Wiley
Advances in Civil Engineering
Volume 2024, Article ID 8048645, 15 pages
https://doi.org/10.1155/2024/8048645

WILEY

Research Article

Application of Fully Connected Neural Network-Based PyTorch in Concrete Compressive Strength Prediction

Xuwei Dong,[1] **Yang Liu**,[1] **and Jinpeng Dai**[2]

[1]*Key Laboratory of Opto-Electronic Technology and Intelligent Control, Ministry of Education, Lanzhou Jiaotong University, Lanzhou 730070, China*
[2]*National and Provincial Joint Engineering Laboratory of Road and Bridge Disaster Prevention and Control, Lanzhou Jiaotong University, Lanzhou 730070, China*

Correspondence should be addressed to Xuwei Dong; dxw007@lzjtu.edu.cn

Received 30 November 2023; Revised 30 April 2024; Accepted 10 May 2024; Published 31 May 2024

Academic Editor: Junwei Ma

Compressive strength of concrete is an important parameter in the design of concrete structures and the prediction of their durability. Therefore, it is of great significance to predict the compressive strength of concrete. In this study, a fully connected neural network model is developed using the PyTorch framework to predict the compressive strength of concrete and compared with six other machine learning models. These models are multiple linear regression, K-nearest neighbor regression, support vector machine, decision tree, random forest, light gradient boosting machine, and artificial neural network. The model is trained using 4,253 data with seven input parameters, including cement (C), fly ash (F), mineral powder (K), fine aggregate (FA), coarse aggregate (CA), water reducer admixture (WRA), and water (W). Three thousand six hundred twenty-one data in the datasets are used to train the prediction model after data cleaning, and 632 data are used to validate the model. The results show that the fully connected neural network model based on PyTorch frame can predict the compressive strength of concrete with higher accuracy. Therefore, it is a reliable and useful method to optimize the artificial network model. So, it has important application value in practice. The strength of concrete can be predicted in advance, making the project more efficient and reducing costs. Besides, by adjusting the mix ratio, combining the strength prediction results in different environments and industries to ensure the quality of construction.

1. Introduction

Among various performance indicators of concrete, compressive strength is the most fundamental because it directly relates to the structural safety of construction products [1]. Therefore, predicting concrete strength is greatly important. Within the first 3 days, 75% of the maximum strength of concrete is achieved, and the strength increases gradually with time, reaching 95% of the maximum strength at 28 days. Thus, the 28-day compressive strength is used for evaluation in this study. The traditional approach to obtain concrete compressive strength includes physical experiments, where cube or cylindrical specimens are prepared with a specific mix design, cured, and tested using a compression testing instrument. However, this experimental method is time-consuming and costly [2]. Researchers have proposed different machine learning methods to predict the compressive strength of concrete based on different component compositions and mix designs [3, 4, 5]. Moreover, research results show a nonlinear relationship between the content of each concrete component and its compressive strength, resulting in poor prediction results for both methods.

In recent years, with the continuous development of artificial intelligence technology, machine learning (ML) has been widely applied in various fields, and prediction using machine learning has become a mainstream method. For example, traditional natural disaster prediction is often based on experience and expert judgment, which has certain subjectivity and uncertainty. Machine learning models can improve the accuracy and reliability of early warning by training and learning from large amounts of data. Machine learning models can discover the rules and associations hidden behind huge data, so as to predict the

probability and degree of natural disasters more accurately and provide more accurate early warning information. Jia et al. [6], Long et al. [7], and Ma et al. [8] used machine learning methods to conduct multilevel comprehensive evaluation of natural disasters such as ground fractures and landslides and quantified the specific values, which can be used to monitor the activity evolution of different natural disasters. Liu et al. [9] proposed a reliable reservoir landslide displacement prediction method based on optimal support vector regression (EOA-SVR). Experiments show that EOA-SVR can stably provide reliable predictions while maintaining the optimal solution.

Meanwhile, machine learning is often used in the field of civil engineering. In this field, one of the most important types of research is the prediction of concrete strength through machine learning. Concrete strength prediction belongs to the application of regression models, where inputting the features into the model for learning and training can significantly improve the prediction accuracy, providing clear advantages over traditional experimental and statistical methods [10]. Machine learning has been developed for predicting concrete compressive strength for a long time, and different machine learning model algorithms have been used in various research studies. Li and Song [11] used four ensemble learning model algorithms of XGBoost, AdaBoost, GBDT, and RF to construct a model to predict the compressive strength of high performance concrete (HPC). They compared the GBDT model with the other three models and proved that the GBDT model achieved the highest prediction accuracy among the four machine learning models. Al-Shamiri et al. [12] developed a new model for predicting the compressive strength of HPC using the regularized extreme learning machines (RELM) and evaluated the prediction reliability of the developed RELM model through k-fold cross-validation. The results of the study demonstrated that the established RELM model accurately estimated the compressive strength of HPC. Cho et al. [13] estimated the compressive strength of high-strength concrete and recycled aggregate by non-destructive test and numerical analysis. Kumar et al. [14] achieved the highest accuracy in predicting lightweight concrete (LWC) using an optimized GPR model. This new model was economically practical, efficient, and suitable for application by researchers and engineers. Xia et al. [15] predicted the uniaxial compressive strength of concrete using different parameters, such as water–cement ratio, curing days, and substitution rate of recycled aggregate, as input data. The model effectively predicted the compressive strength of recycled aggregate concrete (RAC) and satisfied the engineering requirements. Meesaraganda et al. [16] developed a prediction model based on an artificial neural network (ANN) using 99 experimental data samples of fiber self-compacting concrete. Finally, they examined a feedforward three-layer backpropagation neural network with 10 hidden nodes using learning algorithms, providing realistic results. Ahmed et al. [17, 18] used different model methods in the field of machine learning to predict the compressive strength of concrete in different environments and different proportions and found that machine learning had the advantages of improving cost-effectiveness and saving time for concrete strength prediction. Mohammed et al. [19], Barkhordari et al. [20], and Jaf et al. [21] applied machine learning methods to predict the compressive strength of fly ash concrete. Emad et al. [22] used four models: linear relational model (LR), pure quadratic model (M5P-tree) and artificial neural model (ANN) to predict the compressive strength of high performance fiber concrete (UHPFRC). Mohammed et al. [23, 24] developed a systematic multiscale model to predict the strength of polymer-containing compressed concrete using a machine learning approach. At the same time, the effects of three kinds of water reducing agents on the compressive strength of concrete after curing from 1 to 28 days were studied. Piro et al. [25] used adaptive network based Fuzzy Inference System (ANFIS), namely ANN, multivariate adaptive regression spline (MARS) and M5P-tree model, to predict the compressive strength of concrete after steel slag aggregate replaced coarse aggregate. The results show that the model based on fuzzy adaptive network inference system is superior to other models in predicting the compressive strength of concrete after steel slag replacement, and the compressive strength of concrete after steel slag aggregate replacement is improved.

Although machine learning is widely used in areas such as geological disasters prediction and concrete compressive strength prediction, deep learning is rarely applied to the prediction of concrete strength in the related research. In general, machine learning algorithms usually require manual feature extraction and selection, which is a complex and time-consuming process. In contrast, deep learning algorithms are able to automatically learn feature representations from raw data, reducing human intervention. Besides, deep learning algorithms usually have stronger generalization ability than traditional machine learning algorithms, they are able to learn hierarchical representation of the data and capture more complex patterns, and machine learning is difficult to deal with complex data structures and patterns. Therefore, by comparing deep learning with traditional machine learning, a fully connected neural network model based on PyTorch framework is proposed in this paper. The results show that the fully connected neural network model based on PyTorch framework can accurately predict the compressive strength of concrete much better. Therefore, applying the related algorithms of deep learning to the field of civil engineering can better carry out engineering construction planning and save costs. At the same time, for different environments and different industries, the compressive strength of concrete can be quickly and accurately predicted to ensure the construction quality.

2. Traditional Machine Learning

2.1. Multivariate Linear Regression. The linear regression method in machine learning uses a linear prediction function to model the data and estimate based on the output parameters of the data. The concept of multivariable linear regression multiple linear regression (MLR) has emerged given that regression analysis often involves multiple input variables in

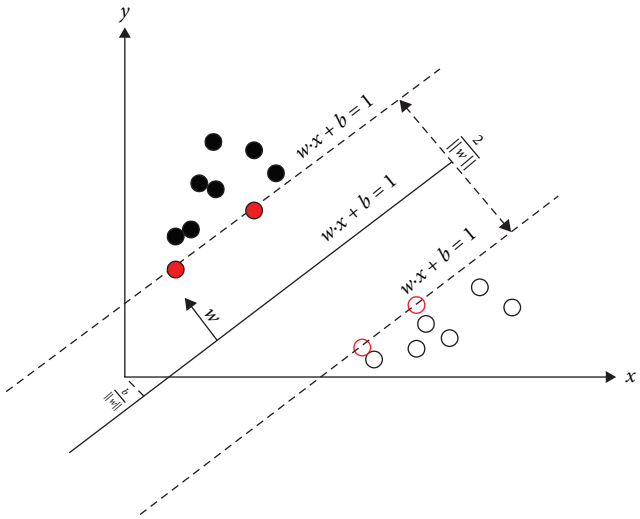

Figure 1: Support vector machine algorithm structure.

many applications. The equation provides the general form of the multivariable linear regression model, as follows [26]:

$$\widehat{Y} = a_0 + \sum_{j=1}^{m} a_j X_j, \tag{1}$$

where Y is the output of the model, X_j is the independent input variables of the model, and $a_0, a_1, a_2, \ldots, a_m$ represent the regression coefficients. The parameters are trained to obtain similar model output to that of the training datasets [27].

