



Prediction of Concrete Compressive Strength for Ready-Mix Concrete: A Comparative Evaluation of Algorithms with Explainability Analysis

Sandesh Patil¹, Dr. R. P. Singh Kushwah²

¹Research Scholar, Department of Civil Engineering, CSMU, Panvel, Mumbai, Maharashtra, India

²Professor and Dean, Engineering & Technology, CSMU, Panvel, Mumbai, Maharashtra, India

Emails: sspatilcivil@gmail.com¹, dce@csmu.ac.in²

Article history

Received: 02 August 2026

Accepted: 31 August 2026

Published: 03 October 2026

Keywords:

Ready-mix concrete;
compressive strength;
machine learning; stacked
ensemble; gradient
boosting; explainable
machine learning; quality
control.

Abstract

Ready-mix concrete is accepted or rejected on cube strengths that arrive weeks after placement, so an estimate available at the batching plant would be worth having. This paper benchmarks twelve machine learning algorithms for that purpose on the public concrete compressive strength database, under one preprocessing and evaluation protocol. Twenty-five exact duplicates were removed before any split was drawn, leaving 1005 records divided 804 for training and 201 for testing. A stacked ensemble of XGBoost, LightGBM, CatBoost and extremely randomized trees was the most accurate model, reaching $R^2 = 0.942$, RMSE = 4.15 MPa and MAE = 2.65 MPa, and leading in 15 of 20 repeated random splits. LightGBM and XGBoost followed at 0.938 and CatBoost at 0.932, while the multilayer perceptron and support vector regression reached about 0.87 and the linear baselines only 0.580. The more useful finding concerns how such results should be read. Tested with the corrected resampled t-test over the repeated splits, the stacked ensemble cannot be separated from LightGBM ($p = 0.207$), although an ordinary paired test on a single split declares a comparable difference highly significant. Permutation importance, partial dependence and SHapley Additive exPlanations all rank age, cement and water first, with mean absolute SHAP values of 8.25, 6.93 and 3.74 MPa, and adding five domain-driven ratio features produced no detectable gain. With an RMSE comparable to reported plant standard deviations of 1.92 to 6.57 MPa, a model of this accuracy suits pre-dispatch screening within the IS 10262 and IS 4926 framework rather than acceptance testing.

1. Introduction

Compressive strength decides whether a batch of concrete is accepted, yet the standard check, crushing cubes after 28 days in accordance with IS 516 (BIS, 2021), is retrospective. By the time a shortfall is found the concrete is usually already in the structure. The delay matters most for ready-mix concrete. IS 4926 (BIS, 2003) requires producers to run a documented quality-control system, but strength variability still differs sharply between plants: across four plants in Ahmedabad and Delhi,

Sarkar (2008) measured standard deviations from 1.92 to 6.57 MPa, with the poorest control at a fully automatic plant working under delivery pressure. Mix design itself still rests on empirical relations. Abrams' water-cement ratio law (Abrams, 1918) and Féret's earlier void-based formulation (Féret, 1892) capture first-order behaviour, and IS 10262 (BIS, 2019) is built on relations of that kind, but they are less dependable for mixtures that combine several binders with chemical admixtures. Machine

learning learns those interactions from data instead of prescribing them. Since Yeh applied neural networks to high-performance concrete (Yeh, 1998), the approach has been extended to random forests (Breiman, 2001), gradient boosting (Friedman, 2001), XGBoost (Chen and Guestrin, 2016) and, more recently, to the second generation of boosted libraries, LightGBM (Ke et al., 2017) and CatBoost (Prokhorenkova et al., 2018), and to stacked combinations of them (Wolpert, 1992). Several reviews now map the field (Ben Chaabene et al., 2020; Nunez et al., 2021; Li et al., 2022; Moein et al., 2023; Tipu and Paruthi, 2026), and test R^2 above 0.90 is routine on laboratory databases (Feng et al., 2020; Islam et al., 2024; Abioye et al., 2025; Nikoopayan Tak et al., 2025). Three problems persist behind those numbers. Algorithms are compared under different preprocessing and splits, so it is hard to tell whether a method is better or merely better tuned; duplicated records left in place inflate the apparent accuracy, an effect Kojić and Milošević (2026) have now quantified. Differences between leading models are rarely tested by a procedure that respects the dependence between repeated splits (Dietterich, 1998; Nadeau and Bengio, 2003). And the largely computational literature seldom connects its results to Indian ready-mix practice, where acceptance is framed by IS 10262 and IS 4926. The proposed work addresses all three: twelve algorithms from six families are benchmarked under one protocol with duplicates removed before splitting, the differences are tested both conventionally and with the corrected resampled test, and the outcome is read against what a plant can actually use.

2. Related Work

Linear regression is the natural baseline but cannot represent the curved strength–age relationship or the interaction between water content and binder type, and non-linear methods regularly outperform it (Chithra et al., 2016). Chou et al. (2011) compared neural networks, support vector machines, regression and tree-based meta-learners on data from 17 laboratories and found additive regression trees the most accurate and the least prone to overfitting. Tree ensembles have led ever since. Feng et al. (2020) reported $R^2 = 0.982$ with AdaBoost on the 1030-mixture database using a 90:10 split, Khodaparasti et al. (2023) improved on random forest, and Nguyen et al. (2021) found

gradient boosting and XGBoost ahead of SVR and a multilayer perceptron. Because most Indian ready-mix concrete contains fly ash or slag, blended-mixture studies are directly relevant (Ghunimat et al., 2023; Rathakrishnan et al., 2022; Ahmed et al., 2023). Recent work pairs accuracy with explanation. Islam et al. (2024) reached $R^2 = 0.959$ with CatBoost and used SHAP to rank age, cement and superplasticizer first. Others report much the same: 0.935 for tuned XGBoost (Abioye et al., 2025), 0.94 for gradient boosting (Nikoopayan Tak et al., 2025), 0.939 for CatBoost on site records (Fu et al., 2025), and 0.959 for LightGBM with engineered features (Hereiz et al., 2026). Neural networks have kept pace where they are tuned properly: Naved et al. (2024) showed systematic hyperparameter optimization lifts deep networks materially, and Ranasinghe et al. (2024) explained a deep model for slag–ash geopolymer concrete with SHAP. Language models have begun to appear at the data-preparation stage (Zaman et al., 2026). Deployable tools have followed (Elshaarawy et al., 2024; Hamed et al., 2026; Abbas and Barbulescu, 2026), and Zhang et al. (2021) took the practical route of predicting seven-day strength from field records. Set against these figures, DeRousseau et al. (2019) reached only $R^2 = 0.51$ on field-placed concrete, and Young et al. (2019) found production records carry more unexplained variance than laboratory data. High laboratory accuracy is not plant-level reliability, and the gap is measured in tenths of R^2 .

