






# Comparative Analysis of Machine Learning Techniques for Prediction of the Compressive Strength of Field Concrete

Ajibola Oyedeji<sup>1</sup>, Adekunle David<sup>2</sup>, Ositola Osifeko<sup>1</sup>, Abisola Olayiwola<sup>1</sup>, Omobolaji Opafola<sup>2</sup>

<sup>1</sup>Department of Computer Engineering, Olabisi Onabanjo University, Ago-Iwoye, Nigeria

<sup>2</sup>Department of Civil Engineering, Olabisi Onabanjo University, Ago-Iwoye, Nigeria

Corresponding author:

Ajibola Oyedeji, Department of Computer Engineering, Olabisi Onabanjo University, Ago-Iwoye, Nigeria  
oyedeji.ajibola@oouagoiwoye.edu.ng

Article History:

Received: 18.01.2024

Accepted: 10.07.2024

Published Online: 23.08.2024

## ABSTRACT

The determination of the concrete compressive strength remains a challenging task in the concrete industry. Machine learning (ML) algorithms offer an alternative and this study presents a comparative analysis of five ML regression models; Gradient Boosting (GB), Random Forest (RF), Decision Tree (DT), K-Nearest Neighbors (KNN), and Linear Regression (LR) on a dataset of 1030 concrete samples. The findings indicate that the GB model achieved the best performance. The developed GB model achieved R-squared values of 91.60%, 91.43%, and 90.18% for the 10-fold, 5-fold, and 3-fold cross-validations, respectively, with mean absolute error, root mean squared error, and mean absolute percentage error values of 2.6776, 4.3523, and 9.19%, respectively. The GB model trained and evaluated was deployed to a web application using Streamlit for real-time prediction of the concrete compressive strength. The results of this research offer a precise and practical method for judging the quality of concrete constructions.

**Keywords:** Machine learning, Concrete compressive strength, Prediction, Regression models, Web application

## 1. Introduction

The compressive strength of concrete (CCS) is an important factor for construction experts' consideration. Concrete is typically made up of cement, fine aggregates, coarse aggregates, and water, with or without the addition of chemical admixtures. CCS at 28 days is essential for designing reinforced concrete constructions [1]. Traditionally, concrete mixture portions that satisfy the minimum 28-day curing period requisite have been determined using empirical prescriptive and performance-based mixture design techniques. However, recent literature has started to explore numerical methods for predicting the CCS after 28 days. Accurate prediction of the 28-day CCS is crucial because it (a) ensures concrete quality, (b) reduces the amount of testing that needs to be conducted on different concrete batches to achieve desired and required strength goals, and (c) enhances safety.

Recent computer studies have established the potential of modern statistical modeling methods or approaches to numerically forecast the CCS for laboratory-mixed concrete, referred to as laboratory concrete [2]–[5]. Therefore, these approaches can provide a means of accurately estimating the 28-day CCS, leading to better concrete quality and improved safety. However, despite these efforts, various uncertainties encountered during the mixing, transport, placement, curing, and finishing of concrete can make it difficult to achieve accurate predictions. These uncertainties are particularly challenging in the case of field concrete, which refers to concrete produced and placed at the construction site. In particular, the environmental conditions that can fluctuate during the production and placement of field concrete, as well as the numerous factors that can affect the quality and performance of the final product, make it difficult to achieve consistent and reliable predictions of the 28-day CCS.

Concrete's 28-day compressive strength estimation is a complex issue. CCS is influenced by the intricate physical and chemical interactions that take place between the components of concrete. Research has shown a nonlinear relationship between the CCS and its air content, which is often increased to improve workability and freeze-thaw resistance [6], [7]. As the water-to-cement ratio is increased, the CCS tends to decrease. However, other factors can affect the CCS that are not as easily identified. For example, the ratio of coarse to fine aggregate, the interfacial bindings between the coarse aggregate and mortar influence compressive strength [8]–[10]. Likewise, field concrete exhibits far greater variability in compressive

strength due to the highly varied job site conditions where it is mixed and applied. Here, the final CCS can be impacted by temperature, humidity, and bad weather [11].

While traditional methods of CCS estimation rely on time-consuming and expensive laboratory tests, machine learning algorithms offer an option for predicting concrete strength from non-destructive testing data. There is an increasing research focus on the application of machine learning (ML) models for concrete mixtures [12], [13]. These models utilize the different mix proportions to predict the CCS, the target variable [14]. The CCS was predicted by making use of ML methods such as decision tree (DT), gradient boosting (GB), and bagging regressor (BR) [15]. The BR technique was the best performing methods achieving the highest R-squared value of 0.92. The mean absolute error (MAE) and root mean square error (RMSE) values for the BR model were 4.26 and 5.69 respectively, which was the lowest compared to the other models trained; MAE = 4.96 and RMSE = 7.05 for GB and MAE = 6.39 and RMSE = 8.95 for DT. A prediction of CCS with electrical resistivity included as an input variable saw the performance of the ML models increase as against the traditional proportion mix as input variables [16]. The models performances were increased as the R-squared values of the decision trees and Gaussian models increased from 0.77 to 0.79 and 0.81 to 0.82 respectively.

Using 1030 data samples from concrete compressive strength tests obtained from the University of California, Irvine, a novel Hybrid Ensemble Model (HENSM) was the best predictor of the CCS in the study [17]. Furthermore, the adapted AdaBoost achieved an accuracy of 98% on the same dataset, outperforming the artificial neural network (ANN) and support vector machine (SVM) [18]. Likewise, ANN has succeeded in surpassing the performance of the LR, SVM and test ensembles of the models by achieving an R-value of 0.909 [19]. A proposed deep convolutional neural network (DCNN) model is trained using a data set consisting of 380 groups of concrete mixes. The results show that the DCNN achieves high coefficients of determination of 0.967 which is better than the performance of the SVM, ANN, and AdaBoost [20].

