Concrete Quality Prediction Model Based on Slime Mould Algorithm

Erdong Zhang, Mingxing Yuan

***Abstract*—Aiming at the difficulty of hyper-parameter tuning commonly found in traditional machine learning algorithms. Combine the Slime Mould Algorithm with traditional machine learning algorithms. Use XGBoost algorithm to construct a concrete compressive strength prediction model, use different algorithms to optimize the parameters, the experimental results show that the SMA algorithm performs well in optimizing the XGBoost parameters, the established concrete strength prediction model is highly efficient, the prediction accuracy is improved significantly, and the complex problem of concrete compressive strength prediction is solved.**

***Index Terms*—Intelligent optimization algorithm; Slime Mould Algorithm; Concrete; Quality prediction; XGBoost**

I. Introduction

The compressive strength of concrete is a key indicator of its quality, and the accuracy of compressive strength is directly related to the safety and stability of buildings. Traditional concrete compressive strength prediction methods mainly rely on empirical formulas and statistical analysis, but these methods are often limited by the amount and complexity of data, and it is difficult to accurately reflect the intrinsic laws of concrete quality.

In contrast, machine learning, as a data-driven technology, can mine the potential characteristics and laws of concrete quality from a large amount of historical data, and then build accurate concrete strength prediction models. These models can not only give the prediction results in a short time, but also dynamically adjust according to real-time data, which greatly improves the accuracy and timeliness of prediction [1][2].

K. Güçlüer et al [3] addressed the key problem of concrete compressive strength prediction, aiming to shorten the experimental period and reduce the cost, by using machine learning algorithms to replace the traditional means of laboratory determination. Four algorithms, namely, artificial neural network, decision tree, support vector machine and linear regression, were compared and analyzed for the prediction of compressive strength of concrete samples at 7 and 28 days of curing in order to find the optimal prediction model. The experiments show that the decision tree algorithm has the best prediction performance and shows the lowest prediction error and the highest prediction accuracy for the compressive strength prediction of concrete samples cured for 28 days. Farroq et al [4] used machine learning algorithms such as Random Forest and Gene Expression Programming, combined with statistical evaluation methods, to effectively predict the compressive strength of high-strength concrete, and a comparative analysis shows the superiority of their model in predicting the mechanical properties of concrete. model outperforms decision trees and artificial neural networks in predicting the mechanical properties of concrete. Debaditya's team [5] successfully constructed a concrete compressive strength prediction model using XGBoost technique and achieved excellent performance with fine tuning. Afshin Marani et al [6] conducted a study of various machine learning algorithms including Random Forest, Extremely Random Tree, Gradient Boosting Decision Tree and XGBoost and tuned various machine learning algorithms and evaluated their prediction accuracy using several metrics. Experimentally, the prediction accuracy of these models was found to be very high. Tuan Nguyen-Sy et al [7] analyzed the performance of three machine learning methods, XGBoost, ANN, and SVM, in predicting the compressive strength of concrete, and found that the XGBoost method outperforms ANN and SVM in terms of accuracy, training speed, and robustness.

The current study shows that machine learning techniques have shown their great potential in predicting concrete strength and have achieved impressive preliminary results [8]-[11]. However, traditional machine learning algorithms generally suffer from the problem of complex and cumbersome hyperparameter tuning process. Hyperparameters refer to some parameters that need to be set before the start of model training; they are not automatically acquired through the training process, but directly affect the learning process and final performance of the model. Since the choice of hyperparameters is often decisive for the performance of the model, failure to set them appropriately may lead to underfitting or overfitting of the model, which seriously affects the prediction accuracy and generalization ability.

To overcome this challenge, academia and industry have turned to the use of intelligent optimization algorithms to assist and improve the hyperparameter tuning process. Such optimization algorithms, such as genetic algorithms, particle swarm optimization, simulated annealing, Bayesian optimization, etc., are able to find the optimal or near-optimal solutions in large-scale or even high-dimensional hyperparameter spaces by virtue of their highly efficient global search strategies and adaptive tuning mechanisms, which greatly reduce the heavy burden of manually tuning the hyperparameters and help to explore the potential performance of the models. Therefore, combining intelligent

Manuscript received March 24, 2024.

