scientific reports

OPEN

Prediction of concrete compressive strength using a Deepforest-based model

Wan Zhang[1], Jiangtao Guo[1], Cuiping Ning[1], Ruifang Cheng[1] & Ze Liu[2✉]

Concrete compressive strength testing is crucial for construction quality control. The traditional methods are both time-consuming and labor-intensive, while machine learning has been proven effective in predicting the compressive strength of concrete. However, current machine learning-based algorithms lack a thorough comparison among various models, and researchers have yet to identify the optimal predictor for concrete compressive strength. In this study, we developed 12 distinct machine learning-based regressors to conduct a thorough comparison and to identify the optimal model. To study the correlation between compressive strength and various factors, we conducted a comprehensive analysis and selected blast furnace slag, superplasticizer, age, cement, and water as the optimized factor subset. Based on this foundation, grid search and fivefold cross-validation were employed to establish the hyperparameters for each model. The results indicate that the Deepforest-based model demonstrates superior performance compared to the 12 models. For a more comprehensive evaluation of the model's performance, we compared its performance with state-of-the-art models using the same independent testing dataset. The results demonstrate that our model achieving the highest performance (R^2 of 0.91), indicating its accurate prediction capability for concrete compressive strength.

Keywords Concrete, Compressive strength, Factor, Machine learning, Deepforest

Concrete has become the prevalent building material in modern society due to its affordability, ease of casting, and substantial compressive strength after hardening[1]. In 2017, the global annual consumption of concrete reached 30 billion tons, and projections indicate a 50% increase by 2050[2]. To ensure the quality of construction, it is imperative to measure the compressive strength of concrete to guarantee the structural reliability of concrete. The traditional methods involve testing specimens in a specialized laboratory, encompassing several stages: ingredient preparation, mixing, curing, and compression testing[3]. Among these stages, achieving the desired concrete strength and optimal workability in the laboratory requires technicians to thoroughly study various admixtures. If the test results fall short of the required strength in the concrete mix design, the entire concrete design process must be reinitiated, leading to an extended testing cycle and a delay in obtaining timely compressive strength data.

Due to the crucial role of concrete compressive strength testing in engineering construction, researchers have developed numerous methods for testing compressive strength. Currently, the primary testing methods include non-destructive testing and micro-damage testing[4,5]. Specifically, the non-destructive testing method quantifies the strength of concrete based on its relevant physical properties, inferring the relationship between strength and these properties without causing damage to the concrete components. These methods primarily comprise rebound[6], ultrasonic pulse velocity[7], and the ray method[8]. The rebound method, for instance, employs a rebound instrument to measure concrete hardness, inferring its compressive strength based on carbonation depth. In comparison to other methods, this technique is simple and flexible, yet its accuracy is relatively low[6]. Ultrasonic pulse velocity measures concrete strength by observing changes in ultrasonic waves. Nevertheless, variations in materials, moisture content, and age significantly affect the relationship between ultrasonic wave propagation speed and concrete strength, posing challenges in determining compressive strength[7]. While γ rays can be employed for detecting concrete compressive strength, their engineering use is rare due to radiation concerns[8]. The primary advantage of non-destructive concrete testing is its ability to measure compressive

[1]College of Architecture Engineering, Yangling Vocational & Technical College, Shaanxi, Yangling 712100, Shaanxi, China. [2]College of Water Resources and Architectural Engineering, Northwest A&F University, Yangling 712100, Shaanxi, China. ✉email: zeliu@nwafu.edu.cn

1

strength without causing damage, facilitating quick and easy testing. Nevertheless, the accuracy of this method requires improvement.

Micro-damage testing assesses the strength and defects of concrete structures through systematic testing. In contrast to non-destructive testing of concrete, micro-damage testing boasts high detection accuracy. Micro-damage detection methods encompass core drilling[9], Pull-out[10], Nail shooting[11], and others. The core drilling method involves drilling core samples from concrete components, followed by calculating concrete strength through destructive tests. While this method yields highly accurate experimental results, it suffers from the drawbacks of relatively complex operation and high cost[9]. The Pull-out method involves pre-embedding riveted parts in structural components, recording external force magnitudes and the destructive force borne by the concrete during specimen extraction, and estimating compressive strength by establishing a relationship between obtained data and compressive strength. In comparison to traditional core drilling methods, the Pull-out method offers advantages such as simplicity of operation, low cost, and easy implementation. Nevertheless, the Pull-out method is influenced by numerous experimental factors, potentially leading to lower accuracy compared to the core drilling method[10]. The Nail shooting method involves using a hammer shaft to shoot a test nail into the concrete, then using a dial gauge to measure the depth of injection into the concrete, and estimating compressive strength based on this depth. However, establishing its correlation formula necessitates conducting a substantial number of concrete specimen failure tests before employing this method for strength detection[11]. Furthermore, the aforementioned methods are exclusively employed for detecting concrete compressive strength during the usage stage. If a method could predict compressive strength with varying factor ratios in the early design stage, it would significantly expedite industrial construction progress and diminish compressive strength testing costs.

Machine learning has found extensive applications across various fields[12–16], with numerous algorithms utilized to predict concrete compressive strength. These algorithms include gene expression programming (GEP), support vector regressor (SVR), decision tree (DT), adaboost, randomforest (RF), artificial neural network (ANN), and extreme learning machine (ELM). Notably, Gholampour et al. employed GEP to predict the 28-day compressive strength of recycled aggregate concrete, constructing a database for performance testing[17]. Javed et al. applied GEP to predict the compressive strength of Bagasse Ash-based concrete, achieving an R^2 value exceeding 0.8 between observed and predicted values[18]. Yang et al. utilized GEP to predict the compressive strength of concrete blended with carbon nanotubes, evaluating the contribution of different features through shapley analysis[19]. The advantage of GEP lies in its ability to generate formulas in a visible manner, making them easy to understand. However, a disadvantage is that optimizing GEP parameters can be challenging, necessitating efforts to enhance its prediction accuracy. Ling et al. combined SVR with k-fold cross-validation to predict the compressive strength of concrete in marine environments. Nevertheless, this model is a black-box model lacking interpretability[20]. Farooq et al. applied adaboost and RF to predict the compressive strength of high-performance concrete (HPC), demonstrating that the integrated model can enhance the performance of compressive strength prediction[21]. Network connectivity-based models, including ANN and ELM, have been widely employed for predicting concrete compressive strength. Yeh utilized ANN to predict the compressive strength of HPC, validating their model's performance through a set of trial batches[22]. Additionally, Keshavarz and Torkian used ANN to predict compressive strength with different mixing ratios, revealing that while ANN can effectively predict concrete compressive strength, its efficacy is inferior to that of adaptive Neuro Fuzzy Inference[23]. ELM, a relatively recent addition to the ANN family, was employed by Al-Shamiri et al. to predict the compressive strength of high-strength concrete (HSC). The prediction results of the ELM-based model surpassed those of ANN[24]. The advantage of ANN lies in its intricate nonlinear mapping capability, though it comes with the drawback of requiring a substantial amount of data and being susceptible to falling into local optima. Deepforest, an integrated model proposed by Zhou et al. in 2017[25], amalgamates the strengths of RF and deep ANN. It deeply integrates traditional tree-based models (like RF) through a cascaded forest structure, endowing it with robust learning ability and enhancing the nonlinear mapping ability of the model through multi-grained scanning. Furthermore, in comparison to deep neural networks, Deepforest requires fewer hyperparameters and is easier to train.