2.2. K-Nearest Neighbor (KNN) Regression. KNN regression is a machine learning algorithm. Whilst K-nearest neighbor (KNN) is commonly used for classification problems, it can also be applied to regression problems. It is a basic algorithm often used for comparison prior to applying more complex and advanced methods. The algorithm finds the K-nearest examples to a query by measuring the distance between the query and all examples in the data and then averages the labels in the case of regression problems [28].

2.3. Support Vector Machines (SVMs). Support vector machine (SVM) are commonly used for classification problems. However, to utilize SVM for regression fitting, Vapnik et al. [29] introduced the insensitive loss function based on SVM classification, that is the SVM for regression (SVR). When SVM is applied to regression fitting, the basic idea is no longer to find an optimal classification plane to separate two classes but to find an optimal classification plane that minimizes the error of all training samples regarding this plane. The algorithm structure of SVMs is shown in Figure 1.

2.4. Decision Trees. Decision trees (DTs) classify instances by arranging them from the root node to a leaf node. Each nonleaf node represents a test on an attribute, and each branch represents one of the test results. Each leaf node represents a class label, simulating the tree structure to establish a classification or regression model. A DT is a tree-like structure resembling a flowchart. It adopts a top–down recursive approach, starting from the root node of the tree and making attribute value comparisons at its internal nodes until a conclusion at the leaf nodes is reached. The algorithm structure of DTs is shown in Figure 2.

2.5. Random Forest. Random forest (RF) is an ensemble learning method in machine learning [30, 31]. It consists of multiple DTs, and the results are collected by randomly selecting features for each tree. The final result is obtained by majority voting or averaging, depending on the problem. In a set of input data $\{H(x, \theta_i), i = 1, 2\ldots, k\}$ with the prediction value of a single DT of $\{H(x, \theta_i)\}$, the final result of the RF prediction model is the average of the predictions of all DTs [32]. The algorithm structure of RF is shown in Figure 3.

2.6. Light Gradient Boosting Machine (LightGBM). Light gradient boosting machine (LightGBM) is a GBDT algorithm framework developed by Microsoft. It supports efficient parallel training. Generally, as the number of training data increases, the efficiency and accuracy of the training decrease. To address this issue, the gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB) are applied to LightGBM. GOSS allows the use of large-scale datasets, and EFB manages many data features. In addition, the algorithm employs leaf-wise tree growth, which is 20 times faster than traditional methods [33]. The algorithm structure of LightGBM is shown in Figure 4.

3. Fully Connected Neural Network-Based on the PyTorch Framework

PyTorch is a framework for building deep learning models developed by Facebook AI Research and several other labs. In recent years, with the continuous development of artificial intelligence, deep learning has been extensively studied, leading to the emergence of a large number of machine learning and deep learning tools. Many popular deep learning frameworks are currently available, namely, Caffe [34], CNTK [35], TensorFlow [36], and Theano [37]; they construct static data flow graphs representing computations that can be repeatedly used with batch data. In theory, this approach can improve model performance and enhance generalization capabilities. However, this approach sacrifices ease of use, ease of debugging, and flexibility in expressing computation types when improving model performance. Therefore, researchers aim to achieve dynamic changes within the framework, thereby creating significant value for deep learning. PyTorch emerged to enable dynamic execution without sacrificing performance by carefully implementing and designing options. This study uses PyTorch, which performs dynamic tensor computations using automatic differentiation and GPU acceleration while maintaining performance comparable to the fastest current deep learning libraries [38].

A fully connected neural network, also known as a dense network or a multilayer perceptron (MLP), consists of input nodes, hidden layers, and output nodes. Each node in a layer is connected to every node in the next layer, creating a dense

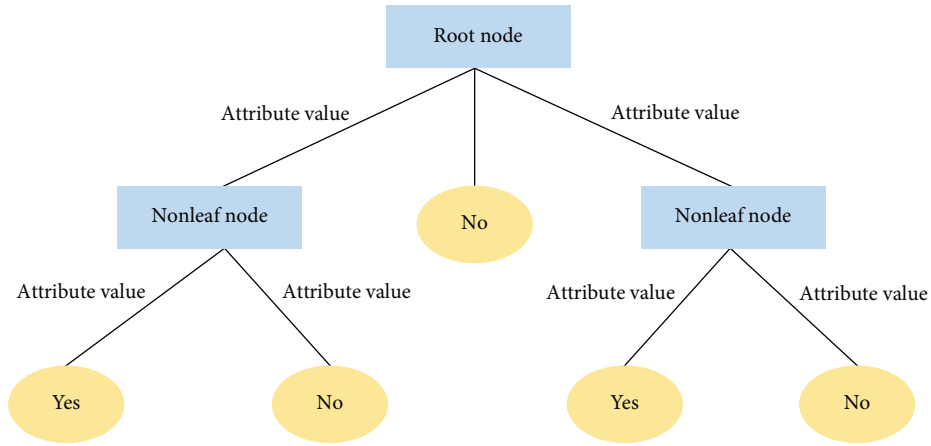

Figure 2: Decision tree algorithm structure.

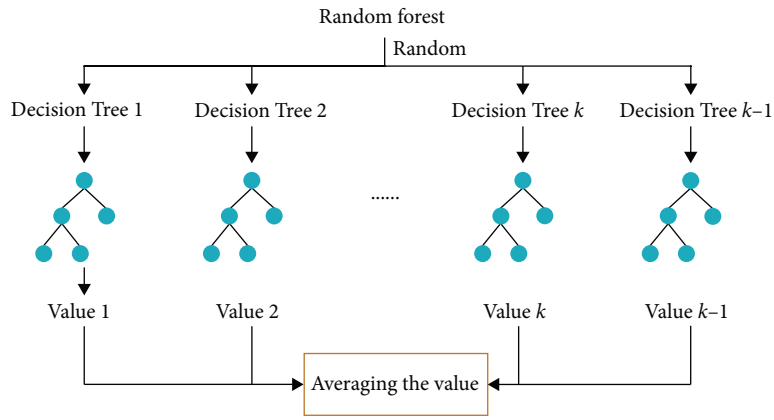

Figure 3: Random forest algorithm structure.

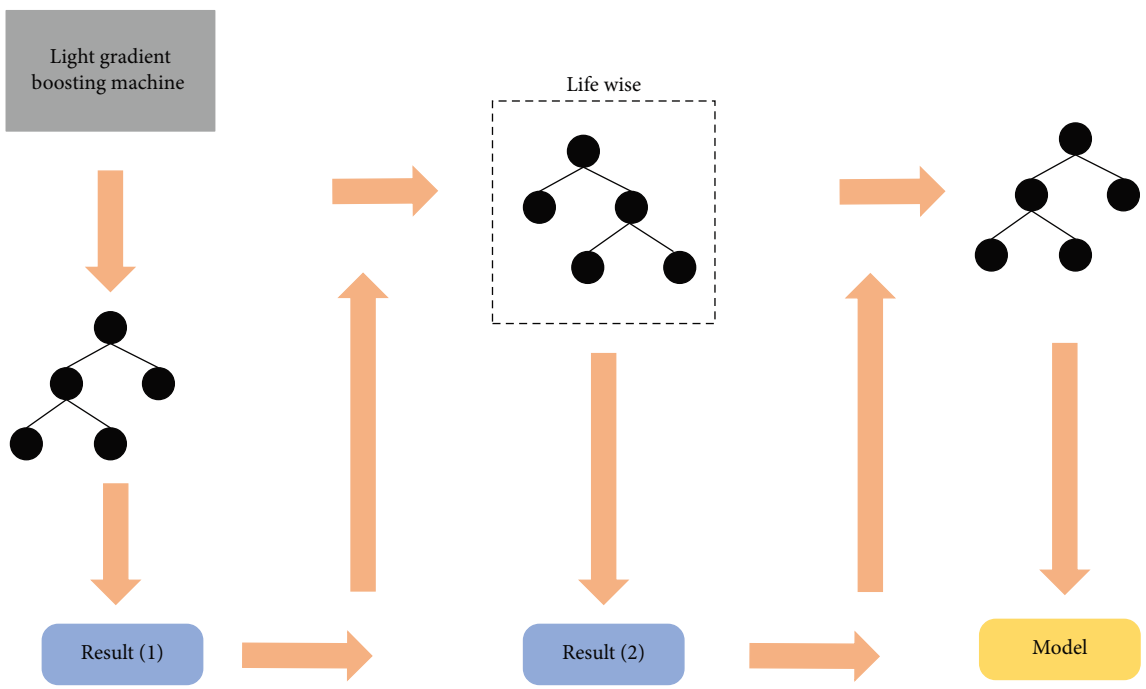

Figure 4: Light gradient boosting machine algorithm structure.