Erdong Zhang, College of Computer Science and Technology, Tiangong University, Tianjin, China

Mingxing Yuan, College of Computer Science and Technology, Tiangong University, Tianjin, China

optimization algorithms with traditional machine learning algorithms can not only effectively solve the complexity of hyperparameter settings, but also significantly improve the predictive ability and stability of the model, and thus promote machine learning to achieve more desirable results in various practical applications [12]-[15].

II. Slime Mould Algorithm

Slime Mould Algorithm (SMA), as one of the emerging swarm intelligence optimization algorithms in recent years, was proposed by Shimin Li et al. [16] in 2020. The Slime Mould Algorithm is inspired by the growth and reproduction behaviors of Slime Mould Algorithm in nature and has high parallelism and adaptability. Compared with other swarm intelligence optimization algorithms, the Slime Mould Algorithm has the advantages of fewer parameters, greater parallelism, and stronger global optimization finding ability, and can be used to solve different types of optimization problems [17]-[20].

The Slime Mould Algorithm mainly simulates the foraging behavior and state changes of Physarum polycephalum in nature under different food concentrations. Slime Mould Algorithm is based on the biological oscillator and uses adaptive weights to simulate the positive and negative feedback processes generated by the slime molds during the foraging process, resulting in different morphologies.

When the slime bacteria approach the food, the slime bacteria will determine the direction of the food according to the concentration of the food, which in turn leads to the generation of stronger waves by the biological oscillator, and the mathematical model that simulates the foraging behavior of the slime bacteria is represented in equation (1):

$$X_{t+1}=\begin{cases} r\cdot(UB-LB)+LB\ , & r<z \\ X_b(t)+v_b\cdot\left(W\cdot X_A(t)-X_B(t)\right), & z\le r<p \\ v_c\cdot X_t, & r\ge p \end{cases} \qquad (1)$$

Where t is the current iteration number, $X_b(t)$ is the optimal position of the slime mould individual at the current iteration number, $X_A(t)$ and $X_B(t)$ are the positions of two randomly selected slime mould individuals, v_b as a control parameter takes the value in the range of [-*a*, *a*], V_c is the parameter that decreases linearly from 1 to 0, and r is the random value between[0,1]. *W* is the quality of the slime mould, which represents the fitness weights *z* is the probability of a mucilage isolate individual searching for other food sources, i.e., the probability of generating variation, which according to the experimental study can be obtained at 0.03 for a good balance between the exploitation phase and the exploration phase. *UB* and *LB* denote the upper and lower bounds of the decision space, respectively.

The mathematical model equations for the control variable *p* and parameter V_b are as follows:

$$p=\tanh\left(\left|S(i)-DF\right|\right) \qquad (2)$$

$$v_b=\left[-a,a\right] \qquad (3)$$

S(i) is the fitness value of the *i* th individual, and *DF* is the optimal fitness value in all iterations. The calculation formula of *a* is shown in Equation (4):

$$a=\operatorname{arctanh}\left(-\left(\frac{t}{t_{max}}\right)+1\right) \qquad (4)$$

The fitness weight *W* is shown in Equation (5):

$$W(i)=\begin{cases} 1+r\cdot\log\left(\dfrac{bF-S(i)}{bF-wF}+1\right), & condition \\ 1-r\cdot\log\left(\dfrac{bF-S(i)}{bF-wF}+1\right), & other \end{cases} \qquad (5)$$

where *r* is a random value between [0,1], *S* is the overall set of fitness values (calculated by the objective function), *S(i)* is the fitness value of the *i* th individual, *bF* denotes the best fitness value currently obtained, and *wF* denotes the worst fitness value obtained in the current iteration. When updating the weights, the updating formulas are different for different individuals, where condition denotes the slime mould individual with the first half of the fitness value. From the formula, it can be seen that the weights of the better part of the individuals take larger values within [1,1.3], and the better the individual's weights take values closer to 1; the worse part of the individuals take values within [0.7,1], and the worse the individual's weights are closer to 0.7.