The objective of this study is to introduce the latest machine learning algorithm and comprehensively evaluate the predictive performance of various machine learning algorithms in predicting the compressive strength of concrete. To identify the optimal combination of factors, the study compared the model's performance with various factor combinations and ultimately determined the optimal subset, comprising blast furnace slag, superplasticizer, age, cement, and water. On this basis, 12 distinct machine learning algorithms were employed for modeling and parameter optimization. Ultimately, results on the independent testing dataset demonstrated that the Deepforest-based model yielded the most effective predictions for concrete compressive strength, making it a more efficient predictor.

Methods
Dataset construction
The dataset used in this study consists of a total of 1030 concrete samples (Downloaded from https://archive.ics.uci.edu/dataset/165/concrete+compressive+strength). It was randomly divided into the training dataset and the independent testing dataset, maintaining an approximate ratio of 8:2. Consequently, 822 samples were utilized for model training, and 208 samples were designated for performance evaluation. Each sample comprises eight factors, namely blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, fine aggregate, age, and cement. The statistical analysis of the sample data is presented in Table 1.

Correlation between different factors and concrete compressive strength
Correlation analysis is a statistical method used to understand the interactions between different factors and delve deeper into their impact on a specific variable[26]. To investigate the correlation between compressive strength and

Factor	Unit	Mean	Mode	Maximum	Minimum	Standard deviation
Blast furnace slag	kg/m^3	73.90	0	359.4	0	86.28
Fly ash	kg/m^3	54.19	0	200.1	0	64.00
Water	kg/m^3	181.57	192	247	121.8	21.35
Superplasticizer	kg/m^3	6.20	0	32.2	0	5.97
Coarse aggregate	kg/m^3	972.92	932	1145	801	77.75
Fine aggregate	kg/m^3	773.58	594	992.6	594	80.18
Age	day	45.66	28	365	1	63.17
Cement	kg/m^3	281.17	425	540	102	104.51
Strength	MPa	35.82	33.4	82.6	2.33	16.71

Table 1. Statistical analysis for the sample data. Mean, mode, maximum, minimum, and standard deviation are the average, mode, maximum, minimum, and standard deviation of each factor.

different factors, all 1030 samples in the benchmark dataset were used for analysis, and the sample distribution of different factors relative to compressive strength is shown in Fig. 1. The compressive strength of each sample ranges from 2.33 to 82.6. As for the different factors—blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, fine aggregate, age, and cement—their values range from 0 to 359.4, 0 to 200.1, 121.8 to 247, 0 to 32.2, 801 to 1145, 594 to 992.6, 1 to 365, and 102 to 540, respectively. Owing to the substantial difference between the values of different factors, the model may exhibit bias towards factors with larger values. Hence, normalizing different characteristic factors is necessary before the follow-up process. Additionally, we observed that the median values of blast furnace slag, fly ash, and high-efficiency water reducing agent in 45.73%, 54.79%, and 36.80% of the samples were 0, respectively. This may be attributed to the non-utilization of these elements or the presence of missing records. If records are missing, it can impact the accuracy and reliability of the machine learning model. Hence, strategies such as deletion, mean/median interpolation, or regression-based interpolation (such as K-Nearest Neighbors, KNN) can be employed to enhance data integrity.

To further investigate the correlation between different factors, the correlation coefficient between each factor was calculated. As shown in Fig. 2, the correlation coefficient of compressive strength and each factor—blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, fine aggregate, age, and cement—were 0.13, − 0.11, − 0.29, 0.37, − 0.16, − 0.17, 0.33, and 0.5, respectively. The superplasticizer factor has the strongest correlation with compressive strength. Furthermore, we also observed that the correlation coefficient between fly ash, water, coarse aggregate, and fine aggregate was negative, suggesting that we should reasonably control the addition of these elements during the experiments.

Normalization

Normalization is employed to convert data into a unified proportional range, thereby eliminating dimensional differences between different factors[27]. By utilizing normalization, data can be mapped to a specific range, such as [0, 1] or [− 1, 1], ensuring that different factors have similar numerical ranges. Some machine learning algorithms, such as ANN and SVR, are sensitive to the numerical range of the input data. If the numerical range of the input data varies significantly, it may lead to slower convergence speed in the model training process. In this study, the maximum-minimum normalization method was applied to map different factors to [0, 1], and the specific calculation formula is as follows:
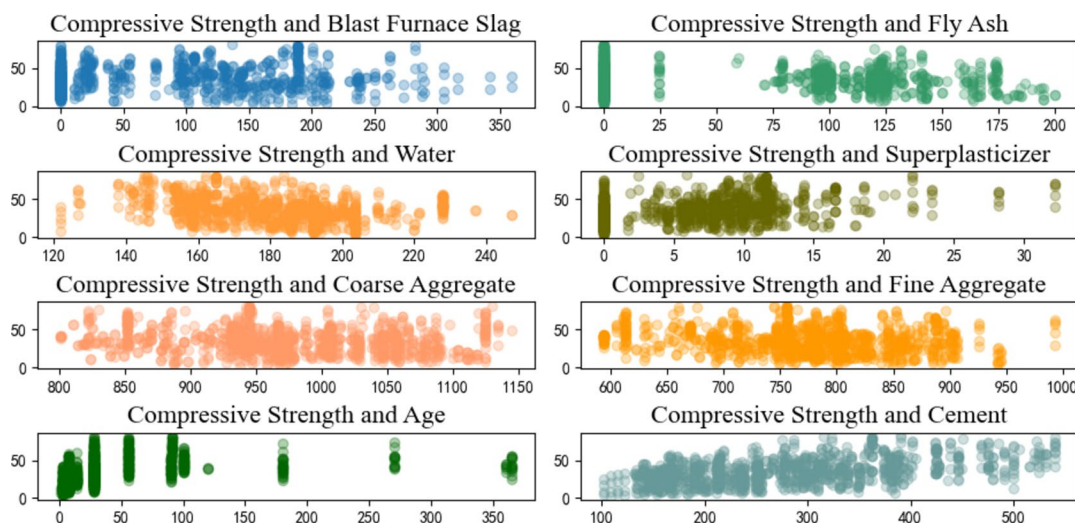