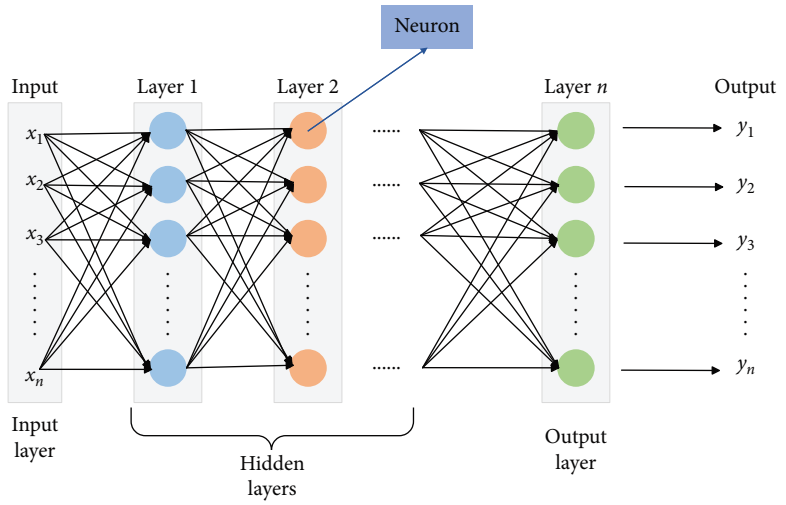

Figure 5: Fully connected neural network algorithm structure.

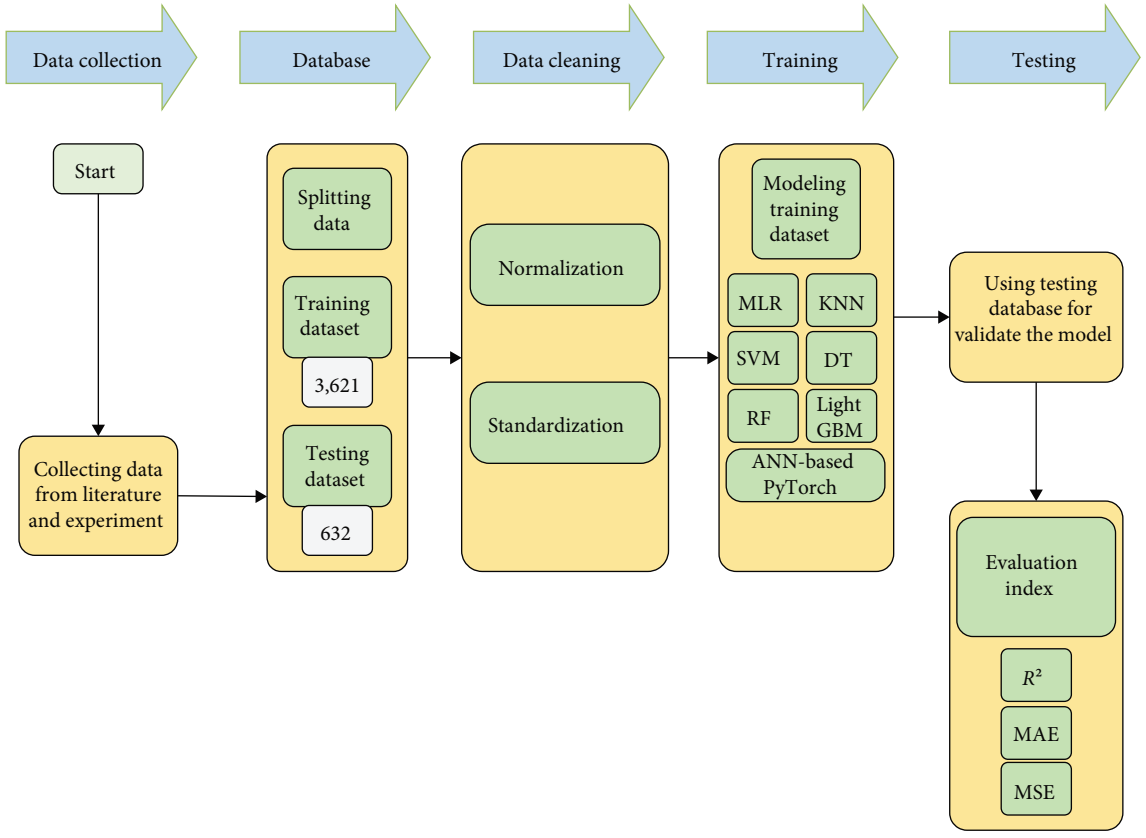

Figure 6: The flowchart of this study.

network where information can flow through all neurons. The flowchart for this process is shown in Figure 5.

4. Data Preprocessing and Feature Handling

The methodology of the current study includes a series of steps which has been drawn as a flowchart and is shown in Figure 6.

4.1. Data Collection and Acquisition. The primary source of data for this study is experimental data from the laboratory. A total of 4,253 data points were collected during the data acquisition process, with 3,621 of them used for training the prediction model after data cleaning. The collected data were consolidated into a .csv file, which includes seven attributes: six input variables (cement in kg/m³, fly ash in kg/m³, mineral powder in kg/m³, fine aggregate in kg/m³, coarse

Table 1: Statistical summary of data variables.

Data variable	Unit	Statistic measures				
		Average	Median	Minimum	Maximum	Standard deviation
Cement	kg/m^3	316.941	300.000	3.590	667.000	102.202
Water	kg/m^3	169.943	165.000	16.034	379.000	27.930
Fly ash	kg/m^3	54.708	58.330	0.000	367.330	58.498
Mineral powder	kg/m^3	43.401	0.000	0.000	375.000	59.543
Fine aggregate	kg/m^3	731.723	746.000	30.000	1,293.000	127.353
Coarse aggregate	kg/m^3	1,006.780	1,038.120	108.000	1,846.000	183.907
Water reducing admixture	kg/m^3	4.026	4.130	0.000	32.200	3.389
Compressive strengths	kg/m^3	45.346	44.600	5.660	95.300	13.526

aggregate in kg/m³, and water reducing admixture in kg/m³ and water in kg/m³) and one output variable (compressive strengths).

4.2. Data Cleaning. Data cleaning is an essential step after data collection given that most machine learning algorithms require data processing. The quality of the final product depends on the quality of the raw materials; thus, data cleaning aims to remove "dirty" data.

(1) Initially, duplicate values were identified. Here, duplicate values refer to rows where all data values are identical. A total of 243 rows of duplicate data were found and removed.

(2) When dealing with missing values, Pandas was used to read the data, and features with NaN values indicated missing data. Missing values can have a significant impact on model training, so they need to be handled. The two approaches to handling missing values are as follows: deleting the rows with missing values directly and filling the missing values with the median or mean. In this study, the first approach was selected, and any existing missing values were removed. After data cleaning, 3,621 data points remained.

4.3. Data Feature Handling

4.3.1. Standardization. Standardization processes the data column-wise based on the feature matrix. It transforms the features into a standard normal distribution, correlating them with the overall sample distribution. Each data point can influence the standardization. Different standardization methods will have different effects on the evaluation results of the system. In this study, Z-score standardization is adopted, which gives the mean and standard deviation of the original data for data standardization. The processed data conforms to the standard normal distribution, that is, the mean is 0 and the standard deviation is 1, and the normalized data maintains useful information in the outliers, making the algorithm less sensitive to outliers.

Standardization involves calculating the mean and standard deviation of the features using the following formula:

$$x' = \frac{x - \overline{X}}{S}, \tag{2}$$

where $\overline{X}$ represents the mean and S represents the standard deviation.

4.3.2. Normalization. Normalization transforms the feature values of samples to a unified scale, mapping the data to the (0, 1) or (a, b) interval. Normalization changes the original distances, distribution, and information of the data, whereas standardization generally does not.

In the process of data processing, the units of input data are different, and the range of some data may be particularly large, resulting in slow convergence and long training time. At the same time, inputs with a large data range may play a larger role in pattern classification, while inputs with a small data range may play a smaller role. Because the range of activation function of output layer of neural network is limited, it is necessary to map the target data of network training to the range of activation function. In this study, our output is limited to (0, 1), so the training data are normalized to the interval (0, 1).

The normalization formula is as follows:

$$x' = \frac{x}{\sqrt{\sum_{j}^{m} x[j]^2}}. \tag{3}$$

4.4. Data Visualization Processing. The 3,621 cleaned data points were subjected to visualization processing. Visualization allows us to observe the distribution of variables and the relationships between input and output variables.

4.4.1. Data Statistics. Python was used to obtain statistical measures (average, median, minimum, maximum, and standard deviation) for each column of the datasets. The results are summarized in Table 1.

The range of concrete mixes used in this study is as follows, with cement content (C) ranging from 3.59 to 667 kg/m^3, fly ash content (F) ranging from 0 to 367.33 kg/m^3, mineral powder

content (K) ranging from 0 to 375 kg/m^3, the fine aggregate content (FA) ranges between 30 and 1,293 kg/m^3, the coarse aggregate content (CA) ranges between 108 and 1,846 kg/m^3, the water reducing agent content (WRA) ranges between 0 and 32.2 kg/m^3, and the water content (W) ranges between 16.034 and 379 kg/m^3. The compressive strength of the collected datasets is ranged from 5.66 to 95.3 MPa.