III. Model Building

XGBoost (Extreme Gradient Boosting) is a powerful machine learning algorithm mainly used for classification and regression tasks, developed by Tianqi Chen et al. [21] It is popular in data science and machine learning competitions due to its efficient performance and excellent prediction ability, and has achieved remarkable results in practical applications.The XGBoost algorithm is based on the gradient boosting Decision Tree (GBDT) algorithm, which has been heavily optimized and extended.

In the XGBoost model, hyperparameter optimization plays a crucial role; these parameters are not automatically learned by the training process, but need to be set manually according to the specific problem domain characteristics and data properties. Appropriate selection of hyperparameters can significantly improve the predictive performance of the model, reduce the risk of overfitting, and increase the generalization ability of the model. These parameters include max_depth (maximum depth of each decision tree), learning_rate (step size per iteration), n_estimators (number of trees in the ensemble), gamma (minimum loss reduction required to split a node), min_child_weight (minimum permissible weight for instances in the leaves), subsample (fraction of samples used per tree construction), and colsample_bytree (subsampling rate of columns), among others.

In order to improve the performance and generalization ability of the concrete compressive strength prediction model, this study proposes to use two optimization algorithms, WOA and SMA, to optimize the key parameters of XGBoost, respectively. Meanwhile, four machine learning algorithms, namely SVR algorithm, decision tree algorithm, random forest algorithm and XGBoost algorithm, are used to construct concrete compressive strength prediction models as comparison models, respectively.

For the optimization model, firstly, we formalize the optimization problem as follows: by adjusting the hyperparameters mentioned above, we search for the optimal combination that can maximize the prediction accuracy of the model. In the implementation of each algorithm, all methods adopted the same optimization objective, i.e., based on the performance metrics obtained from cross-validation as the value of the fitness function.

In the implementation stage, each algorithm follows its standard procedure, and the initial parameters and the number of iterations are reasonably set to ensure a fair assessment of the optimization effect of each algorithm.

The dataset used in this paper comes from the UCI Machine Learning Database, which contains 1030 sets of measured data values under nine parameters: cement, blast furnace slag, fly ash, water, high-efficiency water reducer, coarse aggregate, fine aggregate, age and concrete compressive strength. When constructing the initial index system of the concrete compressive strength prediction model, we choose the first eight parameters as input variables, and the ninth parameter, concrete compressive strength, as the output variable.

The XGBoost model was optimized by the above two algorithms, and the concrete compressive strength prediction experiment was completed. By comparing and analyzing the predictive performance indexes obtained by each prediction model, we aim to reveal and evaluate the practical efficacy and applicability of these different optimization strategies in solving the problem of tuning the concrete compressive strength prediction model.

SMA-XGBoost algorithm steps:

Step 1: Initialization phase. First set the key parameters of the algorithm, including the population size *N*, the maximum number of iterations *T*, the dimension *D* of the problem to be optimized, and the upper limit *UB* and lower limit *LB* of the search space.

Step 2: Adaptation evaluation. Under the condition of the current iteration number $t<T$, the initialized corresponding position parameters of the slime mold individuals are applied to the XGBoost model for training, and the fitness values of the individuals are calculated based on the performance metrics on the training set. In this process, the optimal fitness *DF* and the worst fitness *wF* in the current population are recorded.

Step 3: Dynamic weight adjustment and parameter update. Update the adaptation weight matrix *W* and control parameters V_c , V_b , *p*.

Step 4: Position update. Based on the updated state variables, iteratively update the position of each slime mold within the population.

Step 5: Iteration termination determination. Check whether the current iteration number *t* reaches the preset maximum number of iterations *T*. If the maximum number of iterations $t>T$ has been reached, output the optimal positional solution and its corresponding optimal fitness value in the current population of slime molds; if the maximum number of iterations $t<T$ has not been reached, go back to Step 2, and continue to carry out the optimization process in the next iteration cycle.

