \Omega(f_i) \\ &= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \sum_{i=1}^{t} \Omega(f_i) \\ &= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + C \end{aligned} \tag{4}$$

Where C is a constant. Next, we need to find an $f_t(x_i)$ that minimizes the value of the objective function. Therefore, we approximate the objective function by performing a second-order Taylor expansion, resulting in an approximation of the objective function as:

$$Obj^{(t)} \cong \sum_{i=1}^{n}\left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) + C \tag{5}$$

Where $g_i = \partial_{\hat{y}_i^{(t-1)}} l\left(y_i, \hat{y}^{(t-1)}\right)$ and $h_i = \partial^2_{\hat{y}_i^{(t-1)}} l\left(y_i, \hat{y}^{(t-1)}\right)$ are derived from the second-order Taylor expansion. Since $\hat{y}_i^{(t-1)}$ is a known constant at step t, $l(y_i, \hat{y}_i^{(t-1)})$ is a constant and does not affect the optimization of the function. Therefore, we can remove all constant terms, resulting in the objective function:

$$\begin{aligned} Obj^{(t)} &\cong \sum_{i=1}^{n}\left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \\ &= \sum_{i=1}^{n}\left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \gamma T + \frac{1}{2}\lambda \left\| \omega^2 \right\| \end{aligned} \tag{6}$$

3.2 Grid Search Method

In this study, the grid search method was used to optimize the hyperparameters of the XGBoost model. Specifically, the grid search method was employed to determine the maximum depth of the decision trees and the number of trees in the XGBoost model. The grid search method systematically explores multiple combinations of parameters and identifies the optimal parameter combination through cross-validation. In this process, a range and step size need to be specified for each parameter, and all possible parameter combinations are generated [8]. In the XGBoost model, since the maximum depth of the decision trees and the number of trees are integers, it is relatively easy to find the optimal solution within the specified parameter space using the grid search method. However, due to the potentially large search space for other floating-point parameters in the XGBoost model, a particle swarm optimization algorithm will be used for their optimization in subsequent steps. This hybrid approach of using both the grid search method and the particle swarm optimization algorithm ensures both model performance optimization and computational efficiency.

3.3 Particle Swarm Optimization Algorithm

Particle Swarm Optimization (PSO) is an evolutionary computation technique. This method simulates the foraging behavior of a flock of birds. In the search space, each "bird" (referred to as a "particle" or an "individual") has a fitness value determined by a fitness function. Each particle knows its own best position (i.e., the position with the highest fitness it has found) and the globally best position. In each iteration, the particles update their velocities and positions to move towards their own best position and the globally best position [9]. The optimization process of the particle swarm is illustrated in Figure 2.

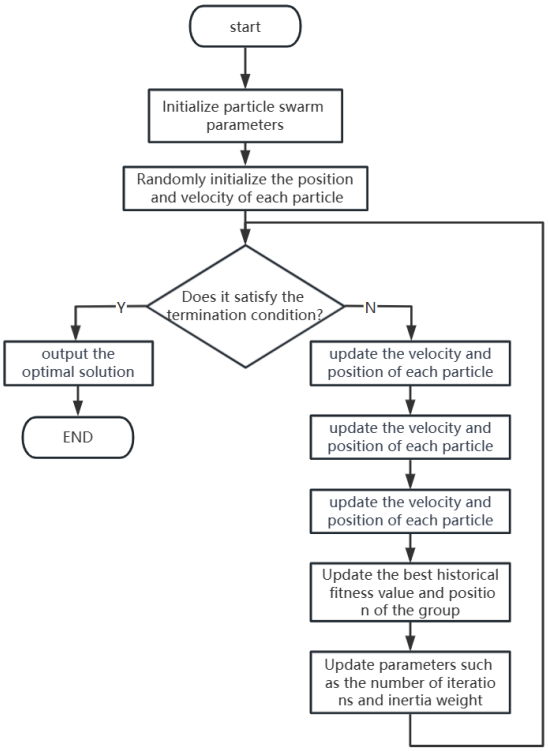

Figure 2: PSO Algorithm Process

4. Based on the hybrid optimization of the XGBoost algorithm for slump prediction

4.1 Evaluation Metrics

In the experimental process, the concrete-related dataset was first read and processed. The dataset includes influencing factors such as water, cement, fine aggregate, coarse aggregate, and water reducer, as well as the slump value of concrete as the output of the model. Then, the dataset was divided into a training set and a test set. Due to the limited number of data in this study (295 samples), the training set was set to 80% of the total, with 236 samples, and the test set was set to 20% of the total, with 59 samples.

Three evaluation metrics were used to assess the model: coefficient of determination (R2), mean squared error (MSE), and mean absolute error (MAE). These metrics were chosen because they can measure different aspects of the model's prediction performance. R2 measures the accuracy of the model's predictions, while MSE and MAE measure the magnitude of the prediction errors. Together, these three metrics provide a comprehensive evaluation of the model.