3. Materials and Methods

3.1. Dataset and Pre-processing

The study uses the concrete compressive strength database compiled by Yeh (1998) and distributed through the UCI Machine Learning Repository (Yeh, 2007): 1030 laboratory mixtures giving the contents of cement, blast-furnace slag, fly ash, water, superplasticizer, coarse and fine aggregate, the age at testing and the measured strength (Table 1). Ages span 1 to 365 days, 419 of the unique records being at 28 days, and strengths run from 2.33 to 82.60 MPa. These fields correspond to the main entries of an Indian batch record, where the equivalent materials would meet IS 3812 (BIS, 2013), IS 16714 (BIS, 2018), IS 9103 (BIS, 1999) and IS 383 (BIS, 2016). No values were missing. Twenty-five rows were exact copies of other rows and were removed before any partition was drawn.

A duplicated mixture that falls on both sides of a split makes the test error look better than it is. Kojić and Milošević (2026) measured that effect on a different dataset and found it larger than any difference between algorithms, which is why we treat the order of operations as part of the method rather than housekeeping. The remaining 1005 records were split 804 for training and 201 for testing with a fixed seed. Inputs were standardized inside each model's pipeline, so scaler parameters were estimated from training data only, in the final fit and inside every cross-validation fold.

3.2. Algorithms and Evaluation

Twelve algorithms were implemented, covering six families (Table 2). Multiple linear and ridge regression (Hoerl and Kennard, 1970) serve as interpretable baselines. A depth-limited decision tree, a random forest (Breiman, 2001) and an extremely randomized forest (Geurts et al., 2006) cover single-tree and bagged learners, and support vector regression with an RBF kernel (Drucker et al., 1997) covers kernel methods. The boosted ensembles are gradient boosting (Friedman, 2001), XGBoost (Chen and Guestrin, 2016), LightGBM (Ke et al., 2017) and CatBoost (Prokhorenkova et al., 2018). The neural network is a multilayer perceptron with three hidden layers, trained by back-propagation (Rumelhart et al., 1986) with the Adam optimizer (Kingma and Ba, 2015). The twelfth is a stacked ensemble (Wolpert, 1992) whose base learners are XGBoost, LightGBM, CatBoost and the extremely randomized forest, combined by a ridge meta-learner fitted on out-of-fold predictions only. Hyperparameters were fixed in advance from values common in the literature and

were never tuned against the test set. The analysis ran in Python 3.11 with scikit-learn 1.8.0 (Pedregosa et al., 2011), XGBoost 3.2.0, LightGBM 4.7.0, CatBoost 1.2.10 and SHAP 0.51.0.

Performance was measured with RMSE, MAE, R^2 and MAPE, defined in Eqs. (1) to (4), where y_i and $\hat{y}_i$ are the measured and predicted strengths of record i , $\bar{y}$ is the mean measured strength and n the number of records.

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

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

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (3)$$

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (4)$$

Stability was assessed by five-fold cross-validation on the training set and by twenty further random 80:20 splits. Differences among the three best models were tested on the single held-out split with a paired t-test and, because the residuals were not normal (Shapiro–Wilk $p < 0.001$), the Wilcoxon signed-rank test. Repeated splits share training data, so the twenty differences are not independent and an ordinary t-test on them is anti-conservative. The corrected resampled statistic of Nadeau and Bengio (2003) was used instead. Its variance term is inflated by $(1/k + n_{\text{test}}/n_{\text{train}})$, which for these partition sizes multiplies the estimated standard error by roughly 2.4. Interpretation used impurity importance from the random forest, permutation importance on the test set (Strobl et al., 2007), SHAP values (Lundberg and Lee, 2017) and partial dependence (Friedman, 2001). Table 1 shows Variables of The Database and Their Statistics (1005 Unique Records)

Table 1 Variables of The Database and Their Statistics (1005 Unique Records)

S.No	Variable	Unit	Role	Min	Max	Mean	SD
1	Cement	kg/m ³	Input	102.0	540.0	278.6	104.3
2	Blast-furnace slag	kg/m ³	Input	0.0	359.4	72.0	86.2
3	Fly ash	kg/m ³	Input	0.0	200.1	55.5	64.2
4	Water	kg/m ³	Input	121.8	247.0	182.1	21.3
5	Superplasticizer	kg/m ³	Input	0.0	32.2	6.0	5.9
6	Coarse aggregate	kg/m ³	Input	801.0	1145.0	974.4	77.6
7	Fine aggregate	kg/m ³	Input	594.0	992.6	772.7	80.3
8	Age at testing	days	Input	1	365	45.9	63.7
9	Compressive strength	MPa	Output	2.33	82.60	35.3	16.3

Table 2 Hyper Parameters used for Each Model

Model	Key hyperparameters
Multiple linear regression	Ordinary least squares; no regularization
Ridge regression	alpha = 1.0 (L2 penalty)
Decision tree	max_depth = 10; random_state = 42
Random forest	n_estimators = 300; max_depth = None; random_state = 42
Extra trees	n_estimators = 300; max_depth = None; random_state = 42
SVR (RBF kernel)	C = 50; epsilon = 0.5; kernel = RBF; gamma = "scale"
ANN (MLP)	hidden layers = (64, 32, 16); activation = ReLU; optimizer = Adam; learning_rate_init = 0.001; early_stopping = True (10% validation); max_iter = 3000
Gradient boosting	n_estimators = 300; learning_rate = 0.05; max_depth = 3
XGBoost	n_estimators = 400; learning_rate = 0.05; max_depth = 4; subsample = 0.85; colsample_bytree = 0.85
LightGBM	n_estimators = 400; learning_rate = 0.05; num_leaves = 31; min_child_samples = 10; subsample = 0.85; colsample_bytree = 0.85
CatBoost	iterations = 400; learning_rate = 0.05; depth = 6; random_seed = 42
Stacked ensemble	base learners = XGBoost, LightGBM, CatBoost, extremely randomized trees; meta-learner = ridge with cross-validated penalty; inner five-fold out-of-fold predictions

4. Results and Discussion

4.1. Model Comparison

The stacked ensemble was the most accurate model on the held-out set (Table 3): $R^2 = 0.942$, RMSE = 4.15 MPa, MAE = 2.65 MPa and MAPE = 8.9%. LightGBM and XGBoost were level on R^2 at 0.938 and separated only on MAE, 2.72 against 2.95 MPa, with CatBoost at 0.932 and the extremely randomized forest at 0.923. Gradient boosting (0.915) and random forest (0.910) formed the next group; the decision tree (0.868), the multilayer perceptron (0.867) and SVR (0.866) were effectively tied below them. Both linear models reached only 0.580, with errors more than two and a half times those of the stack. The problem is

strongly non-linear, and no amount of regularization repairs a linear fit to it. The meta-learner weights are worth reading. After cross-validated selection of the penalty, the ridge combiner placed 0.43 on CatBoost, 0.29 on LightGBM, 0.22 on XGBoost and 0.09 on the extremely randomized forest. CatBoost carries the largest weight despite ranking only fourth on its own, which is exactly what a stack should do: it rewards the base model whose errors are least correlated with the rest, not the one with the best solo score. Figure 1 gives the ranking and Figure 2 shows that the four best models track the measured values closely, with the widest scatter above about 50 MPa where records are sparse.