Step 6: Optimal solution application. After completing all the iterations, the optimal location solution found by the SMA algorithm is used as the final hyperparameter configuration of the XGBoost model, and the XGBoost model is retrained using this configuration to obtain optimal performance.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

Experiments on concrete compressive strength prediction were carried out for each model. The experimental results are shown in Fig. 1, which illustrates the accuracy of the predicted values of the different models in determining the compressive strength of concrete in the form of a scatter plot.

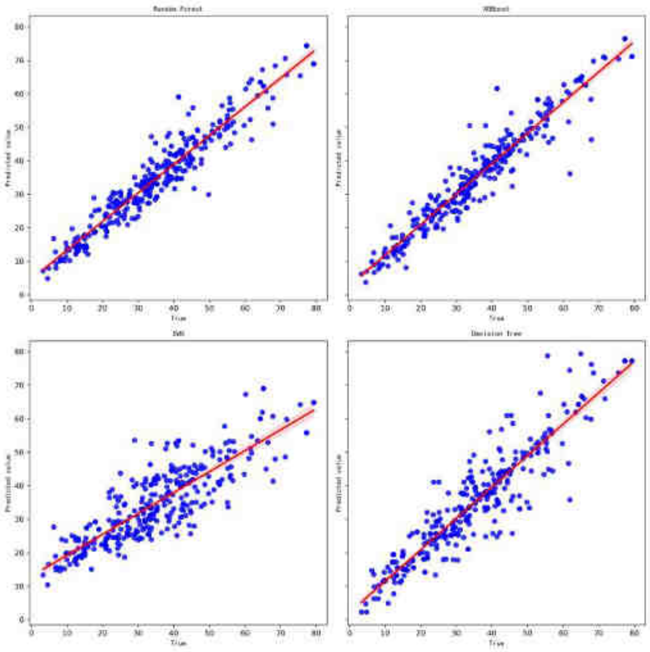

Figure 1 Scatterplot of unoptimized model predictions

In scatterplot analysis, the centerline symbolizes the ideal state in which the predicted compressive strength values exactly match the actual measured values, which means that the closer the data points are to the centerline, the higher the prediction accuracy will be. Observing Fig. 1, it can be seen that among the models without optimization treatment, the random forest and the original XGBoost models present excellent prediction efficacy, and they are highly represented in terms of accuracy and stability. In contrast, the decision tree model is slightly less effective in prediction, suggesting that deep optimization or fine tuning of parameters may need to be implemented to enhance the prediction accuracy. As for the SVR model, its performance is the weakest this time. Although it has some prediction function, there is a large deviation between the predicted data points and the real values.