4.4.2. Data Variable Distribution. In machine learning or deep learning regression problems, data are a crucial factor. Good data lead to better-fitting results. For deep learning, if the data are close to a normal distribution, then neural networks train faster and can avoid overfitting. However, in practical applications, most acquired data do not follow a normal distribution. Therefore, prior to training a model, the data should be evaluated, and normalization or standardization methods should be applied to obtain a data distribution closer to normal.

Skewness, a measure of the skewness and degree of deviation of statistical data distribution, was used in this study to determine the differences between data distribution and normal distribution. Skewness is the third standardized moment of a sample. The general formula is shown in Formula (4), as follows:

$$\text{Skew}\,(X) = E\left[\left(\frac{X-\mu}{\sigma}\right)^3\right] = \frac{k_3}{\sigma^3} = \frac{k_3}{k_2^{3/2}}. \tag{4}$$

The quantile–quantile plot (Q–Q plot) is a graphical tool used to compare the quantiles of data with the quantiles of a normal distribution. If the data follow a normal distribution, then all points fall on a straight line. In this study, cement, water reducing admixture, and water content were used as examples. The plots show that, except for the cement feature, the other features have significant deviations from a normal distribution. Therefore, data normalization was applied to obtain a data distribution closer to normal. The distribution before and after normalization is shown in Figure 7.

Research has shown that normalizing data can make them closer to a normal distribution. Skewness values noticeably decrease when data are normalized. However, normalization does not change the correlation coefficients between the feature values and the predicted compressive strength. Therefore, normalization is an essential step in data analysis and deep learning.

4.4.3. Examining the Correlation of Feature Variables. The correlation coefficient between two feature variables is used to estimate the relationship and strength of correlation between variables. The level of correlation between variables is determined by observing the magnitudes of different correlation coefficients in a heatmap. The formula for calculating the correlation coefficient is as follows:

$$\rho_{x1x2} = \frac{\text{Cov}\,(X_1, X_2)}{\sqrt{DX_1, DX_2}} = \frac{EX_1X_2 - EX_1 \times EX_2}{\sqrt{DX_1 \times DX_2}}, \tag{5}$$

where ρ represents the correlation coefficient, Cov represents covariance, and E represents the mathematical expectation/mean.

Some specific variables may depend on each other. Therefore, as shown in Figure 8, all possible correlation coefficients between variables have been derived. High positive or negative correlation coefficients between input variables may lead to poor efficiency and difficulty to determine the impact of these variables on the outcome. The figure shows that the highest correlation among independent input variables is among cement, fly ash and slag, reaching 0.46. Regarding the concrete strength CS, the highest correlations are with cement content and superplasticizer content, at 0.39 and 0.33, respectively.

5. Model Establishment and Prediction

For model establishment, the following steps are performed:

(1) The necessary libraries are imported.
(2) The data are preprocessed, including importing the datasets and processing the data, specifically handling missing values and normalizing the data.
(3) The appropriate machine learning model is selected, and the model is trained using the training set.
(4) The dataset to be predicted is input into the trained model to obtain the prediction results.

5.1. Data Partitioning. Prior to model establishment, the dataset is divided into training set and test set. The training set is used to train the model, whereas the test set is used to validate the model's fit. The data are split using the "train_test_split" function from the "sklearn" library in Python. Typically, the ratio between the training set and the test set is approximately 9 : 1, 8 : 2, 7 : 3, or 5 : 5. In this study, a split ratio of 8 : 2 is used, resulting in 3,621 (80%) training set data for model training and 632 (20%) test set data for data validation.

5.2. Model Training

5.2.1. Traditional Machine Learning Methods. In this study, six machine learning algorithm models are used to predict the compressive strength of concrete: MLR, KNN regression, SVM, DT, RF, and LightGBM. The performance of these models is evaluated using three evaluation metrics: coefficient of determination (R^2), mean absolute error (MAE), and mean squared error (MSE).

R^2, MAE, and MSE are all measures of a model's predictive performance, but each has a different emphasis. R^2 is an important indicator in regression analysis, which indicates how well the model fits the data. The value of R^2 is between 0 and 1, and the closer the value is to 1, the better the explanatory power of the model, that is, the better the independent variable explains the change of the dependent variable. R^2 is calculated by comparing the difference between the actual

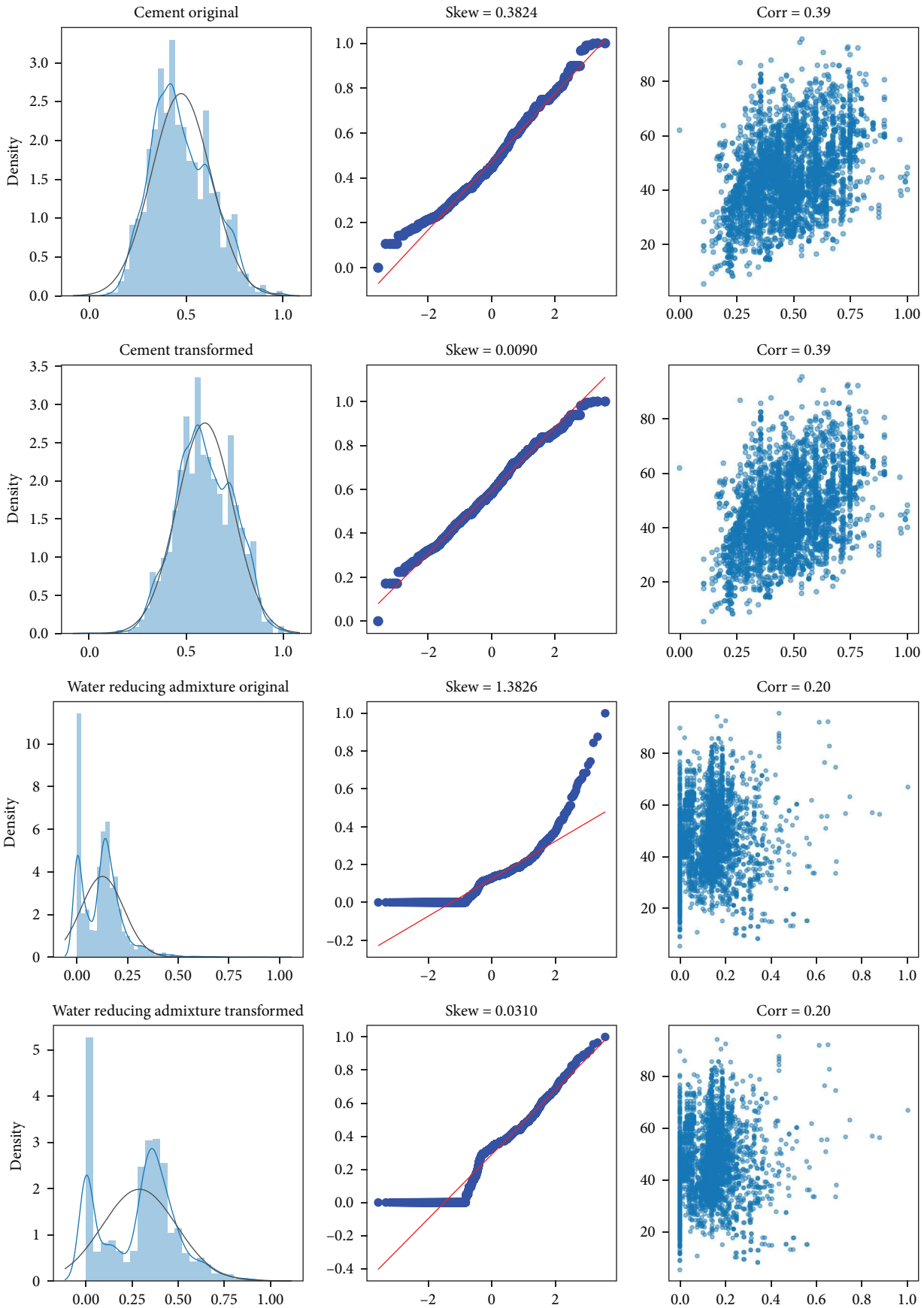

Figure 7: Continued.