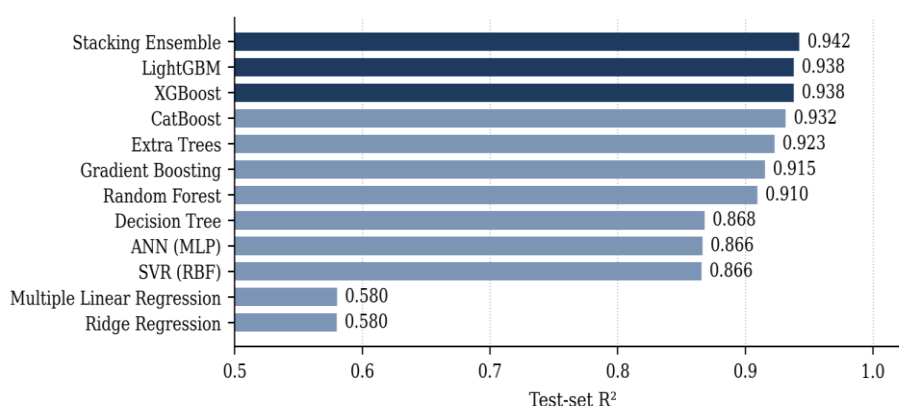
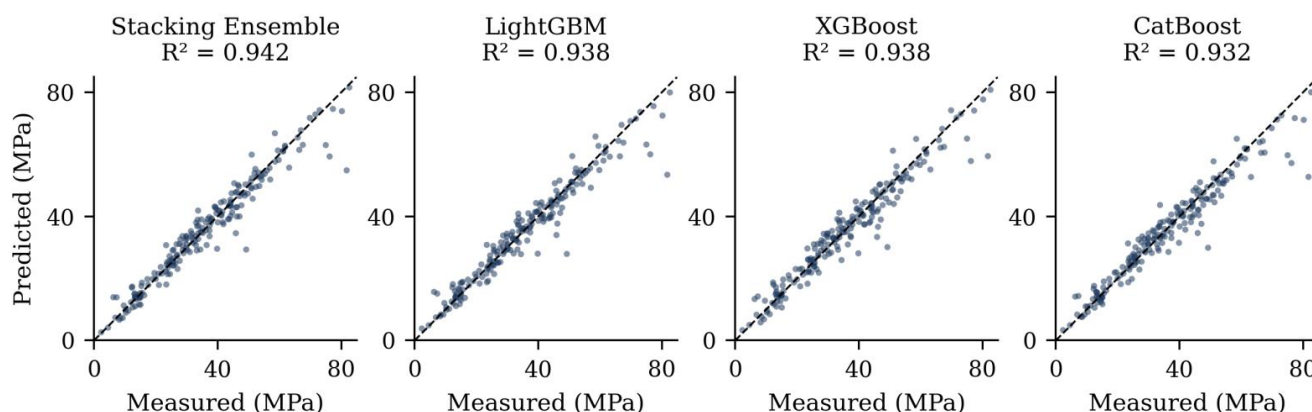
**Figure 1 Test-Set R^2 of the Twelve Models (Three Best in Dark Blue)**

Table 3 Performance of The Twelve Models On the Held-Out Test Set (n = 201)

Rank	Model	RMSE (MPa)	MAE (MPa)	Test R ²	MAPE (%)	Train R ²	5-fold CV R ² (mean $\pm$ SD)
1	Stacked ensemble	4.15	2.65	0.942	8.9	0.986	0.922 $\pm$ 0.017
2	LightGBM	4.31	2.72	0.938	9.5	0.992	0.914 $\pm$ 0.016
3	XGBoost	4.31	2.95	0.938	10.2	0.982	0.917 $\pm$ 0.020
4	CatBoost	4.52	2.91	0.932	9.6	0.973	0.919 $\pm$ 0.016
5	Extra trees	4.80	2.96	0.923	9.9	0.996	0.898 $\pm$ 0.023
6	Gradient boosting	5.02	3.65	0.915	12.1	0.958	0.899 $\pm$ 0.019
7	Random forest	5.20	3.52	0.910	12.3	0.984	0.886 $\pm$ 0.024
8	Decision tree	6.27	4.42	0.868	15.2	0.979	0.786 $\pm$ 0.037
9	ANN (MLP)	6.31	4.47	0.867	15.2	0.907	0.858 $\pm$ 0.011
10	SVR (RBF)	6.33	4.34	0.866	15.4	0.914	0.847 $\pm$ 0.018
11	Multiple linear regression	11.19	8.90	0.580	34.7	0.610	0.590 $\pm$ 0.032
12	Ridge regression	11.19	8.90	0.580	34.7	0.610	0.590 $\pm$ 0.032

Cross-validation was carried out on the 804 training records only.

**Figure 2** Predicted Versus Measured Compressive Strength On the Test Set for The Four Best-Performing Models

Five-fold cross-validation on the training set broadly reproduced the held-out ordering, though not exactly: the stack again led at 0.922 ± 0.017 , followed by CatBoost at 0.919 ± 0.016 , XGBoost at 0.917 ± 0.020 and LightGBM at 0.914 ± 0.016 . CatBoost therefore does better under cross-validation than its held-out rank suggests, which is consistent with its ordered-boosting scheme being

designed to resist the optimism that appears when a model is scored on records adjacent to those it was fitted on (Prokhorenkova et al., 2018). The spread of the four leading ensembles lies between 0.016 and 0.020, so none is noticeably less stable than the others, and all are far steadier than the decision tree (± 0.037) or the linear models (± 0.032). The gap between training and test accuracy is equally

informative. The decision tree fitted the training data almost perfectly ($R^2 = 0.979$) but fell to 0.868 on the test set. Averaging many such trees in a random forest raised test R^2 to 0.910, which is the variance reduction bagging is meant to deliver; drawing the split thresholds at random as well added a further 0.013. Boosting went a step beyond by correcting residual errors sequentially, and stacking added a final increment by exploiting the fact that the three boosted libraries make partly different mistakes. The multilayer perceptron generalized better than the single tree under cross-validation but did not reach the ensembles. We read that as a consequence of its pre-selected configuration rather than a ceiling for neural networks. Tuned deep networks have reported higher accuracy on comparable data (Naved et al., 2024; Hamed et al., 2026).