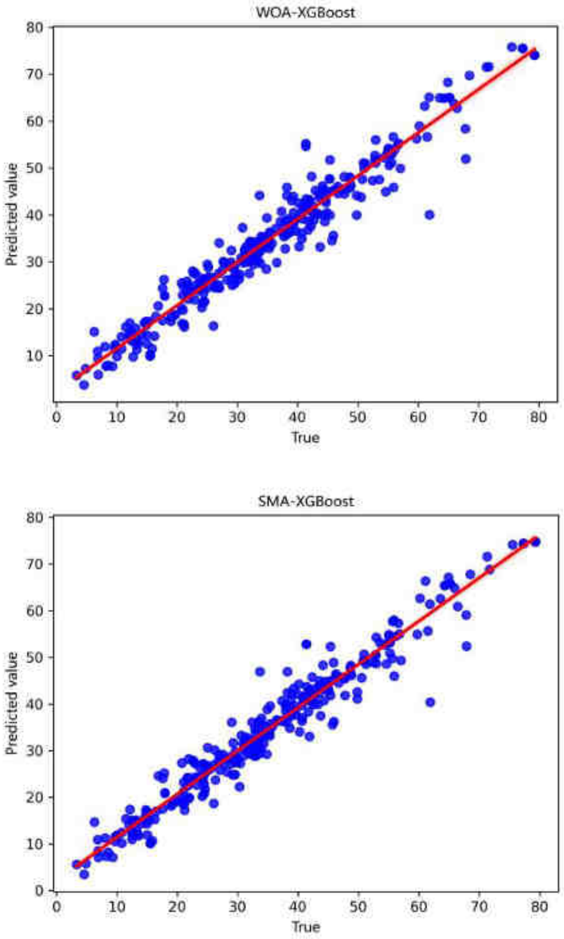

Figure 2 Optimized Scatterplot of Model Predictions

As shown in Fig. 2, In the optimized model camp, the graphical trends of SMA-XGBoost and WOA-XGBoost show an improvement, and both of them show better prediction performance than the unoptimized model. Among them, the SMA-XGBoost model has the most outstanding performance, as its data points are evenly distributed on both sides of the center line without obvious anomalies, which strongly indicates that the predicted concrete compressive strength values of the model are highly close to the actual values, and that the overall model fitting effect and prediction accuracy are excellent.

In order to further evaluate the superiority of the SMA-XGBoost model for the prediction of concrete compressive strength, the performance evaluation indexes of each prediction model were calculated on the basis of the same dataset as follows: coefficient of determination (R^2), mean square error (MSE), mean absolute error (MAE), and average absolute error (AAE). error, MAE). Among them, R^2 judges the feasibility and linear relationship of the model; MSE reflects the expectation of the squared difference between the predicted value and the actual value; MAE reflects the average of the absolute value of the error between the predicted value and the actual value. The prediction performance evaluation indexes of each model are calculated as shown in Table 1.

Table1 The performance criteria values of the models

Method	Testing set		
	R^2	MSE	MAE
XGBoost	0.922	4.563	2.960
Random Forest	0.911	4.862	3.603
Decision Tree	0.852	6.283	4.229
SVR	0.710	8.788	6.993
WOA-XGBoost	0.942	3.885	2.793
SMA-XGBoost	0.947	3.748	2.606

Based on the performance index values shown in Table 1, it can be clearly observed that in the test set, the newly proposed SMA-XGBoost model in this study shows more excellent performance compared with the traditional XGBoost, WOA-XGBoost and other similar models. The R^2 score of the SMA-XGBoost model is as high as 0.947, which is a good validation of the goodness of fit of the model. SMA-XGBoost model has a high R^2 score of 0.947, which is a strong proof of the model's goodness of fit.

Compared with other models, SMA-XGBoost achieves a significant decrease of 30.9%, 5.6%, 10.1% and 9.9% in the MSE of the test set, which clearly demonstrates the advantage of the model in terms of prediction accuracy. In addition, SMA-XGBoost also achieved the optimal level of MAE on the test set, which further confirms the accuracy and effectiveness of the model in predicting the compressive strength of concrete.

Taken together, the experimental results clearly reveal the remarkable effectiveness of the SMA algorithm in optimizing the XGBoost parameter settings. By using the optimized SMA-XGBoost model of the SMA algorithm to construct an efficient and accurate concrete strength prediction tool, the prediction accuracy has been substantially improved. This definitely proves the unique superiority of the SMA-XGBoost model proposed in this study in dealing with the complex task of concrete compressive strength prediction.

V. Conclusion

In this paper, to address the challenges of traditional machine learning algorithms in hyper-parameter tuning, Slime Mould Algorithm (SMA) is introduced to integrate with the classical XGBoost algorithm with a view to enhancing the performance of concrete compressive strength prediction model. It is demonstrated through experiments that the optimization of XGBoost parameters by SMA not only effectively solves the hyperparameter tuning challenge, but also significantly improves the performance of the prediction model.

References

[1] ALAA A-S, PASŁAWSKI J, NOWOTARSKI P. Quality Management to continuous improvements in process of Ready Mix Concrete production; proceedings of the IOP Conference Series: Materials Science and Engineering, F, 2019 [C]. IOP Publishing.

[2] NUNEZ I, MARANI A, FLAH M, et al. Estimating compressive strength of modern concrete mixtures using computational intelligence: A systematic review [J]. Construction and Building Materials, 2021, 310: 125279.