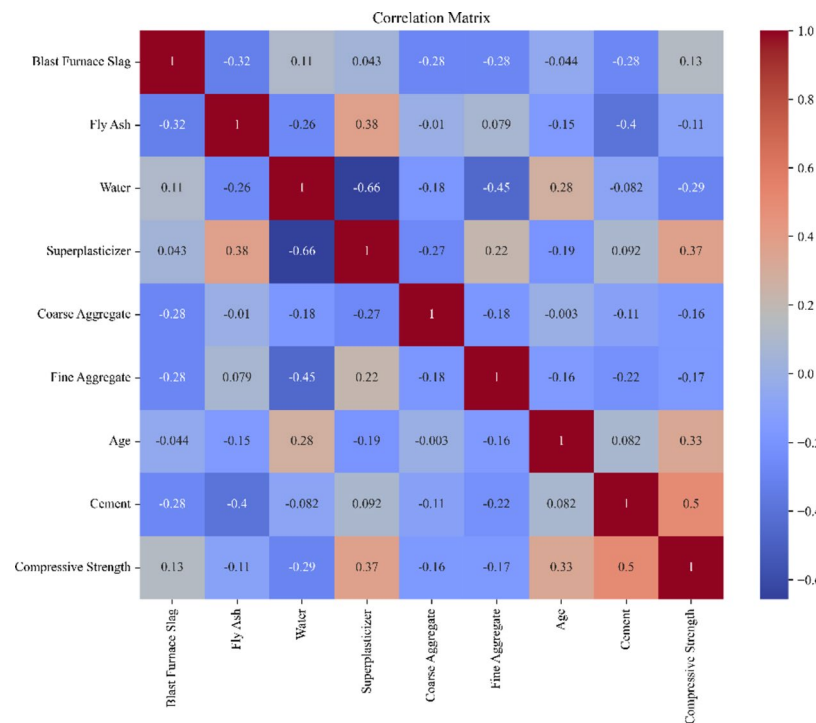**Figure 1.** Sample distribution of different factors relative to concrete compressive strength.

Figure 2. Correlation analysis of different factors. The meanings of different correlation coefficient values are as follows: 0.1–0.3 stands for weak correlation; 0.3–0.5 represents moderate correlation; 0.5–1.0 is strongly correlated.

$$x^* = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{1}$$

where x is the factor value of each sample, $\min(x)$, $\max(x)$ are the minimum and maximum values of the factor for all samples, respectively. x^* stands for the normalized factor for each sample.

RandomForest and ExtraTrees

Randomforest (RF) is an ensemble learning algorithm introduced by Breiman et al. in 2001, integrating multiple CART decision trees for prediction[28]. It excels in handling nonlinear, high-dimensional complex problems. To augment the number of decision trees in RF, researchers frequently employ multiple random vectors to construct each decision tree in RF. The algorithmic process of the random forest is as follows:

Input: Training dataset $D = \{(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)\}$, the number of features for each sample x_i is *N*, and the number of decision trees *T* is set in advance.

Training:

   for *t* from 1 to *T*:

   (1-1) Perform bootstrap sampling from training dataset *D*, and for each sample *x*, sample with a probability of *1/N* to establish a subsample set D_t.

   (1-2) Randomly select *n* (*n<<N*) features from the training dataset D_t, and then a decision tree *t* is constructed using a certain factor partitioning criterion (Gini_index).

Output:

   (2-1) For the unseen sample *x*, use each decision tree *t* for prediction and obtain the predicted result y_t.

   (2-2) For regression, the final prediction value is determined by the average of all decision trees.

Algorithm 1.   The algorithm of randomforest.

ExtraTrees is a variant of RF[29]. The distinction between ExtraTrees and RF lies in the determination of the training set for each decision tree: bootstrap sampling is used in RF, while the original training set is employed in ExtraTrees. Moreover, RF selects an optimal factor partition point, similar to the traditional decision tree. However, ExtraTrees will randomly select a factor value to partition the decision tree. Due to the random selection of factor partition points rather than the optimal position, the size of the decision tree generated by ExtraTrees is generally larger than that of the decision tree generated by RF. In other words, the variance of the model is further reduced relative to RF, but the bias is increased relative to RF. At certain times, the generalization ability of ExtraTrees is superior to that of RF.

Deepforest

Deepforest, also known as cascade forest, was initially proposed by Zhou et al. in 2017[25]. To comprehensively integrate the distinct advantages of RF and ExtraTrees, each layer of Deepforest comprises two RF-based models and two ExtraTrees-based models (Fig. 3). The prediction of concrete compressive strength for each cascade layer is the average of the four models.

During model training, the model performs layer-by-layer calculations, transferring the calculation results from the previous layer to the next layer, and assessing the difference between the prediction error of each layer and the preceding layer. If the prediction error of the current layer decreases compared to the previous layer, the
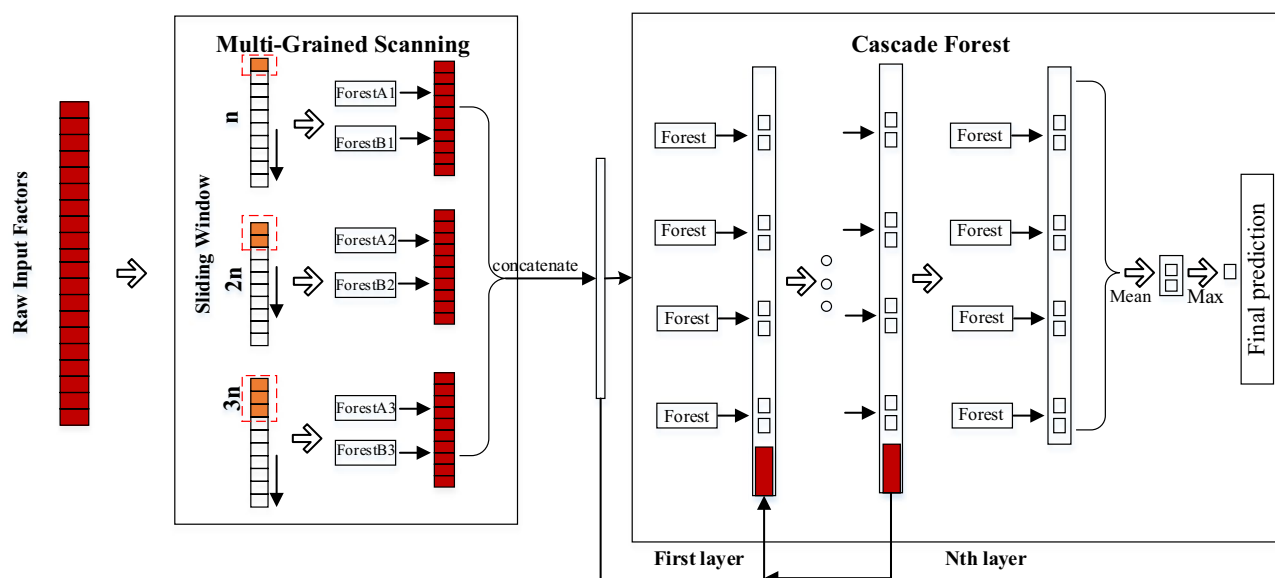