4.2. Stability and Statistical Significance

Twenty repeated splits (Table 4, Figure 3) confirm the ranking does not depend on one fortunate partition, and at the same time show how narrow the margins are. The stack had the highest mean test R^2 , 0.939 over a range of 0.911 to 0.956, and led in 15 of the 20 splits. LightGBM took first place in three splits at a mean of 0.936; XGBoost and CatBoost each led once, at 0.935 and 0.932. The top four are separated by 0.006 in mean R^2 while each varies by about 0.012 across splits. The leader board is decided by a difference smaller than the noise in any single measurement of it. That is the finding we would most want a reader to take away. Table 4 shows Test-set R^2 Over Twenty Additional Random 80:20 Splits

Table 4 Test-set R^2 Over Twenty Additional Random 80:20 Splits

Model	Mean R^2	SD	Minimum	Maximum	Ranked first
Stacked ensemble	0.939	0.012	0.911	0.956	15 / 20
LightGBM	0.936	0.013	0.904	0.952	3 / 20
XGBoost	0.935	0.012	0.906	0.951	1 / 20
CatBoost	0.932	0.012	0.908	0.951	1 / 20
Extra trees	0.917	0.013	0.896	0.936	0 / 20
Gradient boosting	0.914	0.011	0.892	0.934	0 / 20
Random forest	0.911	0.013	0.883	0.926	0 / 20
SVR (RBF)	0.868	0.019	0.808	0.896	0 / 20
ANN (MLP)	0.853	0.019	0.814	0.903	0 / 20
Decision tree	0.833	0.026	0.766	0.874	0 / 20
Ridge regression	0.593	0.043	0.501	0.655	0 / 20
Multiple linear regression	0.593	0.043	0.500	0.656	0 / 20

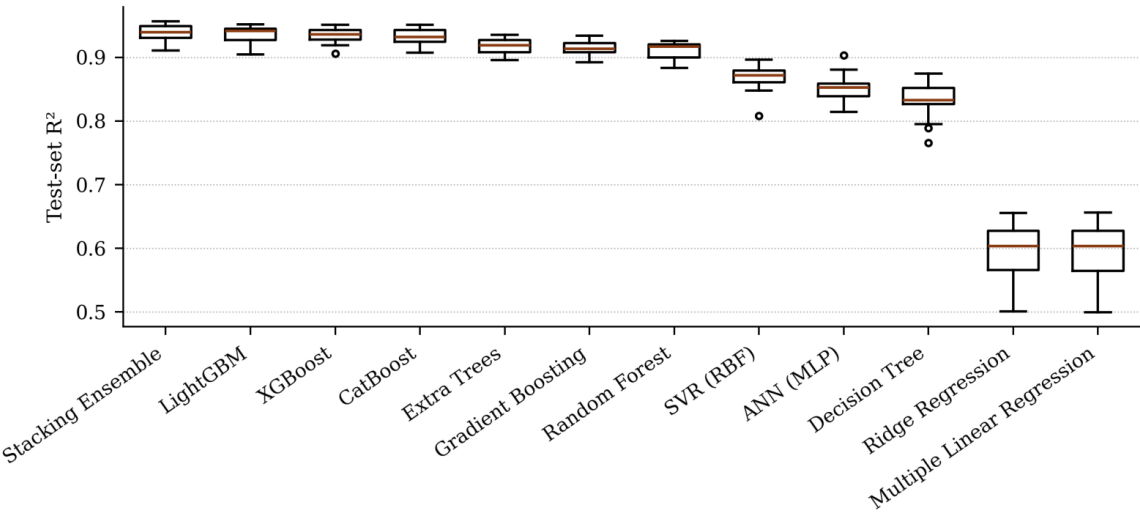


Figure 3 Distribution of Test-Set R^2 for Each Model Over the Twenty Repeated Splits

Table 5 sets the two testing regimes side by side, and they disagree in a way worth stating plainly. On the single held-out split the mean absolute error of the stack (2.65 MPa) was significantly lower than XGBoost's (2.95 MPa; $t = -3.88$, $p < 0.001$), with Wilcoxon agreeing. Repeat the comparison over twenty splits and apply the correction, and the stack keeps a significant edge over XGBoost only just ($\Delta R^2 = 0.0043$, $p = 0.027$), while its advantage over LightGBM is firmly non-significant ($\Delta R^2 = 0.0025$, $p = 0.207$). LightGBM against XGBoost is the clearest case: Wilcoxon on one split gives $p = 0.023$,

the corrected test over twenty gives $p = 0.531$. The practical reading is that the stack is the safest of the four, having led in three-quarters of the splits and never fallen below $R^2 = 0.911$, but that any of the three boosted libraries would serve almost as well. A plant already running LightGBM or XGBoost would gain little measurable accuracy from a stack and would pay for it in training time and in the loss of a single inspectable model. Table 5 shows The Three Best Models Under Single-Split And Repeated-Split Testing

Table 5 The Three Best Models Under Single-Split And Repeated-Split Testing

Model A	Model B	MAE A (MPa)	MAE B (MPa)	t (single split)	p (t-test)	P (Wilcoxon)	ΔR^2 (20 splits)	P (corrected)
Stacked ensemble	LightGBM	2.65	2.72	-0.77	0.439	0.568	+0.0025	0.207
Stacked ensemble	XGBoost	2.65	2.95	-3.88	< 0.001	< 0.001	+0.0043	0.027
LightGBM	XGBoost	2.72	2.95	-1.89	0.061	0.023	+0.0018	0.531

ΔR^2 is the mean difference in test R^2 over the twenty repeated splits; the corrected statistic is that of Nadeau and Bengio (2003).

4.3. What the Model Learned

All three importance measures (Figure 4) put age and cement well ahead of everything else. In the random forest they account for about two-thirds of the total impurity reduction (0.356 and 0.308). Permutation importance for the stack gives the same leading pair: shuffling age costs 0.758 of test R^2 and shuffling cement 0.538. SHAP puts the result in engineering units, with mean absolute values of 8.25 MPa for age and 6.93 MPa for cement against

3.74 MPa for water and 3.35 MPa for slag. Three measures, three methods, one ranking. Fly ash contributes only 0.35 MPa, which does not mean it has no effect: its influence is spread across interactions with age and cement, and attributing one variable at a time does not capture shared effects well. The ranking matches the SHAP findings of Islam et al. (2024), Abioye et al. (2025) and Hamed et al. (2026). (Figure 4)