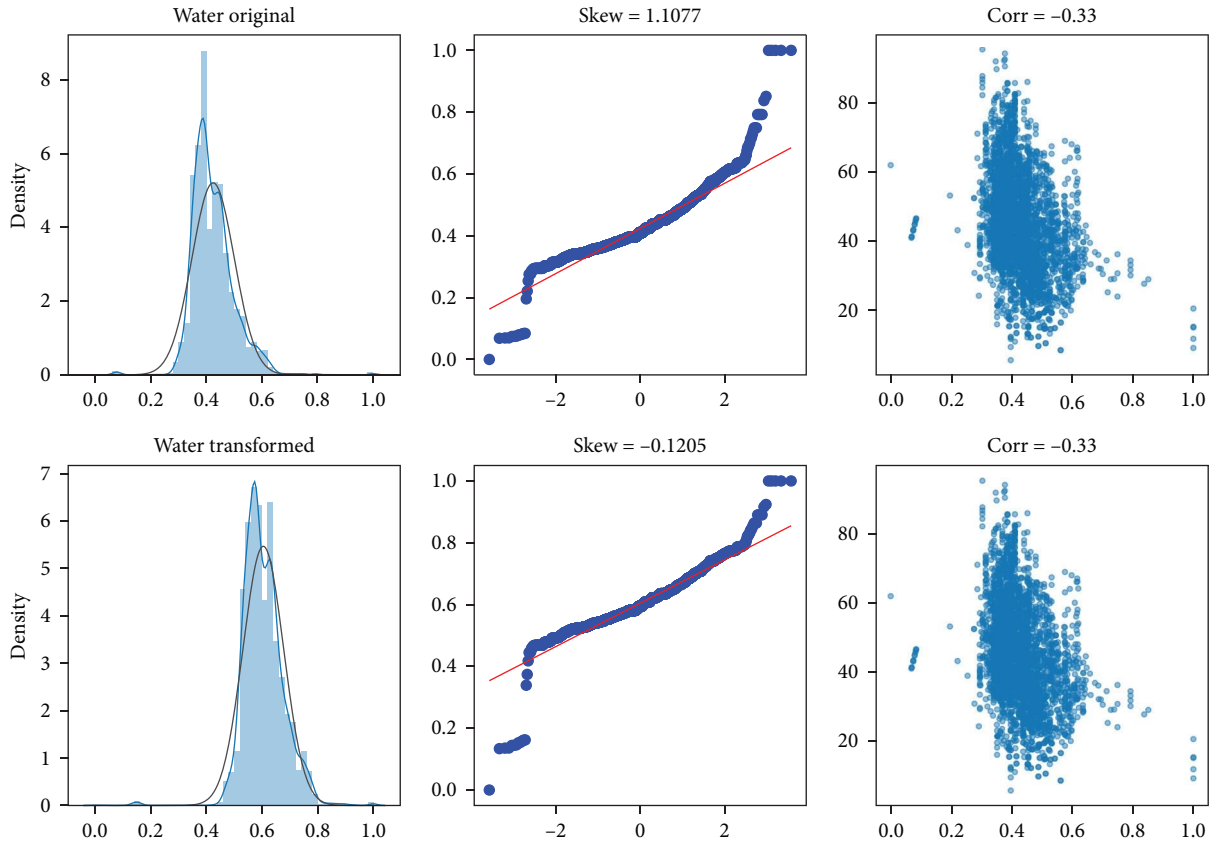

Figure 7: Comparison of distribution before and after data normalization.

Figure 8: Pearson correlations between variables.

observed value and the predicted value of the model, but it does not directly reflect the size of the forecast error; MAE is the average of the absolute value of the difference between the predicted value and the actual value. It can better reflect the true level of prediction error. The smaller the MAE value, the closer the prediction result of the model is to the actual data. Because MAE is more concise to compute, it is often used to actually evaluate the predictive performance of the model. MSE is a common measure of prediction error, which is the average of the sum of squared differences between the predicted and actual values. NSE is more sensitive to larger errors and can amplify the effect of larger errors.

In summary, R^2 is mainly used to evaluate the fit degree of the model, while MAE and MSE are used to measure the size of the prediction error. This study uses R^2 as the main evaluation metric given its ability to reflect the degree of explanation of independent variables for the dependent variable and capture certain data distribution patterns, with the two other metrics serving as auxiliary indicators.

(a) R^2.

The R^2 value reflects the extent to which the regression model explains the variation of the dependent variable or how well the model fits the observed values. The formula for calculating R^2 is as follows:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=0}^{n_{\text{samples}}-1}(y_i - \hat{y}_i)^2}{\sum_{i=0}^{n_{\text{samples}}-1}(y_i - \overline{y}_i)^2}. \quad (6)$$

(b) Mean absolute error.

Table 2: Prediction performance of machine learning algorithms.

Machine learning algorithm models	R^2 (%)	MAE (Mpa)	MSE (Mpa)
Multiple linear regression	0.480395	7.756060	104.636304
K-nearest neighbor regression	0.699114	4.869058	54.626699
Support vector machines	0.545818	6.884988	85.594238
Decision tree	0.617070	5.499421	67.524841
Random forest	0.699438	5.344389	52.527007
LightGBM	0.668214	5.844512	63.049502

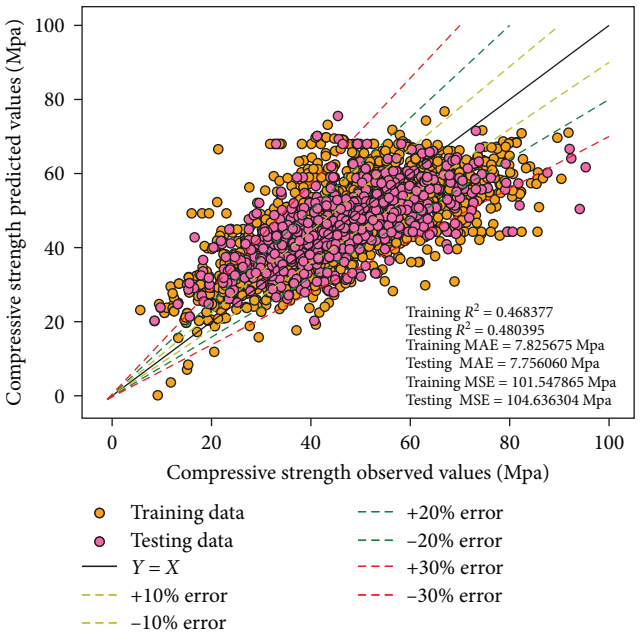

Figure 9: Prediction of multiple linear regression models.

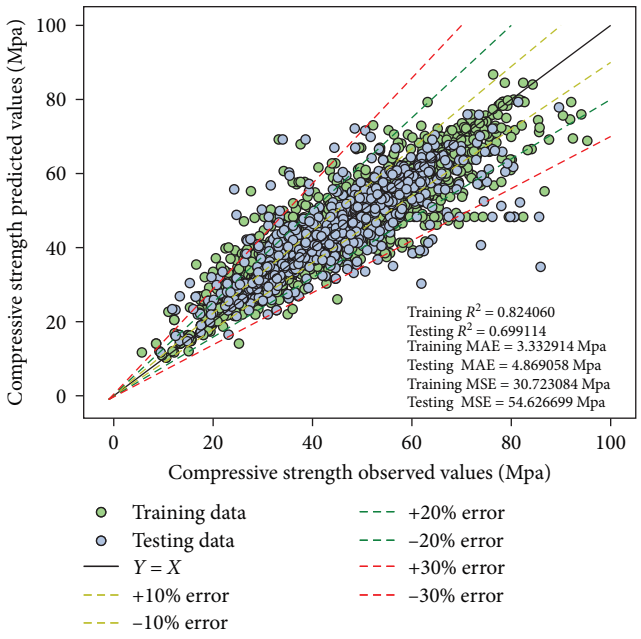

Figure 10: Prediction of K-nearest neighbor regression models.

MAE is the absolute difference between the predicted values and the true values. The formula for calculating MAE is as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i| = \frac{1}{n}\sum_{i=1}^{n}|e_i|. \tag{7}$$

(c) Mean squared error.

MSE is the expected value of the squared difference between the estimated parameters and the true parameters. MSE can conveniently measure the mean error and can be used to evaluate how much data have changed. The formula for MSE is as follows:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\mathrm{observed}_i - \mathrm{predicted}_i)^2. \tag{8}$$

From Table 2 and Figures 9, 10, 11, 12, 13, and 14, it is found that the nonlinear model is better than the linear model in predicting the concrete strength, and ensemble learning has the highest prediction accuracy in the same data, mostly above 0.60. Overall, RF and KNN regression models have the best-fitting effects, with R^2 values of 0.69.

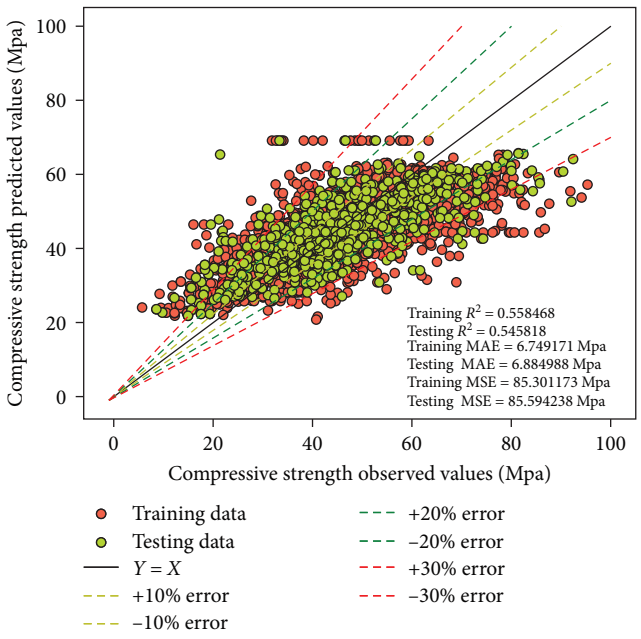

Figure 11: Prediction of support vector machines models.