Figure 3.   Schematic diagram of Deepforest.

calculation of the next layer continues. However, if the prediction error of this layer is higher than that of the previous layer, the calculation of the next layer is halted. Subsequently, the layer with the lowest prediction error is selected as the output layer to generate the prediction result.

The optimal number of the tth layer of Deepforest should minimize the prediction error of the model when the tth layer was served as the output layer. Therefore, the optimal level number $\hat{t}$ of Deepforest should be obtained by the following formula.

$$
\begin{aligned}
\min_t MSE &= \frac{1}{N} \sum_{i=1}^{N} \left(y_i - G(\boldsymbol{x_i}) \right)^2 \\
s.t. \quad G(\boldsymbol{x_i}) &= \begin{cases} C_1(\boldsymbol{x_i}), & t = 1 \\ C_t([\boldsymbol{x_i}, C_{t-1}(\boldsymbol{x_i})]), & t > 1 \end{cases} \\
C_t(\boldsymbol{x_i}) &= \frac{1}{4} \left[F_{t,1}(\boldsymbol{x_i}) + F_{t,2}(\boldsymbol{x_i}) + CF_{t,1}(\boldsymbol{x_i}) + CF_{t,2}(\boldsymbol{x_i}) \right]
\end{aligned}
\tag{2}
$$

where MSE is the mean square error, $\boldsymbol{x_i}$ stands for the ith sample, y_i represents the label of the ith sample, N is the size of the training dataset. $G(\boldsymbol{x_i})$ is the predicted result on the ith sample. $C_t(\boldsymbol{x_i})$ is the prediction result of the ith sample at the tth layer. Each layer was made up of two Randomforests $\left(F_{t,1} \text{ and } F_{t,2} \right)$ and two ExtraTrees $\left(CF_{t,1} \text{ and } CF_{t,2} \right)$. For each sample $\boldsymbol{x_i}$, the prediction of concrete compressive strength of each cascade layer $C_t(\boldsymbol{x_i})$ is the average of the above-mentioned four models.

Performance evaluation

12 different regressors, including Linear[30], KNN[31], DT[32], SVR[33], Least absolute shrinkage and selection operator (LASSO)[34], Multilayer Perceptron (MLP)[35], ExtraTrees[29], RF[28], AdaBoost[36], GradientBoosting[37], Bagging[38] and Deepforest[25], were implemented using Python. For model training, Grid search[39] was applied to find the optimized hyperparameter of the different models.

Fivefold cross-validation was employed to assess the performance of different models. The training dataset is approximately divided into 5 folds, wherein each fold uses 4 folds for training, leaving one fold for testing. Subsequently, the average results of the 5 tests are computed to evaluate the model's performance. For a more objective evaluation of different models, this study derived 4 standard indicators: coefficient of determination (R^2)[40], mean square error (MSE)[41], mean absolute error (MAE)[42], and root mean square error (RMSE)[43]. Among these indicators, the range of R^2 is $(-\infty, 1]$, and the compressive strength predictor achieves optimal performance when the values of these indicators are equal to 1. Regarding MSE, MAE, and RMSE, the closer the value is to 0, the smaller the prediction error. The indicators can be defined as follows.

$$
\begin{cases}
R^2 = 1 - \dfrac{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{\frac{1}{N} \sum_{i=1}^{N} (y_i - \bar{y})^2} \\
MSE = \dfrac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \\
MAE = \dfrac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i| \\
RMSE = \sqrt{\dfrac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2}
\end{cases}
\tag{3}
$$

where $\hat{y}_i$ and y_i are the predicted and actual value of the ith sample, respectively. $\bar{y}$ is the average compressive strength of all samples. N is the size of the training dataset. Using graphic approaches can provide an intuitive vision to study concrete compressive strength prediction. Thus, the framework of our model is shown in Fig. 4.
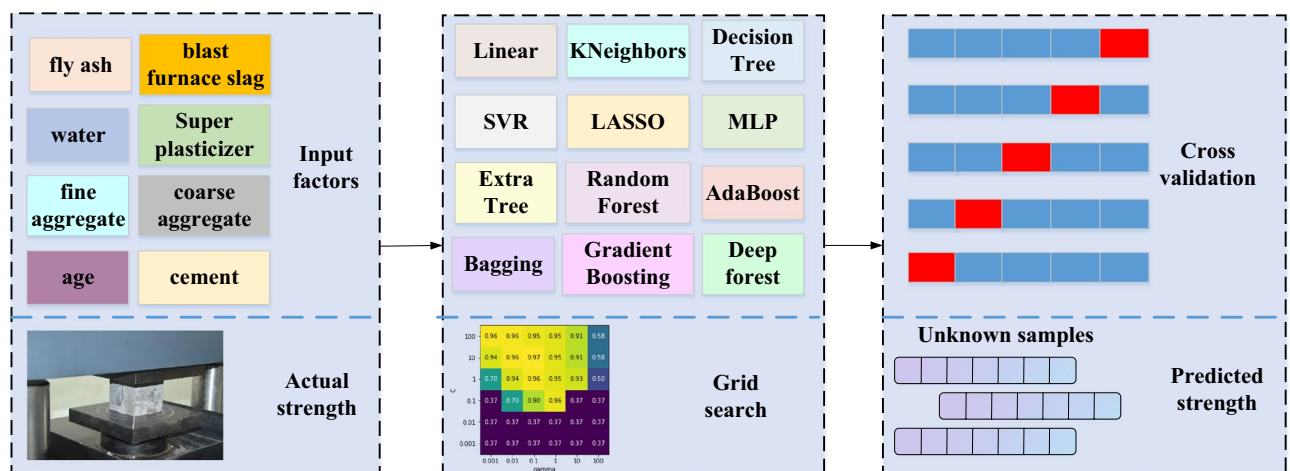

Figure 4. The framework of our model.