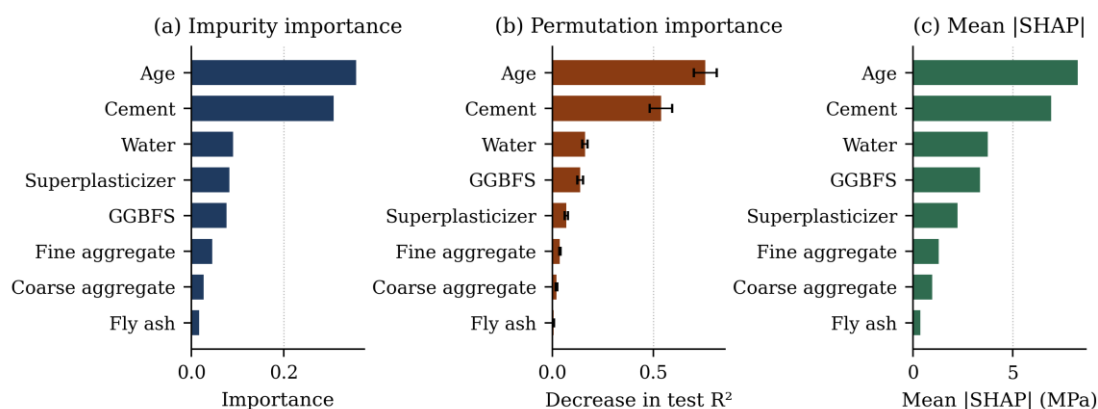


Figure 4 Feature Importance: (a) impurity-based from The Random Forest; (b) Permutation Importance for The Stacked Ensemble; (c) Mean Absolute SHAP Value

The partial dependence curves (Figure 5) show the stack has learned relationships consistent with established concrete behaviour without any of them being built in. Predicted strength rises with cement content, from about 24 MPa at 144 kg/m³ to 51 MPa at 477 kg/m³. It falls with water, from about 44 MPa at 146 kg/m³ to 30 MPa at 207 kg/m³, which is what Abrams’ law (Abrams, 1918) requires. Strength gain with age is steep at first, about 16 MPa at 1 day,

25 at 7 days and 38 at 28 days, then flattens beyond 56 days to roughly 50 MPa at one year. The further 8 MPa between 28 and 56 days is consistent with the late pozzolanic contribution of the slag and fly ash in many of the mixtures. Figure 5 shows Partial Dependence of the Predicted Strength of the Stacked Ensemble On Cement Content, Water Content and Age

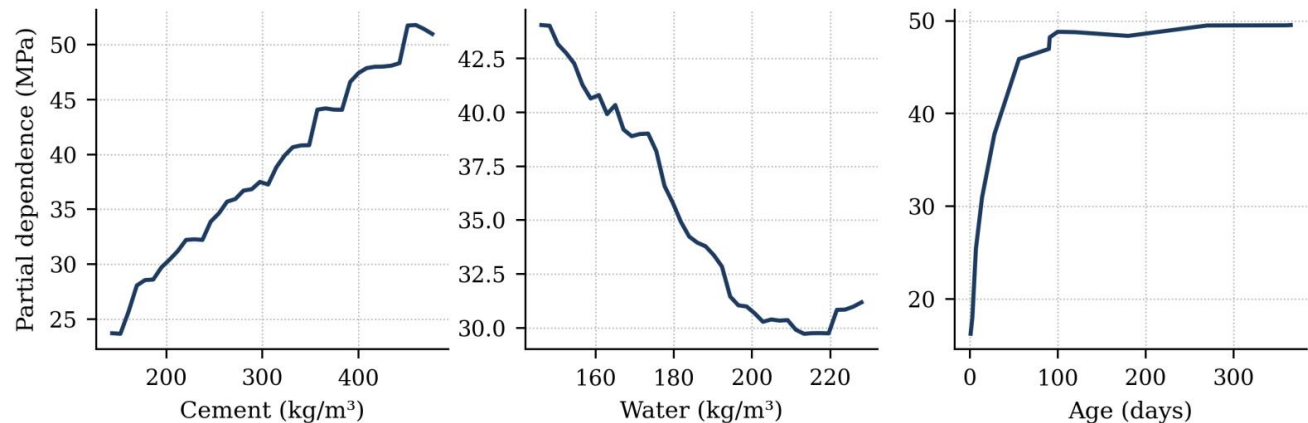


Figure 5 Partial Dependence of the Predicted Strength of the Stacked Ensemble On Cement Content, Water Content and Age

4.4. Engineered Features and Residuals

Reports that ratio features lift accuracy (Hereiz et al., 2026; Nikoopayan Tak et al., 2025) were treated as a hypothesis rather than a recipe. Five physically motivated variables were derived from the eight raw inputs and XGBoost was trained on both sets across the same twenty splits. Mean test R² rose from 0.9345 to 0.9368. The corrected test cannot distinguish that improvement of 0.0023 from zero (t = 0.58, p = 0.568; Table 6). We report it as a negative result rather than quietly dropping it. The negative result is worth recording: a gradient-boosted model with enough trees can already recover a ratio such as w/c from its two constituents, so engineered variables earn their place when they encode something the model cannot compose, or when they make the explanation easier for an engineer to read. Residuals of the stack (Figure 6)

were centred close to zero, mean +0.36 MPa with a standard deviation of 4.14 MPa, and showed no trend with the predicted value. Of the 201 test predictions, 88% fell within ±5 MPa of the measured strength and 69% within ±10%. The distribution is sharply peaked with a few large errors, the worst reaching 26.9 MPa, which explains the non-normality noted earlier. Errors varied with age: mean absolute error was 2.42 MPa for the 56 specimens tested at 7 days or earlier, 3.21 MPa for the 89 tested between 8 and 28 days, and 1.99 MPa for the 56 tested beyond 28 days. The 8-to-28-day band is the acceptance window. The largest errors therefore sit precisely where a screening tool would be asked to make its call, and any deployment should account for that rather than quote a single overall RMSE.

Table 6 Effect of Adding Five Domain-Driven Ratio Features (Xgboost, Twenty Repeated Splits)

Input set	Mean test R ²	SD	t (corrected)	p (corrected)
Eight raw inputs	0.9345	0.0115	—	—
Raw inputs + five ratio features	0.9368	0.0109	0.58	0.568