This paper aims to predict the concrete compressive strength using 5 ML methods and evaluate the performance of the models. The input variables used for predicting the Concrete Compressive Strength (CCS) are cement (C), Blast Furnace Slag (BFS), Fly Ash (FA), Water (W), Superplasticizer (SP), Coarse Aggregate (C\_Ag), Fine Aggregate (F\_Ag), all measured in  $\text{kg/m}^3$ , and Age (in days). The machine learning models for the regression problem used in our study are Linear Regression (LR), Random Forest (RF), Decision Tree (DT), K-Nearest Neighbors (KNN), and Gradient Boosting Regression (GB). The models have been selected among the many available models to study and compare different methods or approaches including the linear statistical approach, tree method, ensemble methods and distance-based models. The result of this study is of great importance to the cost-effective method of determining the strength of concrete structures capable of producing quality concrete structures, improving safety, and reducing maintenance costs over the lifetime of these structures. Furthermore, a web application has been deployed for easy interaction and prediction of the CCS based on the various inputs for real-time use.

The paper is presented as follows; the materials and methodology are presented in Section 2, section 3 contains the results and discussion and the conclusion is in Section 4.

## 2. Materials and Method

For this research, the following steps were followed; data collection, data preprocessing and analysis, model development, testing and evaluation, and model deployment. Figure 1 shows the process flow diagram for the listed steps.



Figure 1. Concrete Compressive Strength Prediction Process Flow Diagram

### 2.1. Data Collection and Analysis

The Concrete Compressive Strength dataset used in this research project has been retrieved from the UCI Machine Learning Repository available at <https://archive.ics.uci.edu/dataset/165/concrete+compressive+strength> [21], [22]. The data used was curated by experiments from 17 different sources. Data was assembled for concrete containing Portland cement plus fly ash, blast furnace, and superplasticizer cured under normal conditions. The details of the experimental setup are presented in [21]. The dataset is acceptable as collecting and testing the data met accepted guidelines in the concrete industry.

The data set is of the form  $\mathcal{D} = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  where  $x_i \in X$  is the  $i^{th}$  input and  $y_i \in Y$  the corresponding target.  $X = \mathbb{R}^d$ , in which case  $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$  is a  $d$ -dimensional vector called an instance. The data set includes 1030 instances and 9 quantitative attributes including 8-input variables and 1-output or target variable. After analyzing the data collected, 25 instances were duplicated and the duplicate records were dropped remaining total of 1005 instances. The dataset also does not contain any missing values. The characteristics and summary statistics of the data collected are shown in Table 1.

Table 1. Summary Statistics of Attributes

Attribute	Abbreviation	Mean	Standard Deviation	Min	25th Percentile	50th Percentile	75th Percentile	Max
Cement ( $kg/m^3$ )	C	278.63	104.34	102	190.7	265	349	540
Blast Furnace Slag ( $kg/m^3$ )	BFS	72.04	86.17	0	0	20	142.5	359.4
Fly Ash ( $kg/m^3$ )	FA	55.54	64.21	0	0	0	118.3	200.1
Water ( $kg/m^3$ )	W	182.08	21.34	121.8	166.6	185.7	192.9	247
Superplasticizer ( $kg/m^3$ )	SP	6.03	5.92	0	0	6.1	10	32.2
Coarse Aggregate ( $kg/m^3$ )	C_Ag	974.38	77.58	801	932	968	1031	1145
Fine Aggregate ( $kg/m^3$ )	F_Ag	772.69	80.34	594	724.3	780	822.2	992.6
Age	Age	45.86	63.73	1	7	28	56	365
Concrete Compressive Strength	CCS	35.25	16.28	2.33	23.52	33.8	44.87	82.6

There is no strong correlation between the attributes. The highest correlation exists between the cement component and the concrete compressive strength attributes with a value of 0.49 followed by the Age and Superplasticizer components with a Pearson’s correlation value of 0.34 respectively as shown in Figure 2. This tells us that no attribute is a strong estimator of the CCS which is the output variable.



Figure 2. Pearson’s Correlation Plot of the Input and Target Variables

### 2.2. Model Development

The data set was randomly split into the train and test data in a ratio of 80:20 using a random state of 42 in Scikit-learn. The training dataset had 804 instances while the test dataset had 201 instances. The GridSearch cross-validation (CV) function and the 3-Fold, 5-Fold, and 10-Fold cross-validation methods were deployed to get the optimum model with the best performance and least error. There are several machine learning models used for supervised learning regression problems. However, the five machine learning regression models used in our analysis are Linear Regression (LR), Random Forest (RF), Decision Tree (DT), K-Nearest Neighbors (KNN), and Gradient Boosting Regression (GB).

### 2.2.1. Linear Regression

The multivariate LR model is a statistical approach for the prediction of the value of a dependent or target variable using several independent or input variables [23]. This is achieved by fitting the line of best fit between the independent or input variable(s)  $x$  and the dependent or output variable  $y$ . For this work, the independent variables are C, BFS, FA, W, SP, C\_Ag, F\_Ag, and Age while the dependent variable is the CCS. This can be represented as

$$f: \mathbb{R}^D \rightarrow \mathbb{R} \tag{1}$$

where the input vector  $x$  is D-dimensional, and the function  $f$  then applied to it returns the CCS value. In essence, we aim for a function  $f(x)$ , such that,

$$f(x) = CCS = \theta^T x + \phi_0 \tag{2}$$

for unknown  $\theta, \phi$ . Upon training the concrete compressive dataset, the linear regression function fitted is defined in Equation 3.

$$CCS = -23.3893 + (0.1167 \times C) + (0.0981 \times BFS) + (0.0846 \times FA) - (0.1314 \times W) + (0.3315 \times SP) + (0.0155 \times C\_Ag) + (0.0206 \times F\_Ag) + (0.1106 \times Age) \tag{3}$$

### 2.2.2. Decision Tree Regression

A DT is a tree data structure made up of several nodes and branches. The decision tree algorithm utilizes a divide-and-conquer method and repeatedly partitions the input variables to classify or predict the output parameter [24].