Results

Model performance with different characteristic factors

To determine the optimized factor subset, various combinations of factors, blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, fine aggregate, age, and cement, were employed to train a DT-based model on the training dataset. For the evaluation of our model with different factor subsets, fivefold cross-validation was utilized, and standard indicators were introduced for comparison. As indicated in Table 2, our model achieved the best performance with an R^2 of 0.39 when using the age factor alone for model training. This outcome underscores the significance of age as the most crucial factor in predicting the compressive strength of concrete. In the early stages of age, the compressive strength of concrete experiences rapid growth, but this growth gradually slows down with the extension of age. Moreover, the water and cement factors also exhibited superior performance compared to other factors, with R^2 values of 0.22 and 0.24, respectively. The pronounced impact of water and cement on predicting the compressive strength of concrete can be attributed to the hydration reaction between water and cement, leading to the formation of hardened cement stone that binds the aggregates, resulting in concrete with a certain strength. Consequently, the amount of water used and its compatibility with cement significantly influence the compressive strength of concrete, necessitating their consideration in prediction models. Through the judicious selection and control of factors such as water-cement ratio, cement variety, and strength grade, the strength and durability of concrete can be enhanced. Furthermore, despite the lowest correlation coefficient of -0.29 between the water factor and compressive strength, it outperformed blast furnace slag, fly ash, superplasticizer, coarse aggregate, and fine aggregate in terms of prediction performance. This indicates a negative correlation between water factors and compressive strength, implying that compressive strength decreases with increasing water content. Our model achieved the highest R^2 of 0.83 using the blast furnace slag, superplasticizer, age, cement, and water factors, surpassing the performance of the single-factor predictor (with the age factor achieving the highest R^2 of 0.39). These results underscore the necessity of comprehensively considering the influence of various factors when predicting concrete strength. Consequently, the blast furnace slag, superplasticizer, age, cement, and water factors were identified as the optimized subset for concrete compressive strength prediction.

Model performance on the training dataset

To obtain the best regressor, the training dataset was used to train the above-mentioned 12 regressors (Linear, KNN, DT, SVR, LASSO, MLP, ExtraTrees, RF, AdaBoost, GradientBoosting, Bagging and Deepforest) and the optimized feature subset, blast furnace slag, superplasticizer, age, cement, and water, was selected into the 12 different regressors. To compare the performance of different regressors fairly, fivefold cross-validation was used on the training dataset with the same fold. The optimized parameters of each model were shown in Table 3.

To identify the best regressor, the aforementioned 12 regressors were trained on the entire training dataset with their optimized parameters. As indicated in Table 4, the Deepforest-based regressor exhibited the best performance with an R^2 of 0.91, MSE of 26.32, MAE of 3.60, and RMSE of 5.11, respectively. Additionally, the RF-based model secured a second-place score with an R^2 of 0.90, MSE of 28.00, MAE of 3.71, and RMSE of 5.26, respectively. Furthermore, it was observed that the ANN-based model, MLP, achieved the poorest performance among all models, suggesting that the neural network-based model is not suitable for addressing this issue.

Model comparison on the independent testing dataset

To further evaluate the performance of our model, the whole training dataset was used to train the 12 regressors with their best hyperparameters, and then the performance of each model were compared on the independent testing dataset. As shown in Table 5, the Deepforest-based model achieved the highest R^2 of 0.91. What's more, we also noticed that the tree-based models, ExtraTrees, RF, Bagging and Deepforest, achieved higher performance (R^2 greater than 0.90) than the LASSO-based model (got the lowest R^2 of 0.61, MSE of 109.09, MAE of 8.22, and RMSE of 10.44), which suggest that the tree-based models are better suited to predict concrete compressive

Factor	R^2	MSE	MAE	RMSE
BFS	0.13	244.58	12.72	15.63
Fly ash	-0.01	287.94	13.66	16.94
Water	0.22	219.77	12.12	14.82
SP	0.09	256.92	13.22	16.00
CA	0.11	249.98	12.54	15.79
FA	0.15	239.72	12.46	15.47
Age	0.39	173.16	10.45	13.14
Cement	0.24	215.39	12.01	14.65
BFS + SP + age + cement	0.81	52.42	4.83	7.21
BFS + SP + age + cement + water	**0.83**	**47.07**	**4.68**	**6.83**
BFS + SP + age + cement + water + CA + FA	0.82	49.55	4.62	7.00
ALL	0.83	47.69	4.52	6.86

Table 2. Performance comparison with different factor combinations. BFS, SP CA, FA stands for blast furnace slag, superplasticizer, coarse aggregate, fine aggregate, respectively. Significant values are in bold.

Regressor	Parameter range	Optimized parameters
Linear	–	–
KNeighbors	n_neighbors:[0, 100]	4
DecisionTree	max_depth:[0, 100]	36
SVR	C:[0, 10], gamma:[0, 10]	C:9.8, gamma:0.3
LASSO	alpha:np.logspace(− 4, 0, 50)	0.13
MLP	hidden_layer_sizes:(10,10) ~ (100,100)	(50, 50)
ExtraTrees	n_estimators:[200, 400]	330
RandomForest	n_estimators:[200, 400]	200
AdaBoost	n_estimators:[200, 400]	240
GradientBoosting	n_estimators:[200, 400]	100
Bagging	n_estimators:[200, 400]	290
Deepforest	layers:[1, +∞]	2

Table 3. Parameter optimization on the training dataset using fivefold cross-validation.

Regressor	R^2	MSE	MAE	RMSE
Linear	0.56	123.63	8.91	11.11
KNeighbors	0.75	69.83	6.31	8.35
DecisionTree	0.82	49.74	4.72	7.04
SVR	0.66	97.73	7.66	9.87
LASSO	0.55	129.25	9.19	11.35
MLP	0.42	162.49	10.23	12.74
ExtraTrees	0.75	68.32	5.37	8.23
RandomForest	0.90	28.00	3.71	5.26
AdaBoost	0.78	61.21	6.35	7.80
GradientBoosting	0.90	28.98	3.96	5.36
Bagging	0.89	32.42	4.00	5.66
Deepforest	**0.91**	**26.32**	**3.60**	**5.11**

Table 4. Performance comparison with different regressors on the training dataset. Significant values are in bold.

Regressor	R^2	MSE	MAE	RMSE
Linear	0.61	108.23	8.18	10.40
KNeighbors	0.79	57.18	5.55	7.56
DecisionTree	0.86	38.88	3.76	6.24
SVR	0.83	47.65	4.96	6.90
LASSO	0.61	109.09	8.22	10.44
MLP	0.72	78.88	6.76	8.88
ExtraTrees	0.90	26.52	3.10	5.15
RandomForest	0.90	26.87	3.45	5.18
AdaBoost	0.78	60.98	6.25	7.81
GradientBoosting	0.89	31.50	3.90	5.61
Bagging	0.90	27.08	3.47	5.20
Deepforest	**0.91**	**26.10**	**3.18**	**5.11**

Table 5. Performance comparison with different regressors on the independent testing dataset. Significant values are in bold.

strength. The reason for this result is that tree-based models can automatically select the most relevant features for prediction, which is particularly useful for determining which features are most important for prediction. In addition, tree-based models have a certain degree of robustness against noise and outliers, as they perform data filtering and dimensionality reduction during the process of constructing trees.