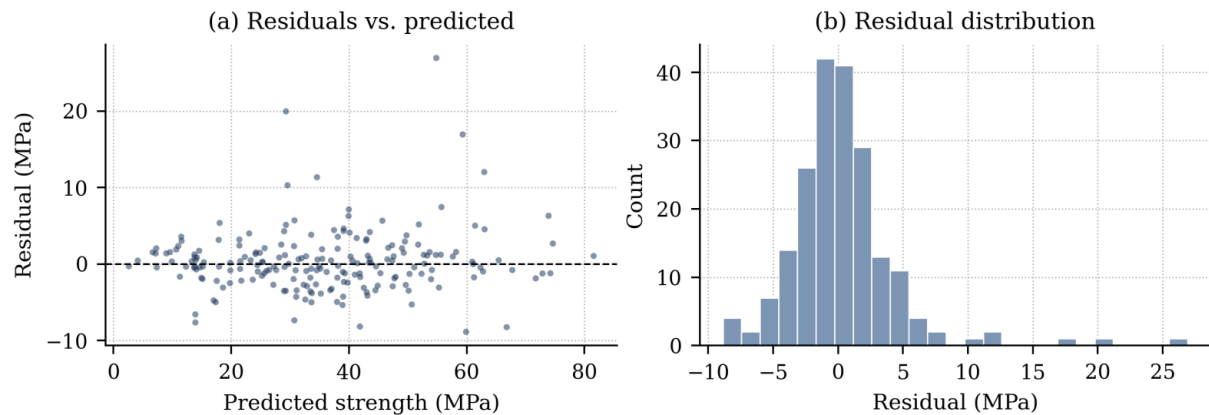


Figure 6 Residuals of The Stacked Ensemble: (a) Residual Versus Predicted Strength; (b) Residual Distribution

4.5. Comparison, Practical Use and Limitations

Table 7 places these results beside published work. The R^2 of 0.942 sits above the 0.935 of Abioye et al. (2025) and the 0.923 of the deployed ConcreteML model (Abbas and Barbulescu, 2026), and below the 0.959 of Islam et al. (2024) and Hereiz et al. (2026) and the 0.982 of Feng et al. (2020). Those comparisons should be read with the caution Section 4.2 established: split ratios differ, duplicate

handling is rarely stated, and several studies tune extensively where the present models used fixed settings. Two entries are more instructive than the accuracy column suggests. Kojić and Milošević (2026) report $R^2 = 0.793$ for a stack of identical construction on a de-duplicated dataset, so the gap to the present figure is a property of the data, not the method; and DeRousseau et al. (2019) reached 0.51 on field-placed concrete.

Table 7 Comparison with Selected Published Studies (Test-Set Results)

Study	Data	Best model	Reported performance
Feng et al. (2020)	Yeh database, 1030 records; 90:10 split	AdaBoost	$R^2 = 0.982$; MAPE = 6.78%
Islam et al. (2024)	High-performance concrete database	CatBoost	$R^2 = 0.959$; MAE = 3.0 MPa
Hereiz et al. (2026)	Yeh database plus 11 engineered features	LightGBM	$R^2 = 0.959$; RMSE = 3.47 MPa
Hamed et al. (2026)	1030 records; deep network with SHAP and PDP	Deep neural network	$R^2 = 0.946$
Nikoopayan Tak et al. (2025)	Yeh database, 911 records after preprocessing	Gradient boosting	$R^2 = 0.94$
Fu et al. (2025)	223 site records, Beijing metro project	CatBoost	$R^2 = 0.939$; RMSE = 2.71 MPa
Abioye et al. (2025)	Yeh database, 1030 records	XGBoost	$R^2 = 0.935$; RMSE = 3.55 MPa
Abbas and Barbulescu (2026)	5186 mixtures, 16 engineered descriptors	XGBoost (deployed)	$R^2 = 0.923$; RMSE = 4.48 MPa
Kojić and Milošević (2026)	603 SCC records, duplicates removed; 25 splits	Stacking ensemble	$R^2 = 0.793 \pm 0.038$
DeRousseau et al. (2019)	Field-placed concrete records	Random forest	$R^2 = 0.51$; RMSE $\approx$ 5.0 MPa
Present study	Yeh database, 1005 unique records; 80:20 plus 20 repeats	Stacked ensemble	$R^2 = 0.942$; RMSE = 4.15 MPa; MAE = 2.65 MPa

Values Are as Reported by The Respective Authors; Differences in Data, Split Ratio, Duplicate Handling and Tuning Limit Direct Comparison.

For practice, IS 10262 (BIS, 2019) requires mixes to be proportioned for a target mean strength of $f'_{ck} = f_{ck} + 1.65s$. A trained model adds a batch-level check. Before dispatch the actual batched quantities are entered, and any batch whose predicted strength falls more than an agreed margin below the target mean is flagged for the quality-control engineer rather than waiting on cube results. Such a check sits naturally inside the producer's system under IS 4926 (BIS, 2003). The margin matters. An RMSE of 4.15 MPa is the same order as the plant standard deviations of 1.92 to 6.57 MPa reported by Sarkar (2008). A model of this accuracy will therefore identify clearly deficient batches. It will not replace cube testing for acceptance under IS 456 (BIS, 2000) and IS 516 (BIS, 2021), and it cannot separate a marginal batch from an acceptable one. We would not claim otherwise. Its value is early warning and triage. Smaller plants without extensive laboratories stand to gain most, provided the model is first recalibrated on their own records. The main limitation is the data. The database holds laboratory mixtures compiled from published sources, not plant records: transport time, ambient temperature, site curing, retempering and batching tolerances are absent, and field studies show these reduce achievable accuracy considerably (DeRousseau et al., 2019; Young et al., 2019). Specimen geometry, cement type and grade, aggregate grading and curing temperature are not recorded either, and Indian acceptance rests on 150 mm cubes (BIS, 2021), so recalibration is required before predictions can be compared with cube results. Hyperparameters were fixed rather than optimized, which most likely disadvantaged the neural network and SVR. The repeated splits measure robustness to partitioning, not to a change of source.

Conclusion

Twelve algorithms were compared for predicting concrete compressive strength from mixture proportions and age, under one protocol that removed duplicates before splitting and combined held-out metrics, cross-validation, twenty repeated splits and two kinds of significance test. The stacked ensemble of XGBoost, LightGBM, CatBoost and extremely randomized trees was the most accurate, at $R^2 = 0.942$, RMSE = 4.15 MPa and MAE = 2.65 MPa, and it led in 15 of the 20 repeated splits; LightGBM and XGBoost followed at 0.938 and CatBoost at 0.932. Those margins do not survive a

correct significance test. Over twenty splits the corrected resampled t-test could not separate the stack from LightGBM ($p = 0.207$), while an ordinary paired test on a single split declared a comparable difference highly significant. Model choice among the leading ensembles should therefore rest on training time, ease of retraining and available tooling rather than on a third-decimal difference in R^2 , and removing 25 duplicated mixtures before splitting mattered more to the credibility of the reported numbers than the choice among those four algorithms did. Age, cement and water were the dominant inputs under all three importance measures, with mean absolute SHAP values of 8.25, 6.93 and 3.74 MPa, and five domain-driven ratio features produced no detectable gain. With an error comparable in size to typical plant standard deviations, a model of this kind is suited to pre-dispatch screening and early warning within the IS 10262 and IS 4926 framework, not to replacing cube-based acceptance. The next step is recalibration on batch records and IS 516 cube results from Indian ready-mix plants, with plant-measurable variables such as slump added to the inputs (Zviazhynski et al., 2025; Swati et al., 2026).