Mathematically, a DT can be represented as a function that maps an input feature vector  $x$  to a class label  $y$  described in Equations 4 and 5 as:

$$y = f(x) \tag{4}$$

The decision tree function  $f(x)$  can be defined recursively as follows:

$$f(x) = \begin{cases} C_1 & \text{if } x_i < t \\ C_2 & \text{if } x_i > t \end{cases} \tag{5}$$

where  $x_i$  is the value of the  $i$ th feature in  $x$ ,  $t$  is the threshold value, and  $C_1$  and  $C_2$  are the class labels for the resulting subsets. The threshold value  $t$  and the class labels  $C_1$  and  $C_2$  are determined during the training process by optimizing a criterion such as information gain or Gini impurity. The decision tree regressor was fitted to the target variable and at each partition instance, the error was calculated between the predicted value and the known target value [25].

The decision tree's hyperparameter is tuned using cross-validation. One such hyperparameter is the depth of the tree (the number of splits a tree can make before coming to a prediction). This was set to 16 for the decision tree regression model trained on the data as presented in Table 2. Figure 3 shows the first 2 splits of the decision tree while Figure 4 shows the importance of the variables used in making prediction. In Figure 4, Cement and Age had the highest significance values at 35.7% and 32.2% respectively, and were strong estimators for predicting the value of the CCS for the DT.

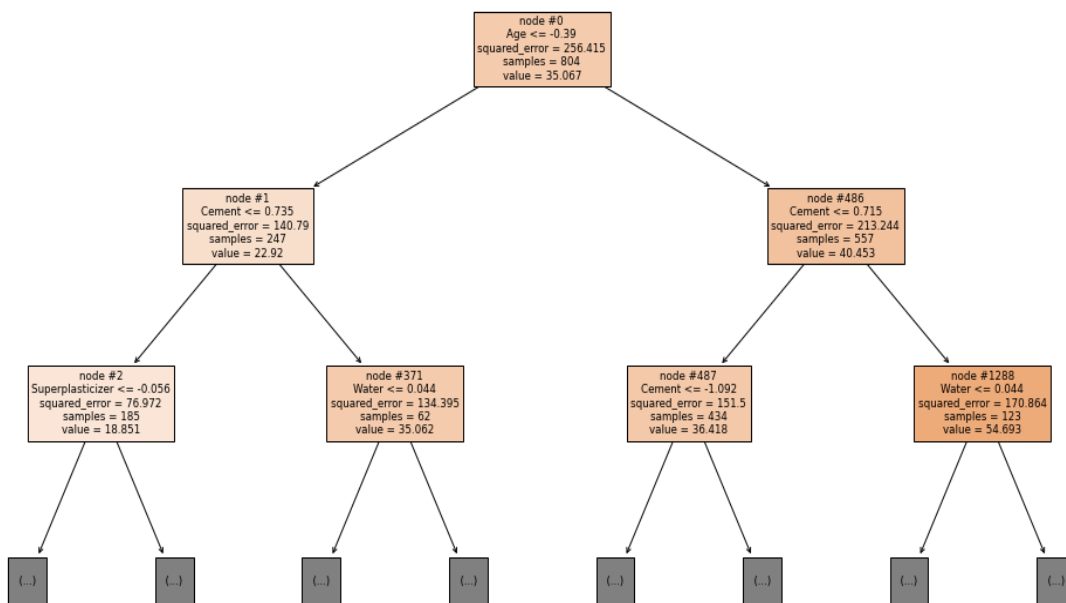


Figure 3. Decision Tree Regression Model Decision Making showing the First Two Splits

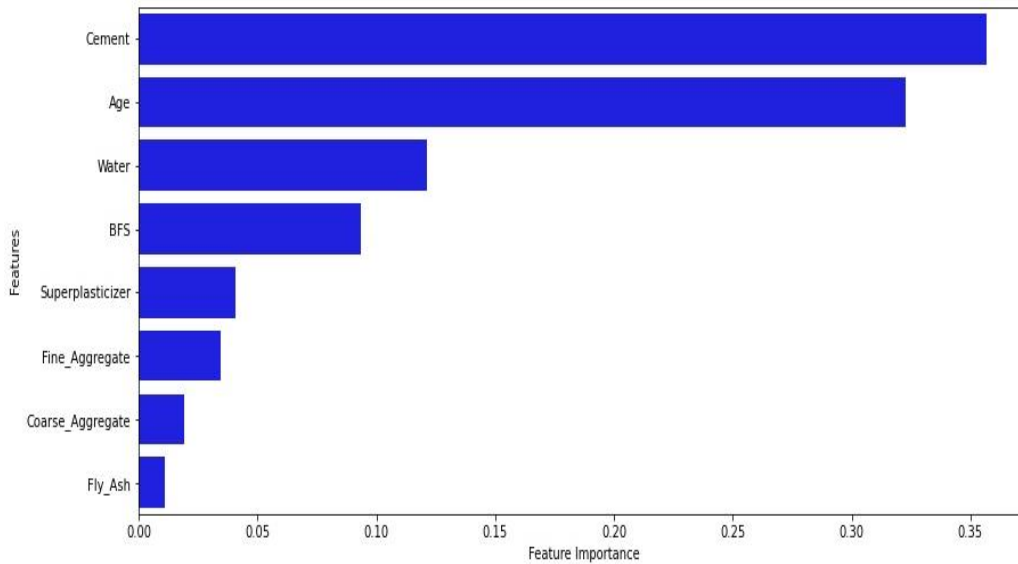


Figure 4. Variable Importance of the DT Model

**2.2.3. Random Forest Regression**

An RF regression is an ensemble regression technique containing numerous decision trees to predict or forecast a target variable [26]. The RF uses the Bagging (that is, bootstrapping and aggregating) method where homogenous weak learners' models (in this case, decision trees) independently learn from one another in parallel. The final prediction is achieved by a model averaging approach [25], [27]. The number of estimators (decision trees) fitted for the RF regression model was 400 while the maximum depth of each DT was 20 as shown in Table 2.