The absolute difference for each sample in the independent testing dataset was shown in Fig. 5. For 31.3% (65/208) of the samples, the absolute difference between the predicted strength and the true strength was within 1.
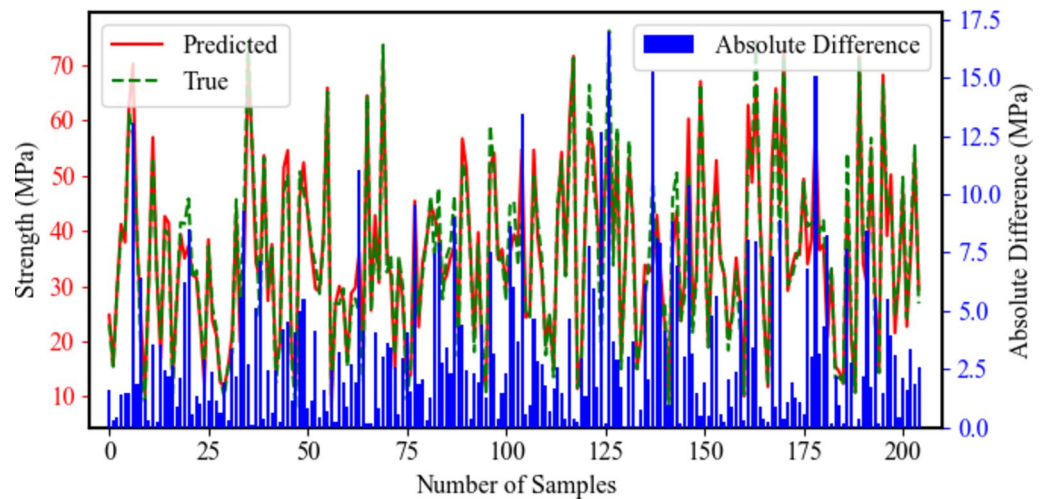
Figure 5. Absolute difference for each sample in the independent testing dataset.

For 80.3% (167/208) of the samples, the absolute difference between the predicted strength and the true strength was within 5. Only 5.8% (12/208) of the samples had an absolute difference greater than 10 between their true strength and predicted strength, indicating the superiority of our model.

Compared with the state-of-the-art predictors

To further explore the performance of the Deepforest-based regressor, we compared our model with the State-of-the-art predictors. These models are constructed based on GEP[17], RF[21], SVR[20], and ELM[24], respectively. For a fair comparison, all models were trained on the same training dataset and tested on the same independent test dataset. Although these models target concrete materials and have different input factors, we compared the performance of the models on our optimized subset. As shown in Fig. 6, the correlation coefficient of our model is greater than 0.95, and the central RMS error and standard deviation are the smallest among the five models, which suggest that our model is most suitable for concrete strength prediction.
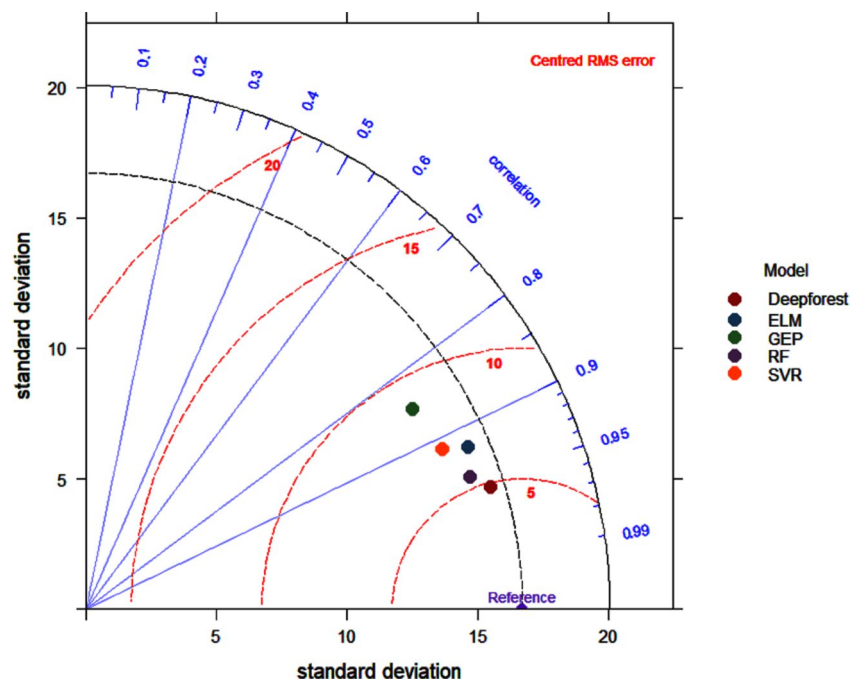

Figure 6. Performance comparison with the State-of-the-art Predictors.

Performance validation using existing data

Due to the difficulty in obtaining experimental data, it is difficult to find a dataset that is completely consistent with our modeling dataset. In this study, we used the dataset from[44] for validation, which collected compressive strength data of the machine-made sand concrete mixed with different amounts of super absorbent polymer. Although the dataset is not exactly the same as our modeling dataset, it contains four of our optimal subsets, namely superplasticizer, age, cement, and water factors. Therefore, we reconstructed the concrete strength prediction model on our training dataset based on these four factors and random selected 15 of their test data for prediction. As shown in Table 6, our prediction bias are within 13% for samples with an age of 3 or 7, while our prediction ability is limited, with a bias of around 20% for samples with an age of 28. These results indicate that despite the different additives in concrete, our model still has a certain predictive ability. However, different additives may cause certain changes in the mechanical properties of concrete. Therefore, establishing specific models for different additives can better predict the compressive strength of concrete[45,46]. For the convenience of researchers, we packaged the model developed in this article using wxpython and pyinstaller. The software includes Feature selection, Regressor selection, Model construction, Model validation and Sample Prediction modules (https://github.com/NWAFU-LiuLab/CCS-Predictor). Researchers can not only use our software for prediction, but also use it for secondary development, exploring optimization combinations of different training datasets, features, and regressors. It should be noted that due to the different random number seeds used to partition the training and testing sets, the software's predicted results may differ slightly from the results in the article.