The calculation formulas for these evaluation metrics are as follows:

$$R^2 = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{7}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{8}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right| \tag{9}$$

4.2 Based on the grid search method for hyperparameter optimization in the XGBoost algorithm

The specific approach to determining the values of two integer hyperparameters, namely the maximum tree depth (max_depth) and the number of trees (n_estimators), in the XGBoost algorithm using grid search is to use the coefficient of determination (R2) as the evaluation metric. The XGBoost model for slump prediction is built using the training dataset. As shown in Figure 3 and Figure 4, the optimal values for max_depth and n_estimators are obtained when the R2 value is maximized during the training process.

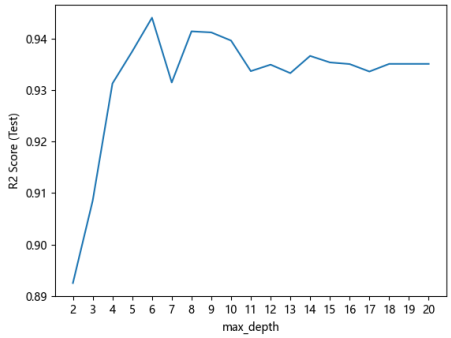

Figure 3: Plot of the relationship between R2 and the max_depth parameter.

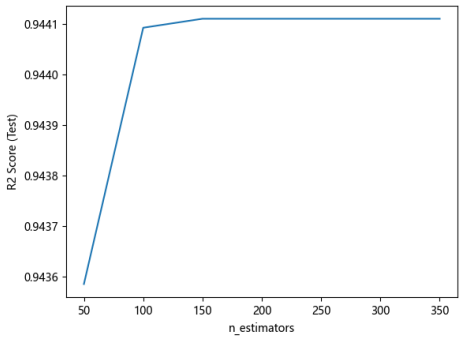

Figure 4: Plot of the relationship between R2 and the n_estimators parameter.

According to the graph, it can be observed that when the maximum tree depth (max_depth) is set to 6 during the training process, the model achieves the highest R2 value of 0.944. As the value of max_depth increases, the R2 values remain below 0.944 and fluctuate around 0.935. Additionally, when the number of trees (n_estimators) reaches 100, the R2 value starts to stabilize. The maximum R2 value of 0.9441 is achieved when n_estimators is set to 123. By using the grid search method, the optimal values of max_depth and n_estimators in the XGBoost model are determined as 6 and 123, respectively.

Similarly, the grid search method is employed to determine the value ranges for three floating-point hyperparameters in the XGBoost model: learning rate (learning_rate), the minimum loss reduction required to make a further partition on a leaf node (gamma), and the subsample ratio of the training instances (subsample). The specified value ranges for these parameters are presented

in Table 3.

Table 3: The range of float hyperparameters in the XGBoost model refers to the range of values that can be assigned to these parameters.

parameters	range
learning_rate	[0,0.3]
gamma	[0.1,0.5]
subsample	[0,0.3]

4.3 Hyperparameter optimization of the XGBoost model based on the PSO algorithm.

Based on the grid search method to determine the value ranges, the particle swarm optimization (PSO) algorithm is utilized to find the optimal values of three floating-point hyperparameters (learning_rate, gamma, and subsample) in the XGBoost model for collapse prediction. Initially, the population size of particles is set to 40, and the number of iterations is set to 50. The training process begins, and the variation of R2 for the XGBoost model with training iterations is illustrated in Figure 5, where the horizontal axis represents the training iterations, and the vertical axis represents R2. It can be observed that as the training iterations increase, the R2 of the XGBoost model gradually improves. When the training iterations reach 22, the R2 stabilizes around 0.972. At this point, the optimal values for the hyperparameters are found as follows: learning_rate = 0.1288, gamma = 0.0068, and subsample = 0.388. Figure 6 depicts the scatter plot of actual values versus predicted values. Table 4 presents a comparison of the performance on the test set between the XGBoost model optimized with the hybrid approach and other models.

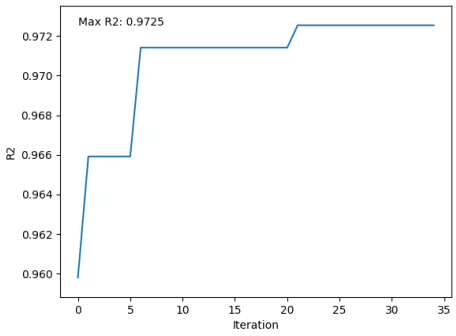

Figure 5: Plot of the variation of R2 with the number of iterations.

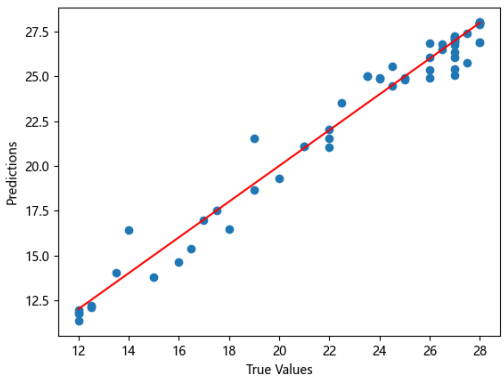

Figure 6: Scatter plot of the actual values and predicted values.