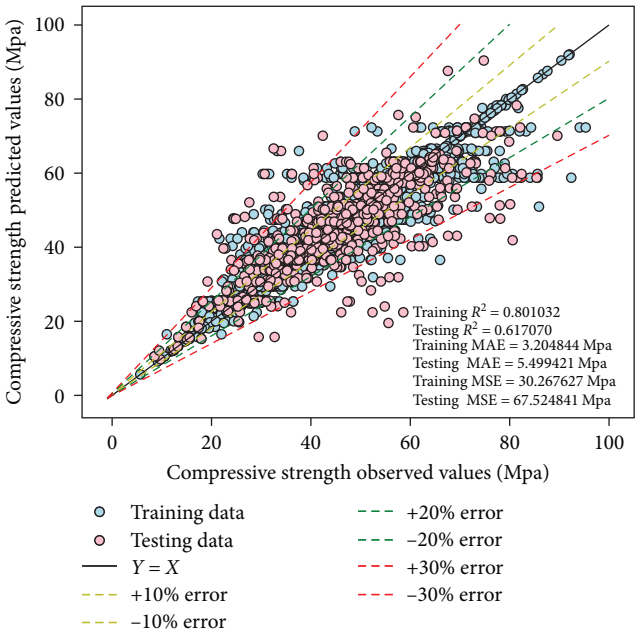

Figure 12: Prediction of decision tree models.

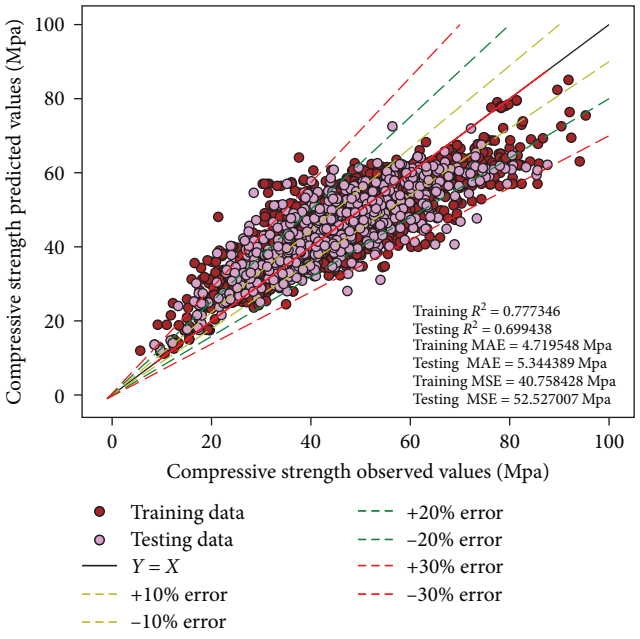

Figure 13: Prediction of random forest models.

However, the MSE of RF is lower than that of the KNN regression model. Therefore, considering the comprehensive evaluation, the RF model has the highest prediction accuracy in machine learning, but it is still approximately 0.69. Therefore, this study decides to optimize the RF model using optimization algorithms. The optimization algorithms selected are Harris Hawk optimization (HHO) and gray wolf optimizer (GWO). The results are summarized in Table 3.

From Table 3 and Figures 15, the comparison between the two algorithms indicates that the GWO has a significant optimization effect on the prediction performance in this

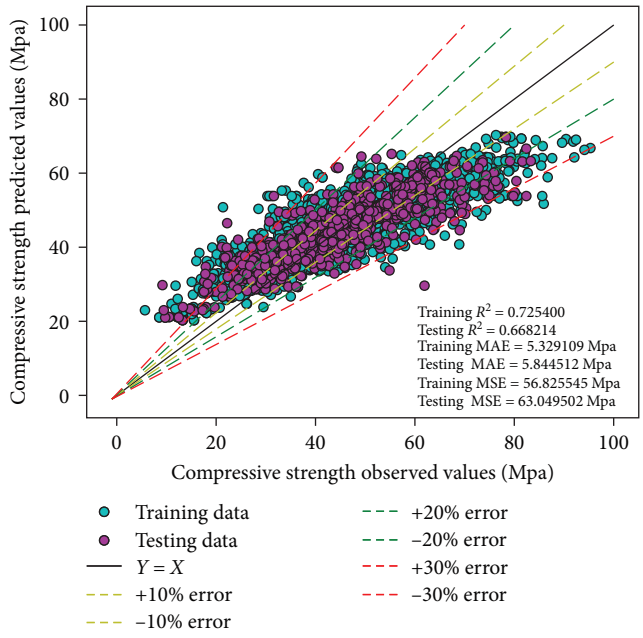

Figure 14: Prediction of LightGBM models.

Table 3: Algorithm performance after optimization.

Optimization algorithm	R^2 (%)	MAE (Mpa)	MSE (Mpa)
Harris Hawk optimization	0.716600	5.231100	53.606300
Gray wolf optimizer	0.737000	4.696600	45.272300

research dataset, improving the fit from 0.69 to 0.74. Despite the noticeable improvement, it does not satisfy the expected effect. Therefore, deep learning methods are introduced for prediction.

5.2.2. Fully Connected Neural Network-Based on the PyTorch Framework. The fully connected neural network used in this study has four layers: the first layer is the input layer with seven nodes, the second and third layers are hidden layers with 16 nodes each, and the fourth layer is the output layer with one node. The activation function used is the ReLU function. The structure diagram is shown in Figure 16. The results are summarized in Table 4.

Figures 17 and 18 find that the fully connected neural network based on PyTorch framework has higher accuracy and less error in predicting concrete strength compared to traditional machine learning. It can be seen from the experiment that the loss value of this model is the lowest and tends to be stable when the iteration is about 100 times, indicating that the stability of the model is better and the learned data features are more accurate. At the same time, the error between the predicted value and the real value is basically about 10% error. Therefore, this model is better than the traditional machine learning model in predicting the compressive strength of concrete and can be applied to engineering practice.

5.3. Best Performing Model. The results indicate that when machine learning algorithms are not used with optimization

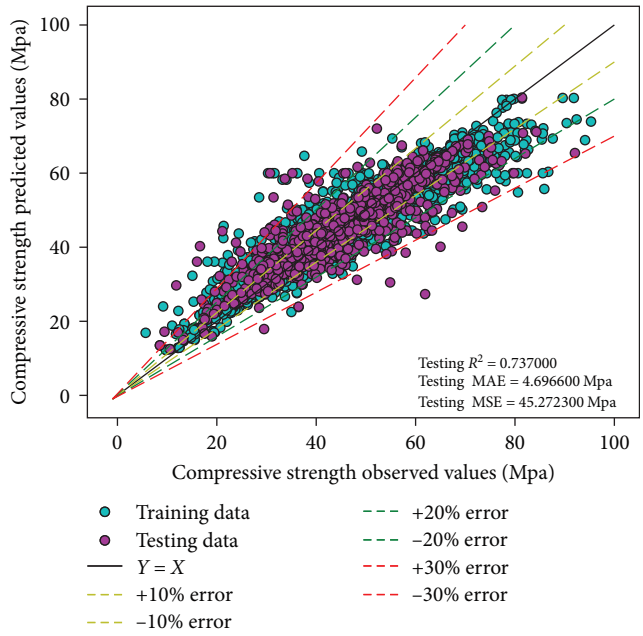

Figure 15: Prediction of gray wolf algorithm optimizes random forest models.

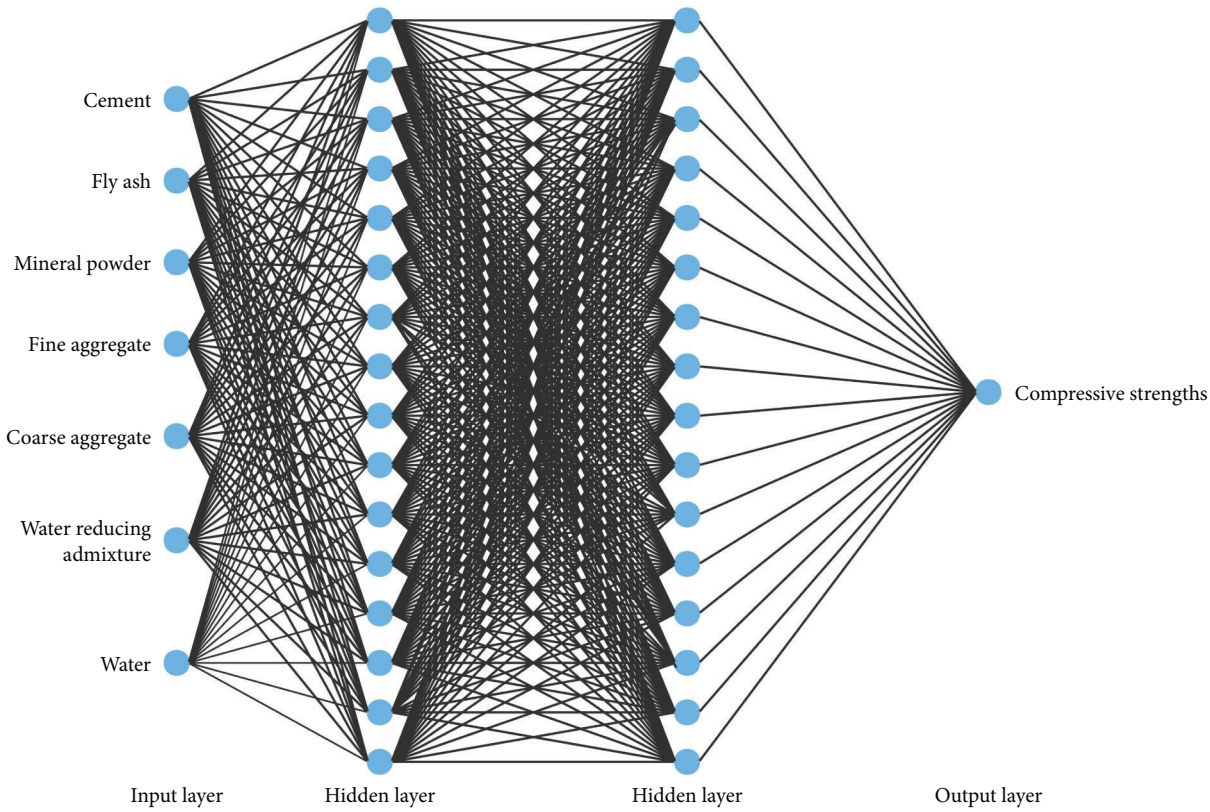

Figure 16: Fully connected neural network model structure.