Acknowledgements

I am thanks to the Dr. R. P. Singh Kushwah Professor and Dean, Engineering & Technology, Department of Civil Engineering, Chhatrapati Shivaji Maharaj University, Panvel, Mumbai, for there valuable time and institutional support during my research work.

References

- [1]. Bureau of Indian Standards. (2021). IS 516 (Part 1/Sec 1):2021, Hardened concrete – Methods of test, Part 1: Testing of strength of hardened concrete, Section 1: Compressive, flexural and split tensile strength. New Delhi, India: BIS.
- [2]. Bureau of Indian Standards. (2003). IS 4926:2003, Ready-mixed concrete – Code of practice (2nd rev.). New Delhi, India: BIS.
- [3]. Sarkar, D. (2008, December). A framework for development of quality control model for Indian ready mixed concrete industry. New Building Materials & Construction World.
- [4]. Abrams, D. A. (1918). Design of concrete mixtures (Bulletin No. 1). Chicago, IL: Structural Materials Research Laboratory, Lewis Institute.

- [5]. Féret, R. (1892). Sur la compacité des mortiers hydrauliques. *Annales des Ponts et Chaussées*, 4, 5–164.
- [6]. Bureau of Indian Standards. (2019). IS 10262:2019, Concrete mix proportioning – Guidelines (2nd rev.). New Delhi, India: BIS.
- [7]. Yeh, I.-C. (1998). Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete Research*, 28(12), 1797–1808. [https://doi.org/10.1016/S0008-8846\(98\)00165-3](https://doi.org/10.1016/S0008-8846(98)00165-3)
- [8]. Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
- [9]. Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5), 1189–1232. <https://doi.org/10.1214/aos/1013203451>
- [10]. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). <https://doi.org/10.1145/2939672.2939785>
- [11]. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems* 30 (pp. 3146–3154).
- [12]. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: Unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems* 31 (pp. 6638–6648).
- [13]. Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5(2), 241–259. [https://doi.org/10.1016/S0893-6080\(05\)80023-1](https://doi.org/10.1016/S0893-6080(05)80023-1)
- [14]. Ben Chaabene, W., Flah, M., & Nehdi, M. L. (2020). Machine learning prediction of mechanical properties of concrete: Critical review. *Construction and Building Materials*, 260, Article 119889. <https://doi.org/10.1016/j.conbuildmat.2020.119889>
- [15]. Nunez, I., Marani, A., Flah, M., & Nehdi, M. L. (2021). Estimating compressive strength of modern concrete mixtures using computational intelligence: A systematic review. *Construction and Building Materials*, 310, Article 125279. <https://doi.org/10.1016/j.conbuildmat.2021.125279>
- [16]. Li, Z., Yoon, J., Zhang, R., Rajabipour, F., Srubar III, W. V., Dabo, I., & Radlińska, A. (2022). Machine learning in concrete science: Applications, challenges, and best practices. *npj Computational Materials*, 8, Article 127. <https://doi.org/10.1038/s41524-022-00810-x>
- [17]. Moein, M. M., Saradar, A., Rahmati, K., Mousavinejad, S. H. G., Bristow, J., Aramali, V., & Karakouzian, M. (2023). Predictive models for concrete properties using machine learning and deep learning approaches: A review. *Journal of Building Engineering*, 63, Article 105444. <https://doi.org/10.1016/j.jobbe.2022.105444>
- [18]. Tipu, R. K., & Paruthi, S. (2026). Advances in machine learning for predicting compressive strength of high-performance concrete: A review. *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, 1–46. <https://doi.org/10.1007/s40996-026-02188-y>
- [19]. Feng, D.-C., Liu, Z.-T., Wang, X.-D., Chen, Y., Chang, J.-Q., Wei, D.-F., & Jiang, Z.-M. (2020). Machine learning-based compressive strength prediction for concrete: An adaptive boosting approach. *Construction and Building Materials*, 230, Article 117000. <https://doi.org/10.1016/j.conbuildmat.2019.117000>
- [20]. Islam, M. M., Das, P., Rahman, M. M., Naz, F., Kashem, A., Nishat, M. H., & Tabassum, N. (2024). Prediction of compressive strength of high-performance concrete using optimization machine learning approaches with SHAP analysis. *Journal of Building Pathology and Rehabilitation*, 9(2), Article 94. <https://doi.org/10.1007/s41024-024-00445-z>
- [21]. Abioye, S. O., Babatunde, Y. O., Abikoye, O. A., Shaibu, A. N., & Bankole, B. J. (2025). Optimized machine learning algorithms with SHAP analysis for predicting compressive strength in high-performance concrete. *AI in Civil Engineering*, 4(1), Article 16. <https://doi.org/10.1007/s43503-025-00061-x>

- [22]. Nikoopayan Tak, M. S., Feng, Y., & Mahgoub, M. (2025). Advanced machine learning techniques for predicting concrete compressive strength. *Infrastructures*, 10(2), Article 26. <https://doi.org/10.3390/infrastructures10020026>
- [23]. Kojić, N., & Milošević, B. (2026). Compressive strength prediction of self-compacting concrete with recycled coarse aggregate using machine learning: Robust multi-split evaluation and data-leakage analysis of a stacking ensemble. *Technologies*, 14(8), Article 474. <https://doi.org/10.3390/technologies14080474>
- [24]. Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7), 1895–1923. <https://doi.org/10.1162/089976698300017197>
- [25]. Nadeau, C., & Bengio, Y. (2003). Inference for the generalization error. *Machine Learning*, 52(3), 239–281. <https://doi.org/10.1023/A:1024068626366>
- [26]. Chithra, S., Kumar, S. R. R. S., Chinnaraju, K., & Alfin Ashmita, F. (2016). A comparative study on the compressive strength prediction models for high performance concrete containing nano silica and copper slag using regression analysis and artificial neural networks. *Construction and Building Materials*, 114, 528–535. <https://doi.org/10.1016/j.conbuildmat.2016.03.214>
- [27]. Chou, J.-S., Chiu, C.-K., Farfoura, M., & Al-Taharwa, I. (2011). Optimizing the prediction accuracy of concrete compressive strength based on a comparison of data-mining techniques. *Journal of Computing in Civil Engineering*, 25(3), 242–253. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000088](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000088)
- [28]. Khodaparasti, M., Alijamaat, A., & Pouraminian, M. (2023). Prediction of the concrete compressive strength using improved random forest algorithm. *Journal of Building Pathology and Rehabilitation*, 8(2), Article 92. <https://doi.org/10.1007/s41024-023-00337-8>
- [29]. Nguyen, H., Vu, T., Vo, T. P., & Thai, H.-T. (2021). Efficient machine learning models for prediction of concrete strengths. *Construction and Building Materials*, 266, Article 120950. <https://doi.org/10.1016/j.conbuildmat.2020.120950>
- [30]. Ghunimat, D., Alzoubi, A. E., Alzboon, A., & Hanandeh, S. (2023). Prediction of concrete compressive strength with GGBFS and fly ash using multilayer perceptron algorithm, random forest regression and k-nearest neighbor regression. *Asian Journal of Civil Engineering*, 24(1), 169–177. <https://doi.org/10.1007/s42107-022-00495-z>
- [31]. Rathakrishnan, V., Beddu, S., & Ahmed, A. N. (2022). Predicting compressive strength of high-performance concrete with high volume ground granulated blast-furnace slag replacement using boosting machine learning algorithms. *Scientific Reports*, 12, Article 9539. <https://doi.org/10.1038/s41598-022-12890-2>
- [32]. Ahmed, H. U., Mostafa, R. R., Mohammed, A., Sihag, P., & Qadir, A. (2023). Support vector regression (SVR) and grey wolf optimization (GWO) to predict the compressive strength of GGBFS-based geopolymer concrete. *Neural Computing and Applications*, 35(3), 2909–2926. <https://doi.org/10.1007/s00521-022-07724-1>
- [33]. Fu, H., Zhou, X., Xu, P., & Sun, D. (2025). Prediction of compressive strength of concrete using explainable machine learning models. *Materials*, 18(21), Article 5009. <https://doi.org/10.3390/ma18215009>
- [34]. Hereiz, A., Elsayh, M., Hassan, K., & El-Metwally, S. (2026). Concrete strength prediction using domain-driven feature engineering and explainable LightGBM. *Scientific Reports*, 16, Article 27281. <https://doi.org/10.1038/s41598-026-68050-3>
- [35]. Naved, M., Asim, M., & Ahmad, T. (2024). Prediction of concrete compressive strength using deep neural networks based on hyperparameter optimization. *Cogent Engineering*, 11(1), Article 2297491.