In Figure 5, Age and Cement had the highest feature importance of 35.3% and 31.1% respectively, and were the strongest estimators for predicting the value of the CCS in the RF regression model.

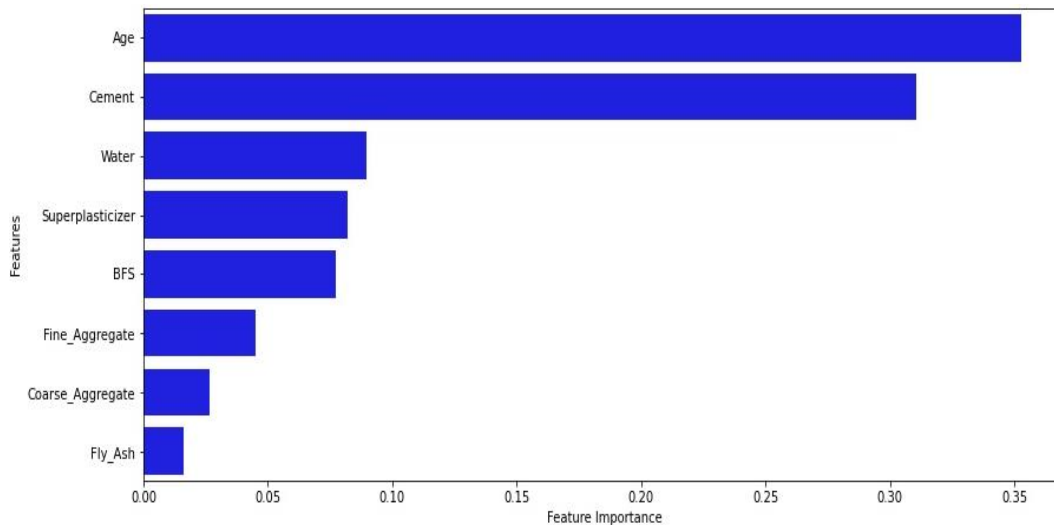


Figure 5. Variable Importance of the RF Model

**2.2.4. Gradient Boosting Regression**

The gradient boosting (GB) regression tree model is a stacked learning approach where a robust predictive model is formed by combining several individual weak learning regression trees (DT) [28].

For the training set described earlier with n data points  $(x_i, y_i)$  where  $x_i$  is the input features and  $y_i$  is the corresponding output or target variable. The goal of gradient boost regression is to find an ensemble of weak regression models  $h_1(x), h_2(x), \dots, h_m(x)$  that can approximate the true underlying function  $f(x)$  that maps the inputs to the target variable.

Each weak model  $h_m(x)$  takes the input features  $x$  and returns a scalar prediction, which is then combined with the predictions of the other weak models to obtain the final prediction. The gradient boost regression algorithm iteratively adds new weak models to the ensemble, with each new model attempting to correct the errors of the previous models.

At iteration m, the current prediction of the ensemble is given in Equation 6:

$$f_m(x) = f_{m-1}(x) + \gamma_m h_m(x) \tag{6}$$

where  $f_{m-1}(x)$  is the previous prediction of the ensemble,  $\gamma_m$  is the learning rate or step size that controls the contribution of the new weak model  $h_m(x)$ , and  $h_m(x)$  is the new weak model that is trained to fit the negative gradient of the loss function with respect to the current prediction  $f_{m-1}(x)$ . The negative gradient is defined in Equation 7 as

$$\gamma_{mi} = - \left[ \frac{\partial L(y_i, f_{m-1}(x_i))}{\partial f_{m-1}(x_i)} \right] \tag{7}$$

where  $L(y, f(x))$  is the loss function that measures the difference between the true target value  $y$  and the predicted value  $f(x)$ .

The weak model  $h_m(x)$  is typically chosen to be a simple regression model such as a decision tree and is trained to minimize the loss function concerning the negative gradients  $\gamma_{mi}$ .

Equation 8 below is the final prediction of the gradient boost regression ensemble after  $m$  iterations:

$$f(x) = f_0(x) + \gamma_1 h_1(x) + \gamma_2 h_2(x) + \dots + \gamma_m h_m(x) \tag{8}$$

where  $f_0(x)$  is the initial prediction of the ensemble and  $\gamma_1, \gamma_2, \dots, \gamma_m$  are the learning rates or step sizes that control the contribution of each weak model in the ensemble [29].

The maximum depth for the Gradient Boosting Regression model was set to 4 and the number of estimators was 200 with a learning rate of 0.2 as shown in Table 2. Fig. 6 shows the feature importance of the gradient-boosting regression model where Age and Cement had the highest importance of 37.8% and 28.8% respectively. It is noteworthy that while Cement and Age features were the strongest estimators, F\_Ag, C\_Ag, and FA had the least feature importance for the DT, RF and GB models.