Table 4: Fully connected neural network prediction performance for the PyTorch framework.

Model building	R^2 (%)	MAE (Mpa)	MSE (Mpa)
PyTorch-based neural networks	0.92000	3.984503	35.19259

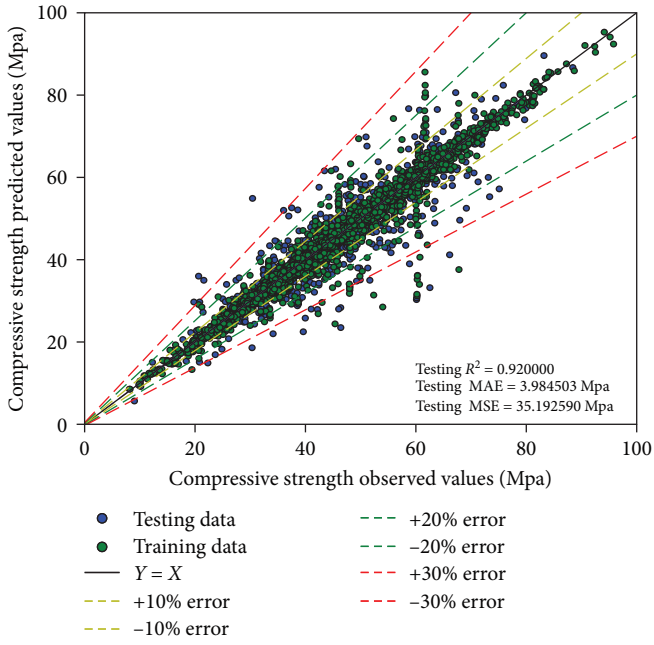

Figure 17: Prediction of fully connected neural network based on PyTorch framework.

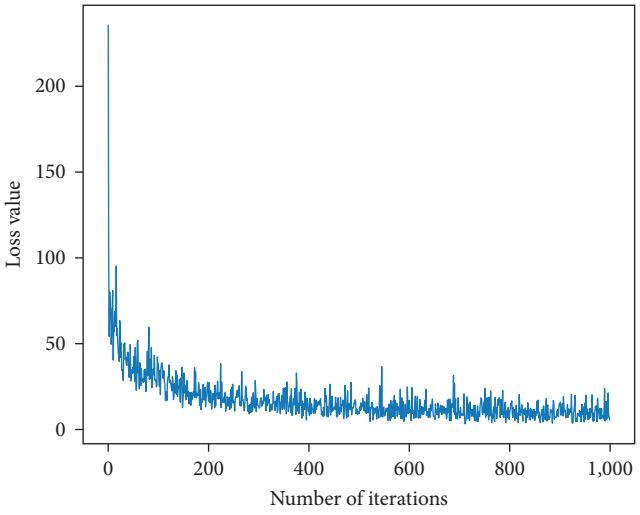

Figure 18: The number of iterations and loss value.

algorithms, RF performs the best in predicting the 28-day concrete strength. The R^2 and MAE values for RF are 0.69 and 5.45. When the GWO algorithm was applied to optimize the RF algorithm, and the R^2 and MAE values of the GWO–RF algorithm reached 0.74 and 4.69, achieving good prediction results.

Although general ANN models do not perform effectively in deep learning, the proposed fully connected neural network model based on the PyTorch framework outperforms all other models in predicting concrete strength, with R^2 and MAE values of 0.92 and 3.98. It demonstrates higher prediction accuracy than regular ANN and machine learning models, with loss values approaching stability after 100 iterations, indicating high prediction accuracy and strong generalization ability.

6. Conclusion

This study investigated the application of machine learning in predicting concrete compressive strength and introduced the concept of deep learning into traditional machine learning models. For the traditional machine learning, Ibrahim et al. [39] established four machine learning models to evaluate the effect of recycled coarse aggregate on the compressive strength of concrete and found that the full quadratic model could predict the compressive strength of recycled concrete aggregate much better. Mai et al. [40] and Li and Song [41] proposed a stacking model to predict the compressive strength of recycled brick aggregate and rice husk ash concrete by using ensemble machine learning method, respectively, and found that the model showed good performance for both different types of concrete. Li et al. [42] proposed a boosting-based ensemble machine learning algorithm, the gradient boosting regression tree (GBRT) algorithm, to predict the compressive strength of concrete. The results showed that GBRT had better prediction accuracy than other machine learning algorithms. Phoeuk and Kwon [43] employed four machine learning models based on ensemble learning algorithms to predict the compressive strength of RAC. The results showed that the proposed model had good accuracy and generalizability.

From the prior literatures, we find that machine learning algorithms usually require manual feature extraction of datasets, which is relatively complex and time-consuming. Deep learning algorithms are able to automatically learn feature representations from the raw data, reducing human intervention, and improving computational efficiency. At the same time, machine learning is generally difficult to deal with complex data structures and patterns, while deep learning algorithms could capture more complex patterns by learning hierarchical representations of data. So, the deep learning algorithms usually have stronger generalization ability. A new fully connected neural network model based on the PyTorch framework is established for predicting the 28-day concrete strength in this study. The following conclusions are drawn:

(1) Comparison of traditional machine learning algorithm models: Prior to using optimization algorithms, the KNN regression model and the RF model in ensemble learning achieve the highest accuracy in predicting concrete strength. However, compared with the KNN, the RF model is more stable and has smaller errors. Therefore, two intelligent optimization algorithms are applied to optimize RF, and the newly constructed GWO–RF model shows the best-fitting effect.

(2) Introduction of deep learning: General neural network models are not as effective as the optimized RF model, but the fully connected neural network model based on the proposed PyTorch framework achieve significantly better results in predicting

concrete strength, with an R^2 of 0.92. This finding further demonstrates that the model can effectively predict the compressive strength of recycled concrete, satisfying the engineering requirements.

(3) In this study, although the traditional machine learning could successfully predict the compressive strength of concrete, the traditional machine learning model has the disadvantages of poor generalization and manual data feature extraction. Therefore, this paper propose a fully connected neural network model based on PyTorch framework to predict the strength of concrete. The results show that the introduction of deep learning method not only strengthened the generalization ability of the model but also significantly improved the prediction accuracy, and achieved better results.

(4) Concrete compressive strength is a key parameter that determines the performance of concrete structure. Accurate prediction of concrete compressive strength is of great significance for engineering design and construction. However, due to the influence of regional environment and climate, the accuracy and reliability of concrete compressive strength prediction under different circumstances should be considered in future research, and more concrete sample data of different types, different ages, and different environmental conditions should be collected.

Data Availability

All data in this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 52368032 and 51808272 and the Natural Science Foundation of Gansu Province under Grant 23JRRA889.

References

[1] D.-C. Feng, Z.-T. Liu, X.-D. Wang et al., "Machine learning-based compressive strength prediction for concrete: an adaptive boosting approach," *Construction and Building Materials*, vol. 230, Article ID 117000, 2020.

[2] G. P. Jang, S. Huan, H. Hao, Y. F. Du, and C. J. Jiao, "Performance of steel reinforced high strength concrete investigated in the gas gun experiment," *Acta Physica Sinica*, vol. 62, no. 1, Article ID 016201, 2013.

[3] C.-H. Lim, Y.-S. Yoon, and J.-H. Kim, "Genetic algorithm in mix proportioning of high-performance concrete," *Cement and Concrete Research*, vol. 34, no. 3, pp. 409–420, 2004.

[4] X. Wang, M. Tian, Q. Qin, and J. Liang, "Hybridization of machine learning algorithms and an empirical regression model for predicting debris-flow-endangered areas," *Advances in Civil Engineering*, vol. 2023, Article ID 9465811, 16 pages, 2023.

[5] M. F. M. Zain and S. M. Abd, "Multiple regression model for compressive strength prediction of high performance concrete," *Journal of Applied Sciences*, vol. 9, no. 1, pp. 155–160, 2009.

[6] Z. Jia, J. Peng, Q. Lu et al., "A comprehensive method for the risk assessment of ground fissures: case study of the Eastern Weihe Basin," *Journal of Earth Science*, vol. 34, no. 6, pp. 1892–1907, 2023.

