Supplementary information to

Prediction of Concrete Strengths Enabled by Missing Data Imputation and Interpretable Machine Learning

Gideon A. Lyngdoh[1], Mohd Zaki[2], N.M. Anoop Krishnan[2,3], and Sumanta Das[1]

[1]Department of Civil and Environmental Engineering, University of Rhode Island, Kingston, RI, USA 02881

[2]Department of Civil Engineering, Indian Institute of Technology Delhi, Hauz Khas, New Delhi, 110016, India

[3]School of Artificial Intelligence, Indian Institute of Technology Delhi, Hauz Khas, New Delhi, 110016, India

1. PREDICTION OF CONCRETE STRENGTHS USING COMMON MACHINE LEARNING TECHNIQUES

In this section, the prediction of concrete compressive and tensile strengths from common machine learning techniques is illustrated. The experimental compressive strength and tensile strength obtained from [1] serve as a database. While the main paper details XGBoost, this supplementary section details other approaches used in the study which are polynomial regression, LASSO, support vector machine, random forest, and neural network. For all the cases presented here, the dataset [1] imputed by kNN (k = 10) is used.

1.1 Polynomial Regression (PR) and LASSO

Polynomial regression is a generalization of linear regression in which the selected predictors are mapped to a higher dimensional feature space according to the desired polynomial order. Polynomial regression is relatively easier to interpret when the polynomial order is low, indicating a lower-dimensional correlation between the dependent variable(s) and the mapped feature coordinates. In general, polynomial regression with N^th degree can be expressed as:

$$y = \beta_0 + \sum_i^N \beta_i x^i \qquad [1]$$

where $x^i$is the input variable (or predictor variable) and y is the output (or response) variable. The terms β_0 and β_i are the fitting parameters corresponding to each degree i. In matrix form, the formulation can be separated into two phases. First, the vector of predictors is mapped to higher polynomial dimensions, i.e., $\boldsymbol{x}^i, i = 2, 3, \dots, n$. Second, the mapped higher-order polynomial predictors are used to formulate regression problems identical to the linear regression,

$$\boldsymbol{y} = \boldsymbol{\beta X} + \beta_0 \qquad [2]$$

Using the least-square method, the coefficients (β_0 and β) can be estimated by minimizing the error, which is the sum of the squared difference between the true responses with those predicted responses. Hence, the complexity of the PR models highly depends on the choice of the Nth polynomial degree considered. After obtaining β_0 and β, the unknown variable vector can be obtained from the new predictor vectors, $\boldsymbol{X_p}$ as follows:

$$\hat{\boldsymbol{y}} = \boldsymbol{\beta X_p} + \beta_0 \qquad [3]$$

The linear model with polynomial mapped features is selected by comparing the mean squared error (MSE) values with increasing polynomial order of the features. Without losing generality, the polynomial features are not limited to only the crossing terms. Figures S1 (a) and (b) plot the MSE of the concrete compressive strength and tensile strength with increasing polynomial order. As it is observed from Figure S1(a), the MSE for the training set reduces with increasing polynomial order whereas the MSE for the validation set starts increasing significantly. The MSE for a polynomial degree of 1 indicates that, in this domain, the model is underfitted. However, when the model incorporating some polynomial terms strictly larger than 3, the models are overfitted. This arises from the fact that when at a high polynomial degree, the model starts to fit the noise of the training rather than the actual overall trend [2]. A similar observation is observed for concrete tensile strength as shown in Figure S1(b).

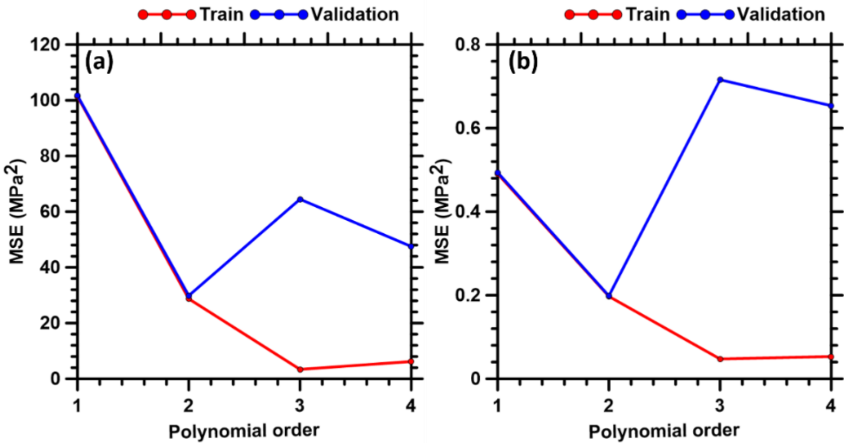

Figure S1. MSE values of (a) the concrete compressive strength, and (b) the concrete tensile strength predictions using the polynomial regression models as a function of the maximum polynomial order for the training set and validation set. The dataset was imputed using kNN (k = 10).

One of the crucial disadvantages of the least-squares approximation is its tendency to overfit the training data. To overcome such a problem, the least absolute shrinkage and selection operator (LASSO) regression offers a useful solution to reduce the complexity of the model, and eventually limit the risk of overfitting [3]. In addition to the cost function used in PR (i.e., the sum of the squared difference between the 'true' and 'predicted' values), LASSO is modified by adding an additional term to penalizes complex models. The modified cost function for LASSO is expressed as:

$$cost\ function = \left\|y - \beta_0 - \sum_i^N \beta_i x^i\right\|_2^2 + \lambda \sum_i^N \|\beta_i\| \quad [4]$$

where λ is a hyperparameter that controls the weight of the penalty associated with the complexity of the model. In practice, LASSO penalizes some of the β_i coefficients to zero in order to minimize the value of the cost function which leads to a decrease in the model complexity. The degree of complexity of LASSO models can be tuned by adjusting the value of λ, where the increasing value of λ yields simpler models.