Table 2. Models' Hyperparameter Tuning

Model	Hyperparameters	Value
LR	None	-
DT	max_depth	16
RF	max_depth	20
	n_estimators	400
GB	max_depth	4
	n_estimators	200
	learning_rate	0.2
KNN	n_neighbors	7

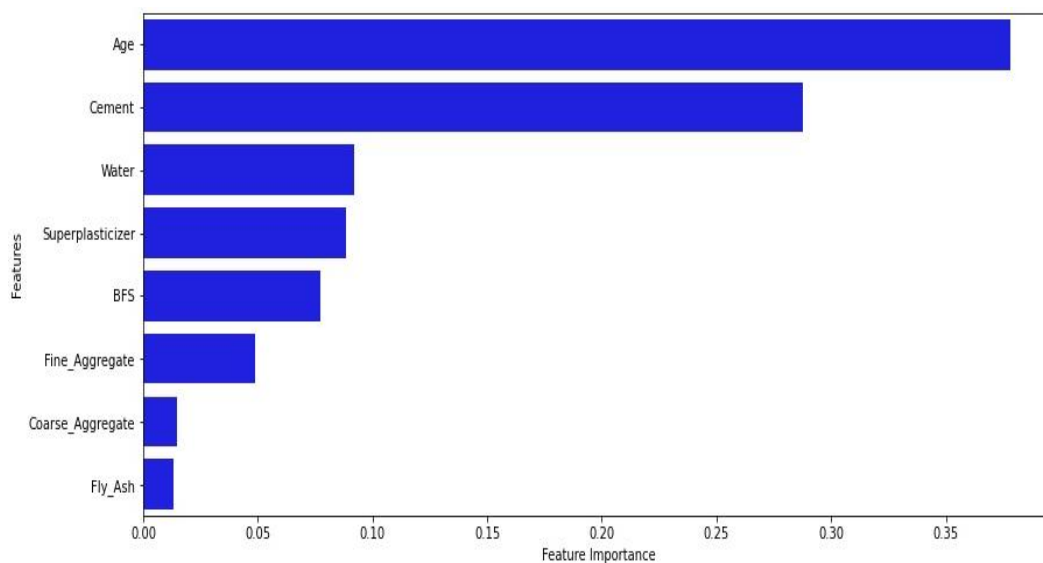


Figure 6. Variable Importance of the GB Regression Model

### 2.2.5. K-Nearest Neighbor Regression

K-Nearest Neighbors (KNN) is a distance-based regression algorithm [30], [31]. For the training set described earlier, the K nearest neighbor regression algorithm works as follows: Given a new input feature vector  $x^*$ , the K nearest neighbors to  $x^*$  are identified from the dataset based on a distance metric, such as Euclidean distance:

$$d(x_i, x^*) = \|x_i - x^*\| \quad (9)$$

The predicted output  $y^*$  for the new input  $x^*$  is computed as the mean of the target values of the K closest neighbors:

$$y^* = 1/K \sum_{i=1}^K y_i \quad (10)$$

where  $K=7$  is the number of nearest neighbors used.

### 2.3. Model Testing and Evaluation

Testing and evaluating the efficiency and performance of a designed ML model is a key step of the ML lifecycle to ensure accurate and reliable predictions. To ensure accuracy and reliability, there are key performance metrics used. Such performance metrics are the mean absolute error (MAE), mean absolute percentage error (MAPE), root mean squared error (RMSE), and the R-squared ( $R^2$ ) [28]. These are defined in Equations 11 – 14.

- i. MAE is the deviations between paired observations [32].

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (11)$$

where  $\hat{y}_i$  is the predicted value;  $y_i$  is the actual value;  $n$  is the number of fitted observations.

- ii. MAPE is the average of the absolute percentage errors of predictions [28].

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{\hat{y}_i - y_i}{y_i} \right| \quad (12)$$

- iii. RMSE is the standard deviation of the predicted errors defined in Equation 13.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}} \quad (13)$$

Generally, the lower the MAE, MAPE, and RMSE values, the better the prediction ability and capacity of the models. The model with the lowest values is said to have done the best fitting.

- iv. R-squared ( $R^2$ ) is the coefficient of determination and ranges between 0 and 1. A value of 1 means the model made predictions without any error.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (14)$$

where  $\bar{y}_i$  is the mean of the actual value. A higher value of  $R^2$  denotes a better fit.

## 3. Results and Discussion

This section analyzes the outcomes and performances of the various regression models employed for predicting CCS, based on the studies done in the above sections.

Cross-validation is a model validation method for assessing how the models generalize to an independent dataset. In Figure 7, the 10-fold CV had the best  $R^2$  performances across all the regression models except for the LR model where the 3-fold and 5-fold cross-validation outperformed it. The gradient boosting regression technique had the best performance for the 3-fold, 5-fold, and 10-fold splits with values of 0.9018, 0.9143, and 0.9160 respectively compared with the other models. The random forest had the second-best performance for all cross-validation splits used with values 0.8799, 0.8878, and 0.8914 while the linear regression had the worst performance having achieved  $R^2$  values of 0.6017, 0.5935, and 0.5892 respectively for the 3, 5 and 10-fold CVs.