[7] Y. Long, W. Li, R. Huang, Q. Xu, B. Yu, and G. Liu, "A comparative study of supervised classification methods for investigating landslide evolution in the Mianyuan river basin, China," *Journal of Earth Science*, vol. 34, no. 2, pp. 316–329, 2023.

[8] J. W. Ma, D. Z. Lei, Z. Y. Ren, C. H. Tan, D. Xia, and H. X. Guo, "Automated machine learning-based landslide susceptibility mapping for the Three Gorges Reservoir Area, China," *Mathematical Geosciences*, 2023.

[9] Z. Liu, J. Ma, D. Xia et al., "Toward the reliable prediction of reservoir landslide displacement using earthworm optimization algorithm-optimized support vector regression (EOA-SVR)," *Natural Hazards*, vol. 120, no. 4, pp. 3165–3188, 2024.

[10] H. Salehi and R. Burgueño, "Emerging artificial intelligence methods in structural engineering," *Engineering Structures*, vol. 171, pp. 170–189, 2018.

[11] Q.-F. Li and Z.-M. Song, "High-performance concrete strength prediction based on ensemble learning," *Construction and Building Materials*, vol. 324, Article ID 126694, 2022.

[12] A. K. Al-Shamiri, T.-F. Yuan, and J. H. Kim, "Non-tuned machine learning approach for predicting the compressive strength of high-performance concrete," *Materials*, vol. 13, no. 5, Article ID 1023, 2020.

[13] Y. S. Cho, S. K. Baek, Y. T. Lee, S. H. Kim, and S. U. Hong, "Estimation of compressive strength of high-strength concrete with recycled aggregate using non-destructive test and numerical analysis," *Materials Research Innovations*, vol. 18, no. sup2, pp. S2-270–S2-277, 2014.

[14] A. Kumar, H. C. Arora, N. R. Kapoor et al., "Compressive strength prediction of lightweight concrete: machine learning models," *Sustainability*, vol. 14, no. 4, Article ID 2404, 2022.

[15] C. Xia, X. Guo, and W. Dai, "Numerical test and strength prediction of concrete failure process based on RVM algorithm," *Buildings*, vol. 12, no. 12, Article ID 2105, 2022.

[16] L. V. P. Meesaraganda, P. Saha, and N. Tarafder, "Artificial neural network for strength prediction of fibers' self-compacting concrete," in *Proceedings of Soft Computing for Problem Solving*, pp. 15–24, Springer, Singapore, 2017, December.

[17] H. U. Ahmed, A. A. Abdalla, A. S. Mohammed, and A. A. Mohammed, "Mathematical modeling techniques to predict the compressive strength of high-strength concrete incorporated metakaolin with multiple mix proportions," *Cleaner Materials*, vol. 5, Article ID 100132, 2022.

[18] H. U. Ahmed, A. S. Mohammed, and A. A. Mohammed, "Multivariable models including artificial neural network and M5P-tree to forecast the stress at the failure of alkali-activated concrete at ambient curing condition and various mixture proportions," *Neural Computing and Applications*, vol. 34, pp. 17853–17876, 2022.

[19] A. Mohammed, R. Kurda, D. J. Armaghani, and M. Hasanipanah, "Prediction of compressive strength of concrete modified with fly ash: applications of neuro-swarm and neuro-imperialism models," *Computers and Concrete*, vol. 27, no. 5, pp. 489–512, 2021.

[20] M. S. Barkhordari, D. J. Armaghani, A. S. Mohammed, and D. V. Ulrikh, "Data-driven compressive strength prediction of

fly ash concrete using ensemble learner algorithms," *Buildings*, vol. 12, Article ID 132, 2022.

[21] D. K. I. Jaf, P. I. Abdulrahman, A. S. Mohammed, R. Kurda, S. M. A. Qaidi, and P. G. Asteris, "Machine learning techniques and multi-scale models to evaluate the impact of silicon dioxide (SiO_2) and calcium oxide (CaO) in fly ash on the compressive strength of green concrete," *Construction and Building Materials*, vol. 400, Article ID 132604, 2023.

[22] W. Emad, A. S. Mohammed, R. Kurda et al., "Prediction of concrete materials compressive strength using surrogate models," *Structures*, vol. 46, pp. 1243–1267, 2022.

[23] A. Mohammed, L. Burhan, K. Ghafor, W. Sarwar, and W. Mahmood, "Artificial neural network (ANN), M5P-tree, and regression analyses to predict the early age compression strength of concrete modified with DBC-21 and VK-98 polymers," *Neural Computing and Applications*, vol. 33, no. 13, pp. 7851–7873, 2021.

[24] A. S. Mohammed, W. Emad, W. S. Qadir, R. Kurda, K. Ghafor, and R. K. Faris, "Modeling the impact of liquid polymers on concrete stability in terms of a slump and compressive strength," *Applied Sciences*, vol. 13, no. 2, Article ID 1208, 2023.

[25] N. S. Piro, A. Mohammed, S. M. Hamad, and R. Kurda, "Artificial neural networks (ANN), MARS, and adaptive network-based fuzzy inference system (ANFIS) to predict the stress at the failure of concrete with waste steel slag coarse aggregate replacement," *Neural Computing and Applications*, vol. 35, no. 18, pp. 13293–13319, 2023.

[26] A. Sadrmomtazi, J. Sobhani, and M. A. Mirgozar, "Modeling compressive strength of EPS lightweight concrete using regression, neural network and ANFIS," *Construction and Building Materials*, vol. 42, pp. 205–216, 2013.

[27] F. Khademi, M. Akbari, S. M. Jamal, and M. Nikoo, "Multiple linear regression, artificial neural network, and fuzzy logic prediction of 28 days compressive strength of concrete," *Frontiers of Structural and Civil Engineering*, vol. 11, no. 1, pp. 90–99, 2017.

[28] A. N. Beskopylny, S. A. Stel'makh, E. M. Shcherban' et al., "Concrete strength prediction using machine learning methods CatBoost, k-Nearest neighbors, support vector regression," *Applied Sciences*, vol. 12, no. 21, Article ID 10864, 2022.

[29] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999.

[30] X. Jiang and S. Li, "BAS: beetle antennae search algorithm for optimization problems," *International Journal of Robotics and Control*, vol. 1, no. 1, pp. 1–5, 2018.

[31] X. H. Li, "Using random forest for classification and regression," *Chinese Journal of Applied Entomology*, vol. 50, no. 4, pp. 1190–1197, 2013.

[32] H. Li, J. Lin, X. Lei, and T. Wei, "Compressive strength prediction of basalt fiber reinforced concrete via random forest algorithm," *Materials Today Communications*, vol. 30, Article ID 103117, 2022.

[33] D. Suenaga, Y. Takase, T. Abe, G. Orita, and S. Ando, "Prediction accuracy of random forest, XGBoost, LightGBM, and artificial neural network for shear resistance of post-installed anchors," *Structures*, vol. 50, pp. 1252–1263, 2023.

[34] Y. Q. Jia, E. Shelhamer, J. Donahue et al., "Caffe: convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM International Conference on Multimedia*, pp. 675–678, Association for Computing Machinery, Berkeley, CA, USA, 2014, November.

[35] F. Seide and A. Agarwal, "CNTK: microsoft's open-source deep-learning toolkit," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 2135, Association for Computing Machinery, San Francisco, CA, USA, 2016, August.

[36] M. Abadi, P. Barham, J. M. Chen et al., "TensorFlow: a system for large-scale machine learning," in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, pp. 265–283, USENIX Association, Savannah, GA, USA, 2016, November.

[37] The Theano Development Team, "Theano: a python framework for fast computation of mathematical expressions," *arXiv e-prints abs/1605.02688*, 2016.

[38] A. Paszke, S. Gross, F. Massa et al., "PyTorch: an imperative style, high-performance deep learning library," in *Proceeding of the 33rd International Conference on Neural Information Processing Systems*, Curran Associates Inc., Vancouver, Canada, 2019, December.

[39] A. K. Ibrahim, H. Y. Dhahir, A. S. Mohammed, H. A. Omar, and A. H. Sedo, "The effectiveness of surrogate models in predicting the long-term behavior of varying compressive strength ranges of recycled concrete aggregate for a variety of shapes and sizes of specimens," *Archives of Civil and Mechanical Engineering*, vol. 23, no. 1, 2023.

[40] H.-V. T. Mai, M. H. Nguyen, S. H. Trinh, and H.-B. Ly, "Toward improved prediction of recycled brick aggregate concrete compressive strength by designing ensemble machine learning models," *Construction and Building Materials*, vol. 369, Article ID 130613, 2023.

[41] Q. Li and Z. Song, "Prediction of compressive strength of rice husk ash concrete based on stacking ensemble learning model," *Journal of Cleaner Production*, vol. 382, Article ID 135279, 2023.

[42] D. H. Li, Z. L. Tang, Q. Kang, X. Y. Zhang, and Y. H. Li, "Machine learning-based method for predicting compressive strength of concrete," *Processes*, vol. 11, no. 2, Article ID 390, 2023.

[43] M. Phoeuk and M. Kwon, "Accuracy prediction of compressive strength of concrete incorporating recycled aggregate using ensemble learning algorithms: multinational dataset," *Advances in Civil Engineering*, vol. 2023, Article ID 5076429, 23 pages, 2023.