Limitations of the study and recommendation for future research

However, our method still has some limitations. Firstly, the dataset utilized in this study dates back to 2007, and the recent trend of using auxiliary cement-based materials (SCM) may change the dataset and thus alter the prediction results[47]. Secondly, only a feature subset was used for modeling, and complex feature engineering was not undertaken in this study. For small datasets, considering the use of pre-trained models in other related fields (such as models trained on other concrete datasets) for transfer learning is possible. Transfer learning can expedite model convergence on new datasets and may enhance prediction accuracy. Although the raw data may lack complex features, more meaningful features can be extracted through feature engineering techniques such as principal component analysis, feature combination, feature selection, etc. Additionally, considering the incorporation of other characteristics related to concrete strength, such as the chemical composition of raw materials, certain parameters during the production process, etc., is feasible. However, it should be noted that these methods may not necessarily be effective in all situations, so in practical applications, selection and adjustment may need to be based on specific circumstances.

Conclusions

So far, research on concrete strength prediction based on machine learning has made some progress, but the identification accuracy remains insufficient, and the hyperparameter adjustment process of the model is complex. Consequently, a Deepforest-based model was developed in this study. For model training, blast furnace slag, superplasticizer, age, cement, and water factors were selected for the aforementioned 12 regressors, and grid search was employed for hyperparameter optimization. Rigorous validation experiments led to the selection of Deepforest as the optimized regressor, achieving an R^2 of 0.91 on the independent testing dataset. The results demonstrate the effectiveness of our model in predicting concrete strength.

According to the experimental results, age is the most crucial factor affecting the strength of concrete, followed by cement. With the addition of other factors, the prediction accuracy of concrete strength gradually improves. Among them, blast furnace slag, superplasticizer, age, cement, and water factors proved to be the best indexes for

No	Age	Cement	Water	SP	True Strength	Predicted Strength	Percent (%)
1	3	413	190.26	0.91	34.44	37.19	8
2	3	413.7	190.27	0.91	38.51	37.19	− 3
3	3	413.3	190.27	0.91	34.47	37.19	8
4	7	440.6	190.81	0.91	41.68	44.49	7
5	7	454.7	191.09	0.91	41.82	44.43	6
6	7	441	190.82	0.91	42.43	44.55	5
7	7	427	190.54	0.91	42.6	43.52	2
8	7	426.6	190.53	0.91	42.3	43.07	2
9	7	427.3	190.55	0.91	44.24	43.87	− 1
10	7	413	190.26	0.91	42.82	38.89	− 9
11	7	413.3	190.27	0.91	43.01	38.89	− 10
12	7	413.7	190.27	0.91	44.57	38.89	− 13
13	28	453.9	191.08	0.91	52.47	42.22	− 20
14	28	440.1	190.8	0.91	53.37	41.69	− 22
15	28	454.3	191.09	0.91	55.26	42.22	− 24

Table 6. Concrete compressive strength prediction using other existing data.

predicting the strength of concrete. However, it was observed that R^2 did not improve when coarse aggregate and fine aggregate were added, indicating that these two factors have no significant effect on enhancing the prediction accuracy of concrete. Compared with other models, the Deepforest-based regressor achieved superior results on the training set and the independent testing dataset and does not require explicit hyperparameter adjustments, reducing the difficulty of model optimization.

In this study, machine learning algorithms are utilized to learn and analyze a substantial amount of experimental data. The advantages of this model include: (1) Establishing a more accurate concrete strength prediction model, thereby enhancing the accuracy of concrete strength prediction; (2) Determining the concrete mix proportion more precisely, thereby optimizing the performance and cost of concrete; (3) Controlling key parameters more accurately in the production process, thereby improving production efficiency and quality. Additionally, implementing machine learning-based models for predicting concrete strength in the real world may also require addressing the following challenges: (1) High-quality data is the foundation of machine learning algorithms, so ensuring the accuracy and completeness of experimental data is necessary; (2) Different machine learning algorithms may be suitable for different datasets and problems, so choosing the appropriate algorithm based on the actual situation is necessary. (3) Machine learning models require extensive training and validation to achieve high prediction accuracy, necessitating sufficient time and resources to be invested. (4) Applying machine learning-based concrete strength prediction models in the real world also requires considering various challenges in practical applications, such as data collection, model deployment, and maintenance.

Data availability

The dataset used in this study and the source code are available at https://github.com/NWAFU-LiuLab/CCS-Predictor.