- <https://doi.org/10.1080/23311916.2023.2297491>
- [36]. Ranasinghe, R. S. S., Kulasooriya, W. K. V. J. B., Perera, U. S., Ekanayake, I. U., Meddage, D. P. P., Mohotti, D., & Rathanayake, U. (2024). Eco-friendly mix design of slag-ash-based geopolymer concrete using explainable deep learning. *Results in Engineering*, 23, Article 102503. <https://doi.org/10.1016/j.rineng.2024.102503>
- [37]. Zaman, S., Eid, M. M., Mattar, E. A., Khafaga, D. S., & Towfek, S. K. (2026). Predicting concrete compressive strength using optimized deep learning and large language models. *Scientific Reports*, 16, Article 11076. <https://doi.org/10.1038/s41598-026-41072-7>
- [38]. Elshaarawy, M. K., Alsaadawi, M. M., & Hamed, A. K. (2024). Machine learning and interactive GUI for concrete compressive strength prediction. *Scientific Reports*, 14, Article 16694. <https://doi.org/10.1038/s41598-024-66957-3>
- [39]. Hamed, A. K., Alsaadawi, M. M., & Elshaarawy, M. K. (2026). Interpretable deep neural network deployment for concrete compressive strength prediction. *Journal of Building Pathology and Rehabilitation*, 11, Article 98. <https://doi.org/10.1007/s41024-025-00752-z>
- [40]. Abbas, M. M., & Barbulescu, A. (2026). ConcreteML: A machine learning software for predicting concrete compressive strength with explainable AI. *SoftwareX*, 34, Article 102729. <https://doi.org/10.1016/j.softx.2026.102729>
- [41]. Zhang, X., Akber, M. Z., & Zheng, W. (2021). Prediction of seven-day compressive strength of field concrete. *Construction and Building Materials*, 305, Article 124604. <https://doi.org/10.1016/j.conbuildmat.2021.124604>
- [42]. DeRousseau, M. A., Laftchiev, E., Kasprzyk, J. R., Rajagopalan, B., & Srubar III, W. V. (2019). A comparison of machine learning methods for predicting the compressive strength of field-placed concrete. *Construction and Building Materials*, 228, Article 116661. <https://doi.org/10.1016/j.conbuildmat.2019.08.042>
- [43]. Young, B. A., Hall, A., Pilon, L., Gupta, P., & Sant, G. (2019). Can the compressive strength of concrete be estimated from knowledge of the mixture proportions?: New insights from statistical analysis and machine learning methods. *Cement and Concrete Research*, 115, 379–388. <https://doi.org/10.1016/j.cemconres.2018.09.006>
- [44]. Yeh, I.-C. (2007). Concrete compressive strength [Data set]. UCI Machine Learning Repository. <https://doi.org/10.24432/C5PK67>
- [45]. Bureau of Indian Standards. (2013). IS 3812 (Part 1):2013, Pulverized fuel ash – Specification, Part 1: For use as pozzolana in cement, cement mortar and concrete (3rd rev.). New Delhi, India: BIS.
- [46]. Bureau of Indian Standards. (2018). IS 16714:2018, Ground granulated blast furnace slag for use in cement, mortar and concrete – Specification. New Delhi, India: BIS.
- [47]. Bureau of Indian Standards. (1999). IS 9103:1999, Concrete admixtures – Specification (1st rev.). New Delhi, India: BIS.
- [48]. Bureau of Indian Standards. (2016). IS 383:2016, Coarse and fine aggregate for concrete – Specification (3rd rev.). New Delhi, India: BIS.
- [49]. Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1), 55–67. <https://doi.org/10.1080/00401706.1970.10488634>
- [50]. Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63(1), 3–42. <https://doi.org/10.1007/s10994-006-6226-1>
- [51]. Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A., & Vapnik, V. (1997). Support vector regression machines. In *Advances in Neural Information Processing Systems 9* (pp. 155–161). Cambridge, MA: MIT Press.
- [52]. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533–536. <https://doi.org/10.1038/323533a0>

- [53]. Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego, CA.
- [54]. Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- [55]. [55]. Strobl, C., Boulesteix, A.-L., Zeileis, A., & Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, 8, Article 25. <https://doi.org/10.1186/1471-2105-8-25>
- [56]. Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30* (pp. 4765–4774).
- [57]. Bureau of Indian Standards. (2000). IS 456:2000, Plain and reinforced concrete – Code of practice (4th rev.). New Delhi, India: BIS.
- [58]. Zviazhynski, B., White, C., Lees, J. M., & Conduit, G. J. (2025). Predicting concrete strength using machine learning and fresh-state measurements. *Data-Centric Engineering*, 6, Article e37. <https://doi.org/10.1017/dce.2025.10018>
- [59]. Swati, Nagar, R., & Gupta, R. (2026). Rapid on-site prediction of concrete compressive strength using integrated machine learning and non-destructive testing. *Asian Journal of Civil Engineering*, 27(10), 4993–5011. <https://doi.org/10.1007/s42107-026-01753-0>