Table 4: Predictive Performance of Five Machine Learning Methods on the Test Set.

Method	R^2	*MSE*	*MAE*
Hybrid-optimized XGBoost	0.9725	0.7808	0.6087
XGBoost	0.9392	1.7298	0.8775
RF	0.9317	1.9417	0.9682
LightGBM	0.9142	2.4395	1.1780
GBDT	0.9099	2.5656	1.1235

5. Conclusion

In this study, a hybrid optimized XGBoost model was developed to predict the slump of concrete, combining grid search and particle swarm optimization (PSO) algorithms. The influence of seven factors including water, cement, fine aggregate, coarse aggregate, and admixture was thoroughly investigated. The experimental results demonstrated that the proposed model exhibited excellent performance on both the training and test sets, with a coefficient of determination (R2) exceeding 0.97.

Compared to other commonly used prediction models such as XGBoost, Random Forest (RF), LightGBM, and Gradient Boosting Decision Trees (GBDT), the hybrid optimized XGBoost model achieved higher prediction accuracy on the test set, with higher R2 scores and significantly reduced mean squared error (MSE) and mean absolute error (MAE). The following advantages of the hybrid optimized XGBoost model can be observed:

Integration of multiple optimization methods: The hybrid optimized XGBoost model combines grid search and particle swarm optimization algorithms. Grid search systematically explores the hyperparameter combinations of the algorithm to find the best model configuration, while particle swarm optimization adjusts model parameters in an adaptive manner to improve performance. By integrating multiple optimization methods, the hybrid optimized XGBoost model can fully leverage the advantages of each method and enhance prediction performance.

Higher prediction accuracy: As shown in Table 4, the hybrid optimized XGBoost model achieved the best performance in terms of R2, MSE, and MAE. It can better fit the data and capture complex relationships within the data, resulting in more accurate predictions. Compared to other algorithms, the hybrid optimized XGBoost model provides more accurate predictions of concrete slump.

Efficient feature learning and ensemble capability: XGBoost, as the base model, is an improved version of gradient boosting algorithm, with powerful feature learning and ensemble capabilities. It can automatically learn the importance of features and perform feature selection to extract the most informative ones. By ensembling predictions from multiple base models, XGBoost can reduce model variance and improve generalization ability.

In summary, the hybrid optimized XGBoost model, combining grid search and particle swarm optimization algorithms, provides a powerful and accurate tool for high-precision prediction of concrete slump. The findings of this study are not only of practical significance for the control and optimization of concrete production processes but also provide strong support and inspiration for related research in the field.

References

[1] Ji T, Lin T, Lin X. A concrete mix proportion design algorithm based on artificial neural networks[J/OL]. Cement and Concrete Research, 2006, 36(7): 1399-1408. DOI:10.1016/j.cemconres. 2006.01.009.

[2] Yeh I C. Exploring Concrete Slump Model Using Artificial Neural Networks [J/OL]. Journal of Computing in Civil Engineering, 2006, 20(3): 217-221. DOI:10.1061/(ASCE)0887-3801(2006)20:3(217).

[3] Moayedi H, Kalantar B, Foong L K. Application of Three Metaheuristic Techniques in Simulation of Concrete Slump[J/OL]. Applied Sciences, 2019, 9(20): 4340. DOI:10.3390/app9204340.
[4] Safayenikoo H, Khajehzadeh M, Nehdi M L. Novel Evolutionary-Optimized Neural Network for Predicting Fresh Concrete Slump [J/OL]. Sustainability, 2022, 14(9): 4934. DOI:10.3390/su14094934.
[5] Shen J.-H. (2013). Application of neural networks in assisting slump concrete mix design with ACI specifications [Master's thesis, National Chiao Tung University]. Retrieved from https://www.airitilibrary.com/ Publication/ alDetailedMesh1? DocID=U0030-0705201411463385. DOI:10.6842/NCTU.2013.00090.
[6] Zhou H.-C. (2022). Experimental study on the influence of different water reducers on concrete performance. Sichuan Cement, (6), 22-24+27.
[7] Zhang H., Zhu J.-P., Zhuo D.-C., et al. (2022). Research on compressive strength prediction model of concrete based on random forest and support vector machine. Engineering and Construction, 36(6), 1784-1788+1815.
[8] Chen H.-N., & Gao X.-L. (2022). Prediction of dissolved gas content in transformer oil based on XGBoost and grid search. Journal of Hebei Normal University (Natural Science Edition), 46(6), 575-581. DOI:10.13763/j. cnki. jhebnu. nse. 202202018.
[9] Li M.-Z. (2020). Research on concrete slump prediction algorithm [Master's thesis, Hunan University]. Retrieved from https://kns.cnki.net/kcms2/article/abstract? v= 3uoqIh G8C475KOm _ zrgu4l QARvep 2SAkue NJRSNVX - zc5TVHKmDNkqWkk0QwGA7r7JKT9zIvlBbzbz5qE7wrH91PzHV29lrv&uniplatform=NZKPT. DOI:10.27135/d. cnki. ghudu. 2020.003737.