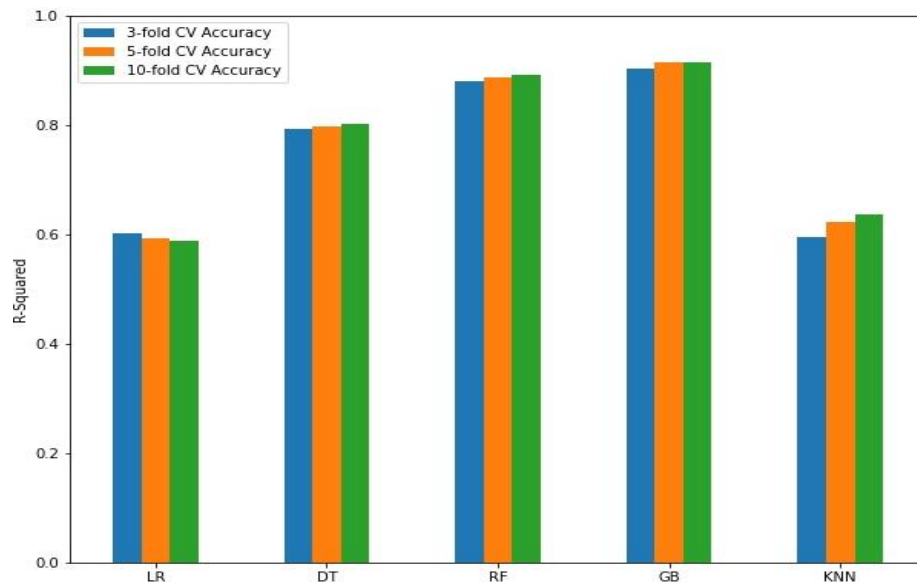


Figure 7. R<sup>2</sup> Score for Cross Validation on the Regression Models

The statistical results for the performance evaluation and testing performed on the models are presented in Table 3. It was observed from the results that the GB regression model had the lowest MAE, RMSE, and MAPE with values of 2.6776, 4.3523, and 0.0919 respectively, followed by the random forest technique with values of 3.4877, 5.1773, and 0.1224 respectively. The linear regression had the worst performance, achieving values of 8.8953, 11.1913, and 0.3469 for the MAE, RMSE, and MAPE, respectively. Likewise, GB, RF, and DT models achieved the highest value of the R-squared with 0.9365, 0.9101, and 0.8784 respectively. The KNN was the fourth-ranked performing regression model for the prediction of CCS while LR performed worst.

Table 3. Statistical Analysis of Predictive Models

ML Approaches	Training				Testing			
	MAE	R <sup>2</sup>	RMSE	MAPE	MAE	R <sup>2</sup>	RMSE	MAPE
LR	7.9599	0.6099	10.0018	0.3080	8.8953	0.5802	11.1913	0.3469
DT	0.0960	0.9964	0.9620	0.0026	3.8500	0.8784	6.0237	0.1390
RF	1.3584	0.9839	2.0323	0.0471	3.4877	0.9101	5.1773	0.1224
GB	0.8823	0.9912	1.5009	0.0316	<b>2.6776</b>	<b>0.9365</b>	<b>4.3523</b>	<b>0.0919</b>
KNN	0.0944	0.9964	0.9615	0.0026	5.8132	0.7592	8.4758	0.2255

For CCS prediction, innumerable works and methods have been done and applied. The comparison of our results with previous studies using individual traditional machine learning regressors is shown in Table 4. From the results, our approach had the best performance among the traditional ML approaches employed having the least errors. However, a study in [20] using a deep convolutional neural network had the overall best R-squared value of 0.97. This shows that deep learning has greater potential for more accurate prediction, albeit at a greater computational cost.

Table 4. Result Comparison in Previous Studies

Study	Algorithm	Performance Metrics			
		MAE	R <sup>2</sup>	RMSE	MAPE
Our work	Best Model: GB	<b>2.68</b>	0.94	<b>4.35</b>	<b>9.19%</b>
Khan et al. [15]	BR	4.26	0.92	5.69	-
Muliauwan et al. [19]	ANN	-	0.91	-	-
Zeng et al. [20]	DCNN	-	<b>0.97</b>	-	-

For a real-time deployment of the model, streamlit, a framework for machine learning, is used to develop interactive web applications. The best-performing model which is the GB regression model was loaded. The developed and deployed web



application is available at the website <https://jblideal-concrete-compressive-strength-cement-strength-app-1i4gpn.streamlit.app/>. The web application is shown in Figure 8 with the input variables set and prediction of CCS.

The deployment is of great benefit for concrete strength determination in practice under normal conditions as it allows determining the concrete compressive strength with different mixture contents and ratios before mixing the different components. Real-time prediction allows varying all inputs or mixtures to achieve the desired target. However, caution must be taken in the use of the tools as it is important that the actual components match the characteristics of the input variables trained and deployed. Likewise, all conditions and guidelines should be adhered to avoid miscalculations or the concrete strength.