[3] GüçLüER K, ÖZBEYAZ A, GöYMEN S, et al. A comparative investigation using machine learning methods for concrete compressive strength estimation [J]. Materials Today Communications, 2021, 27: 102278.

[4] FAROOQ F, NASIR AMIN M, KHAN K, et al. A comparative study of random forest and genetic engineering programming for the prediction of compressive strength of high strength concrete (HSC) [J]. Applied Sciences, 2020, 10(20): 7330.

[5] CHAKRABORTY D, AWOLUSI I, GUTIERREZ L. An explainable machine learning model to predict and elucidate the compressive behavior of high-performance concrete [J]. Results in Engineering, 2021, 11: 100245.

[6] MARANI A, NEHDI M L. Machine learning prediction of compressive strength for phase change materials integrated cementitious composites [J]. Construction and Building Materials, 2020, 265: 120286.

[7] NGUYEN-SY T, WAKIM J, TO Q-D, et al. Predicting the compressive strength of concrete from its compositions and age using the extreme gradient boosting method [J]. Construction and Building Materials, 2020, 260: 119757.

[8] LIAN X, LIU Y, BU X, et al. Combined forecasting approach for product quality based on support vector regression and gray forecasting model [J]. Advanced Engineering Informatics, 2023, 57: 102070.

[9] TRAN V Q, DANG V Q, HO L S. Evaluating compressive strength of concrete made with recycled concrete aggregates using machine learning approach [J]. Construction and Building Materials, 2022, 323: 126578.

[10] BI Z, FU C, ZHU J, et al. Control Ore Processing Quality Based on Xgboost Machine Learning Algorithm; proceedings of the 2023 Asia-Europe Conference on Electronics, Data Processing and Informatics (ACEDPI), F, 2023 [C]. IEEE.

[11] Feng D C, Liu Z T, Wang X D, et al. Machine learning-based compressive strength prediction for concrete: An adaptive boosting approach[J]. Construction and Building Materials, 2020, 230: 117000.

[12] YANG L, SHAMI A. On hyperparameter optimization of machine learning algorithms: Theory and practice [J]. Neurocomputing, 2020, 415: 295-316.

[13] PROBST P, BOULESTEIX A-L, BISCHL B. Tunability: Importance of hyperparameters of machine learning algorithms [J]. Journal of Machine Learning Research, 2019, 20(53): 1-32.

[14] SUN Y-N, CHEN Y, WANG W-Y, et al. Modelling and prediction of injection molding process using copula entropy and multi-output SVR; proceedings of the 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE), F, 2021 [C]. IEEE.

[15] KHALID R, JAVAID N. A survey on hyperparameters optimization algorithms of forecasting models in smart grid [J]. Sustainable Cities and Society, 2020, 61: 102275.

[16] LI S, CHEN H, WANG M, et al. Slime mould algorithm: A new method for stochastic optimization [J]. Future generation computer systems, 2020, 111: 300-23.

[17] CAI Y, SHARMA A. Swarm intelligence optimization: an exploration and application of machine learning technology [J]. Journal of Intelligent Systems, 2021, 30(1): 460-9.

[18] DASI H, YING Z, ASHAB M F B. Proposing hybrid prediction approaches with the integration of machine learning models and metaheuristic algorithms to forecast the cooling and heating load of buildings [J]. Energy, 2024, 291: 130297.

[19] LI J, LI C, ZHANG S. Application of Six Metaheuristic Optimization Algorithms and Random Forest in the uniaxial compressive strength of rock prediction [J]. Applied Soft Computing, 2022, 131: 109729.

[20] KHAJAVI H, RASTGOO A. Predicting the carbon dioxide emission caused by road transport using a Random Forest (RF) model combined by Meta-Heuristic Algorithms [J]. Sustainable Cities and Society, 2023, 93: 104503.

[21] CHEN T, GUESTRIN C. Xgboost: A scalable tree boosting system; proceedings of the Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, F, 2016 [C].