References

1. Chopra, P., Sharma, R. & Kumar, M. Artificial neural networks for the prediction of compressive strength of concrete. *Int. J. Appl. Sci. Eng.* **13**(3), 187–204 (2015).
2. Monteiro, P., Miller, S. & Horvath, A. Towards sustainable concrete. *Nat. Mater.* **16**(7), 698–699 (2017).
3. Atici, U. Prediction of the strength of mineral admixture concrete using multivariable regression analysis and an artificial neural network. *Expert Syst. Appl.* **38**(8), 9609–9618 (2011).
4. Helal, J., Sofi, M. & Mendis, P. Non-destructive testing of concrete: A review of methods. *Electron. J. Struct. Eng.* **14**(1), 97–105 (2015).
5. Ji, Y. *et al.* A state-of-the-art review of concrete strength detection/monitoring methods: With special emphasis on PZT transducers. *Constr. Build. Mater.* **362**, 129742 (2023).
6. Breysse, D. & Juan, L. Assessing concrete strength with rebound hammer: Review of key issues and ideas for more reliable conclusions. *Mater. Struct.* **47**(9), 1589–1604 (2014).
7. Nik, A. & Omran, O. Estimation of compressive strength of self-compacted concrete with fibers consisting nano-SiO_2 using ultrasonic pulse velocity. *Constr. Build. Mater.* **44**, 654–662 (2013).
8. Oh, J. H. *et al.* Aggregate effects on γ-ray shielding characteristic and compressive strength of concrete. *J. Nucl. Fuel Cycle Waste Technol.* **14**(4), 357–365 (2016).
9. Xu, T. & Li, J. Assessing the spatial variability of the concrete by the rebound hammer test and compression test of drilled cores. *Constr. Build. Mater.* **188**, 820–832 (2018).
10. Zheng, Y. *et al.* Experimental investigation of concrete strength curve based on pull-out post-insert method. *Int. J. Distrib. Sens. Netw.* **16**(7), 155014772094402 (2020).
11. Liu, Q. *et al.* Revised the formula of solidification coefficient in continuous casting based on nail-shooting and simulation. *Metal. Int.* **16**(11), 150–154 (2011).
12. Ebadi-Jamkhaneh, M. & Ahmadi, M. Comprehensive investigations of the effect of bolt tightness on axial behavior of a MERO joint system: Experimental, FEM, and soft computing approaches. *J. Struct. Eng.* **12**, 147 (2021).
13. Feng, L. Y. Q. Machine learning-based integration of remotely-sensed drought factors can improve the estimation of agricultural drought in South-Eastern Australia. *Agric. Syst.* **173**, 303–316 (2019).
14. Liu, Z. *et al.* m6Aminer: Predicting the m6Am sites on mRNA by fusing multiple sequence-derived features into a CatBoost-based classifier. *Int. J. Mol. Sci.* **24**, 7878 (2023).
15. Liu, Z. *et al.* HLMethy: A machine learning-based model to identify the hidden labels of m6A candidates. *Plant Mol. Biol.* **101**(6), 575–584 (2019).
16. Ahmadi, M. & Kioumarsi, M. Predicting the elastic modulus of normal and high strength concretes using hybrid ANN-PSO. *Materials Today: Proceedings* IN PRESS (2023).
17. Gholampour, A., Gandomi, A. H. & Ozbakkaloglu, T. New formulations for mechanical properties of recycled aggregate concrete using gene expression programming. *Constr. Build. Mater.* **130**, 122–145 (2017).
18. Javed, M. F. *et al.* Applications of gene expression programming and regression techniques for estimating compressive strength of bagasse ash based concrete. *Crystals* **10**, 737 (2020).
19. Yang, D. W. *et al.* Compressive strength prediction of concrete blended with carbon nanotubes using gene expression programming and random forest: Hyper-tuning and optimization. *J. Mater. Res. Technol.* **24**, 7198–7218 (2023).
20. Ling, H. *et al.* Combination of support vector machine and K-Fold cross validation to predict compressive strength of concrete in marine environment. *Constr. Build. Mater.* **206**, 355–363 (2019).
21. Farooq, F. *et al.* Predictive modeling for sustainable high-performance concrete from industrial wastes: A comparison and optimization of models using ensemble learners. *J. Clean. Prod.* **292**, 126032 (2021).
22. Yeh, I. C. Modeling of strength of high-performance concrete using artificial neural networks. *Cem. Concr. Res.* **28**(12), 1797–1808 (1998).
23. Keshavarz, Z. & Torkian, H. Application of ANN and ANFIS models in determining compressive strength of concrete. *J. Soft Comput. Civ. Eng.* **2**(1), 62–70 (2018).
24. Al-Shamiri, A. K. *et al.* Modeling the compressive strength of high-strength concrete: An extreme learning approach. *Constr. Build. Mater.* **208**, 204–219 (2019).
25. Zhou, Z. & Feng, J. Deep Forest: Towards an alternative to deep neural networks. https://doi.org/10.48550/arXiv.1702.08835 (2017).

26. Tan, Z. *et al.* A system for denial-of-service attack detection based on multivariate correlation analysis. *IEEE Trans. Parallel Distrib. Syst.* **25**(2), 447–456 (2014).
27. Bolstad, B. M. *et al.* A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* **19**(2), 185–193 (2003).
28. Liaw, A. & Wiener, M. Classification and regression by RandomForest. *R News* **23**(23), 18–22 (2002).
29. Geurts, P., Ernst, D. & Wehenkel, L. Extremely randomized trees. *Mach. Learn.* **63**(1), 3–42 (2006).
30. Preacher, K., Curran, P. & Bauer, D. Computational tools for probing interaction effects in multiple linear regression, multilevel modeling, and latent curve analysis. *J. Educ. Behav. Stat.* **31**, 427–448 (2006).
31. Blough, D. *et al.* The k-neighbors approach to interference bounded and symmetric topology control in ad hoc networks. *IEEE Trans. Mob. Comput.* **5**(9), 1267–1282 (2006).
32. Myles, A. *et al.* An introduction to decision tree modeling. *J. Chemom.* **18**, 275–285 (2004).
33. Lu, H. *et al.* A novel method for gaze tracking by local pattern model and support vector regressor. *Signal Process.* **90**(4), 1290–1299 (2010).
34. Meinshausen, N. & Buehlmann, P. High-dimensional graphs and variable selection with the Lasso. *Anna. Stat.* **34**(3), 1436–1462 (2006).
35. Pal, S. & Mitra, S. Multilayer perceptron, fuzzy sets, and classification. *IEEE Trans. Neural Netw.* **3**(5), 683 (1992).
36. Li, X., Wang, L. & Sung, E. AdaBoost with SVM-based component classifiers. *Eng. Appl. Artif. Intell.* **21**(5), 785–795 (2008).
37. Alexey, N. & Alois, K. Gradient boosting machines, a tutorial. *Front. Neurorobotics* **7**, 21 (2013).
38. Bauer, E. & Kohavi, R. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Mach. Learn.* **36**, 105–139 (1999).
39. Cheung, K., Langevin, A. & Delmaire, H. Coupling genetic algorithm with a grid search method to solve mixed integer nonlinear programming problems. *Comput. Math. Appl.* **34**(12), 13–23 (1997).
40. Nagelkerke, N. J. D. A note on a general definition of the coefficient of determination. *Biometrika* **78**(3), 691–692 (1991).
41. Savaux, V. & Bader, F. Mean square error analysis and linear minimum mean square error application for preamble-based channel estimation in orthogonal frequency division multiplexing/offset quadrature amplitude modulation systems. *IET Commun.* **9**(14), 1763–1773 (2015).
42. Coyle, E. J. & Lin, J. H. Stack filters and the mean absolute error criterion. *IEEE Trans. Acoust. Speech Signal Process.* **36**(8), 1244–1254 (1988).
43. Hancock, G. R. & Freeman, M. J. Power and sample size for the root mean square error of approximation test of not close fit in structural equation modeling. *Educ. Psychol. Meas.* **61**(5), 741–758 (2001).
44. Chen, H. *et al.* Based on GA-BP neural network prediction of compressive strength of machine-made sand concrete with SAP internal curing. *Concrete* **5**, 72–76 (2023).
45. Moradi, M. J. *et al.* Predicting the compressive strength of concrete containing metakaolin with different properties using ANN. *Measurement* **183**, 109790 (2021).
46. Moradi, N. *et al.* Predicting the compressive strength of concrete containing binary supplementary cementitious material using machine learning approach. *Materials* **15**, 5336 (2022).
47. Farhangi, V. *et al.* Application of artificial intelligence in predicting the residual mechanical properties of fiber reinforced concrete (FRC) after high temperatures. *Constr. Build. Mater.* **411**, 134609 (2024).

Acknowledgements

Author contributions

W.Z. and Z.L. participated in conceiving and performing the experiments. J.G. and C.N. participated in analyzing the data. All authors contributed to the writing of the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Z.L.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.