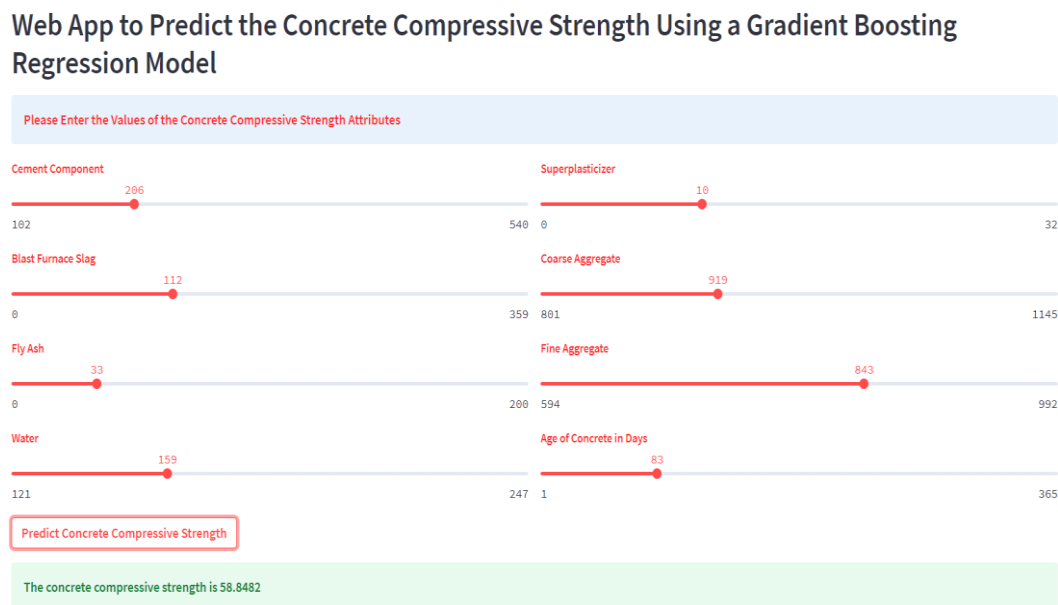


Figure 8. Deployed Web App with Prediction

#### 4. Conclusion

A possible alternative for forecasting concrete strength from non-destructive testing data is machine learning. However, a lack of consensus on which machine learning algorithms are most appropriate for this task, and a need for more comparative studies to evaluate the performance of different methods and identify best practices for applying them in practice. This study addressed this research gap by conducting a comparative analysis of ML methods for determining concrete strength. The study showed that the gradient-boosting regression model had the best level of accuracy for predicting the CCS, followed by the random forest technique. In contrast, the linear regression had the worst performance. These findings have important implications for improving the accuracy and cost-effectiveness of methods for assessing the quality of concrete structures, improving safety, and reducing maintenance costs over the lifetime of these structures. The developed and deployed web application for the best-performing model provides an example of how these models can be used in real-time deployment for practical applications. Further research could explore the effects of external factors such as temperature, humidity, and curing time on the accuracy of the ML models and investigate the generalizability of these findings to other types of concrete.

#### References

- [1] ACI, *AMERICAN CONCRETE INSTITUTE. ACI 318: Building Code Requirements for Structural Concrete*. American Concrete Institute, 2008.
- [2] J. Zhang, Y. Zhao, and H. Li, "Experimental Investigation and Prediction of Compressive Strength of Ultra-High Performance Concrete Containing Supplementary Cementitious Materials," *Adv. Mater. Sci. Eng.*, vol. 2017, 2017, doi: 10.1155/2017/4563164.
- [3] B. G. Aiyer, D. Kim, N. Karingattikal, P. Samui, and P. R. Rao, "Prediction of compressive strength of self-compacting concrete using least square support vector machine and relevance vector machine," *KSCE J. Civ. Eng.*, vol. 18, no. 6, pp. 1753–1758, 2014, doi: 10.1007/s12205-014-0524-0.
- [4] C. Bilim, C. D. Atiş, H. Tanyildizi, and O. Karahan, "Predicting the compressive strength of ground granulated blast furnace slag concrete using artificial neural network," *Adv. Eng. Softw.*, vol. 40, no. 5, pp. 334–340, 2009, doi: 10.1016/j.advengsoft.2008.05.005.
- [5] R. Mustapha and M. EL Aroussi, "High-Performance Concrete Compressive Strength Prediction Based Weighted Support Vector Machines," *Int. J. Eng. Res. Appl.*, vol. 07, no. 01, pp. 68–75, 2017, doi: 10.9790/9622-0701016875.
- [6] S. Popovics, "Analysis of the concrete strength versus water-cement ratio relationship," *ACI Mater. J.*, vol. 87, no. 5, pp. 517–529, 1990, doi: 10.14359/1944.

- [7] J. P. Zaniewski, *MATERIALS FOR CIVIL AND 3 rd Edition Michael S. Mamlouk Arizona State University*. 2011.
- [8] R. Kozul and D. Darwin, "Effects of Aggregate Type, Size and Content on Concrete Strength and Fracture Energy," 1997.
- [9] A. Fernández-Jiménez and A. Palomo, "Characterisation of fly ashes. Potential reactivity as alkaline cements," *Fuel*, vol. 82, no. 18, pp. 2259–2265, 2003, doi: 10.1016/S0016-2361(03)00194-7.
- [10] J. M. Fox, "Fly Ash Classification-Old and New Ideas," in *Fly Ash Classification – Old and New Ideas*, 2017, pp. 1–15.
- [11] A. M. Zeyad, "Effect of curing methods in hot weather on the properties of high-strength concretes," *J. King Saud Univ. - Eng. Sci.*, vol. 31, no. 3, pp. 218–223, 2019, doi: 10.1016/j.jksues.2017.04.004.
- [12] B. A. Young, A. Hall, L. Pilon, P. Gupta, and G. Sant, "Can the compressive strength of concrete be estimated from knowledge of the mixture proportions?: New insights from statistical analysis and machine learning methods," *Cem. Concr. Res.*, vol. 115, no. July, pp. 379–388, 2019, doi: 10.1016/j.cemconres.2018.09.006.
- [13] M. A. DeRousseau, J. R. Kasprzyk, and W. V. Srubar, "Computational design optimization of concrete mixtures: A review," *Cem. Concr. Res.*, vol. 109, pp. 42–53, 2018, doi: 10.1016/j.cemconres.2018.04.007.
- [14] M. Z. Naser, "An engineer's guide to eXplainable Artificial Intelligence and Interpretable Machine Learning: Navigating causality, forced goodness, and the false perception of inference," *Autom. Constr.*, vol. 129, no. September, 2021, doi: 10.1016/j.autcon.2021.103821.
- [15] K. Khan, W. Ahmad, M. N. Amin, F. Aslam, A. Ahmad, and M. A. Al-Faiad, "Comparison of Prediction Models Based on Machine Learning for the Compressive Strength Estimation of Recycled Aggregate Concrete," *Materials (Basel)*, vol. 15, no. 10, pp. 1–36, 2022, doi: 10.3390/ma15103430.
- [16] L. Chi *et al.*, "Machine learning prediction of compressive strength of concrete with resistivity modification," *Mater. Today Commun.*, vol. 36, p. 106470, Aug. 2023, doi: 10.1016/j.mtcomm.2023.106470.
- [17] P. G. Asteris, A. D. Skentou, A. Bardhan, P. Samui, and K. Pilakoutas, "Predicting concrete compressive strength using hybrid ensembling of surrogate machine learning models," *Cem. Concr. Res.*, vol. 145, no. October 2020, p. 106449, 2021, doi: 10.1016/j.cemconres.2021.106449.
- [18] D.-C. Feng *et al.*, "Machine learning-based compressive strength prediction for concrete: An adaptive boosting approach," *Constr. Build. Mater.*, vol. 230, p. 117000, Jan. 2020, doi: 10.1016/j.conbuildmat.2019.117000.
- [19] H. N. Muliauwan, D. Prayogo, G. Gaby, and K. Harsono, "Prediction of Concrete Compressive Strength Using Artificial Intelligence Methods," *J. Phys. Conf. Ser.*, vol. 1625, no. 1, p. 012018, Sep. 2020, doi: 10.1088/1742-6596/1625/1/012018.
- [20] Z. Zeng *et al.*, "Accurate prediction of concrete compressive strength based on explainable features using deep learning," *Constr. Build. Mater.*, vol. 329, 2022, doi: 10.1016/j.conbuildmat.2022.127082.
- [21] I.-C. Yeh, "Modeling of Strength of High-Performance Concrete Using Artificial Neural Networks," *Cem. Concr. Res.*, vol. 28, no. 12, pp. 1797–1808, 1998.
- [22] I.-C. Yeh, "Concrete Compressive Strength." UCI Machine Learning Repository, 2007, doi: 10.24432/C5PK67.
- [23] D. Maulud and A. M. Abdulazeez, "A Review on Linear Regression Comprehensive in Machine Learning," *J. Appl. Sci. Technol. Trends*, vol. 1, no. 4, pp. 140–147, 2020, doi: 10.38094/jastt1457.
- [24] E. Pekel, "Estimation of soil moisture using decision tree regression," *Theor. Appl. Climatol.*, vol. 139, no. 3–4, pp. 1111–1119, 2020, doi: 10.1007/s00704-019-03048-8.
- [25] M. Rakhra *et al.*, "Crop Price Prediction Using Random Forest and Decision Tree Regression:-A Review," *Mater. Today Proc.*, no. xxxx, 2021, doi: 10.1016/j.matpr.2021.03.261.
- [26] V. Rodriguez-Galiano, M. Sanchez-Castillo, M. Chica-Olmo, and M. Chica-Rivas, "Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines," *Ore Geol. Rev.*, vol. 71, pp. 804–818, 2015, doi: 10.1016/j.oregeorev.2015.01.001.
- [27] B. Singh, P. Sihag, and K. Singh, "Modelling of impact of water quality on infiltration rate of soil by random forest regression," *Model. Earth Syst. Environ.*, vol. 3, no. 3, pp. 999–1004, 2017, doi: 10.1007/s40808-017-0347-3.
- [28] J. Cai, K. Xu, Y. Zhu, F. Hu, and L. Li, "Prediction and analysis of net ecosystem carbon exchange based on gradient boosting regression and random forest," *Appl. Energy*, vol. 262, no. 114566, pp. 1–14, 2020, doi: 10.1016/j.apenergy.2020.114566.
- [29] U. Singh, M. Rizwan, M. Alaraj, and I. Alsaïdan, "A machine learning-based gradient boosting regression approach for wind power production forecasting: A step towards smart grid environments," *Energies*, vol. 14, no. 16, pp. 1–21, 2021, doi: 10.3390/en14165196.
- [30] L. E. de Oliveira Aparecido, G. de Souza Rolim, J. R. da Silva Cabral De Moraes, C. T. S. Costa, and P. S. de Souza, "Machine learning algorithms for forecasting the incidence of Coffea arabica pests and diseases," *Int. J. Biometeorol.*, vol. 64, no. 4, pp. 671–688, 2020, doi: 10.1007/s00484-019-01856-1.
- [31] C. Araújo, C. Soares, I. Pereira, D. Coelho, M. Â. Rebelo, and A. Madureira, "A Novel Approach for Send Time Prediction on Email Marketing," *Appl. Sci.*, vol. 12, no. 8310, pp. 1–13, 2022, doi: 10.3390/app12168310.
- [32] A. O. Oyedeji, A. M. Salami, O. Folorunsho, and O. R. Abolade, "Analysis and Prediction of Student Academic Performance Using Machine Learning," *JITCE (Journal Inf. Technol. Comput. Eng.)*, vol. 4, no. 01, pp. 10–15, 2020, doi: 10.25077/jitce.4.01.10-15.2020.

**Author(s) Contributions**

Ajibola Oyedeji: Conceptualization, Methodology, Writing – Original Draft, Software

Adekunle David: Conceptualization, Methodology, Writing – Original Draft

Ositola Osifeko: Methodology, Writing – Original Draft, Software

Abisola Olayiwola: Methodology, Writing – review & editing

Omobolaji Opafola: Methodology, Writing – review & editing

**Acknowledgments**

Not Applicable

**Conflict of Interest Notice**

Authors declare that there is no conflict of interest regarding the publication of this paper.

**Ethical Approval**

It is declared that during the preparation process of this study, scientific and ethical principles were followed, and all the studies benefitted from are stated in the bibliography.

**Availability of data and materials**

Not Applicable

**Plagiarism Statement**

This article has been scanned by iThenticate™.