Figure S2 (a) and (b) plot the mean MSE of the compressive strength and tensile strength for concrete, respectively. In this study, the optimized value of 0.01 is adopted for λ.

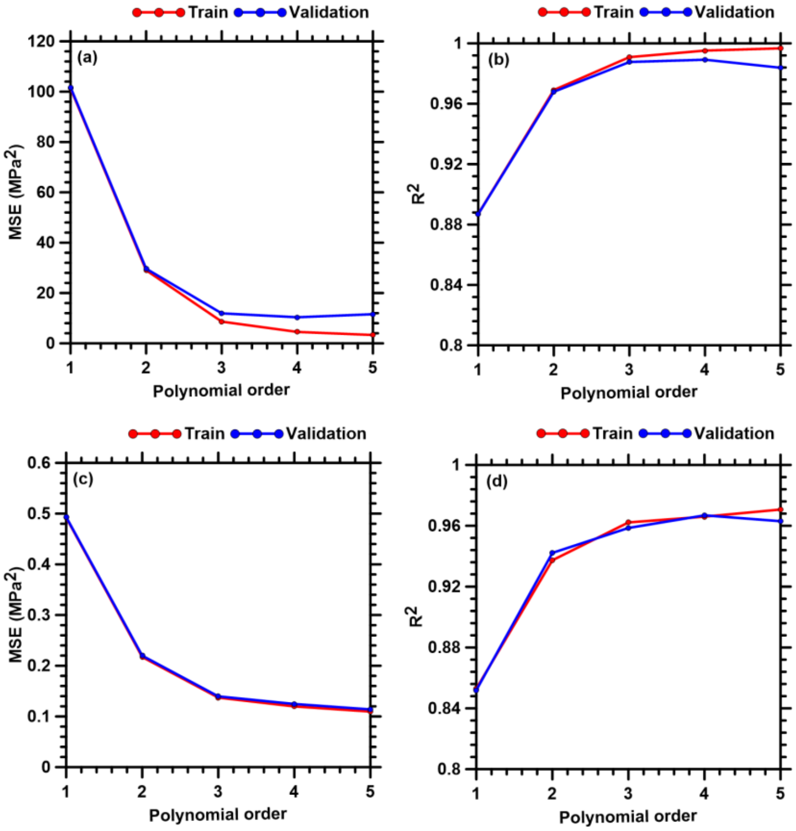

Figure S2. (a) MSE and (b) R^2 values for polynomial regression models with LASSO constraints using the imputed dataset with kNN (k = 10) for compressive strength of concrete. (c) MSE and (d) R^2 values for polynomial regression models with LASSO constraints using the imputed dataset with kNN (k = 10) for tensile strength of concrete. The optimal polynomial order is chosen for such model where minimum MSE and maximum R^2 of the validation set are observed.

As it is observed from Figure S2(a), the MSE for training set reduces with increasing polynomial order whereas the MSE for validation shows little improvement beyond polynomial degree equal to 3. As opposed to the polynomial regression described above (see Figures S1(a) and (b)), the LASSO model reduces the model complexity and prevents the tendency to overfit the data, which is evident from the validation data (see Figures S2(a) and (c)). Figures S2(b) and (d) show the R^2 value for the compressive strength and tensile strength for concrete with increasing polynomial order, respectively. A similar trend was observed for the tensile strength of concrete in which beyond the polynomial order of 3, the accuracy of the model drops as evident from the validation data. Thus, the optimal polynomial degree for LASSO is chosen as 3 with a λ value of 0.01. Figures S3 (a) and (b) demonstrate the comparison of the predicted compressive strength and predicted tensile strength of concrete from the LASSO model with the experimental data. The R^2 value for both compressive strength and tensile strength of concrete outputs are also computed.

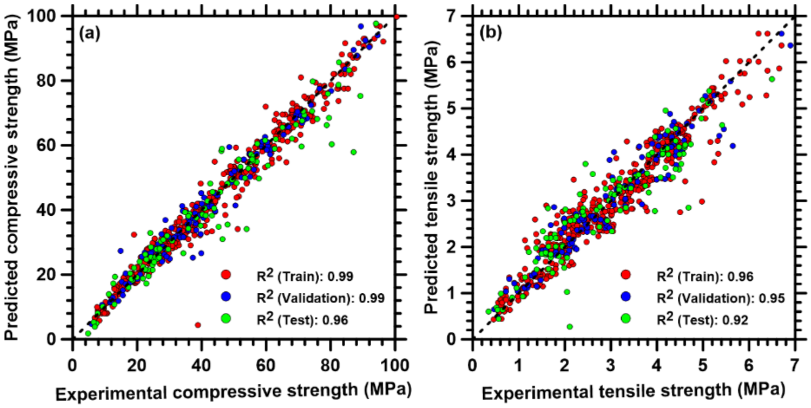

Figure S3. Comparison of (a) the predicted compressive strength and (b) the predicted tensile strength of concrete using the LASSO model (polynomial degree and λ equal to 3 and 0.01, respectively) with the experimental values.

1.2 Support Vector Machine (SVM)

Support vector machine is a support vector classifier that determines the best separating hyperplanes in a higher-dimensional space of the original predictors [4]. The realization of raising the predictors to a space of higher dimension is based on the kernel tricks applied to the predictors. The support vector regression is a convex optimization problem that gives a unique solution to a given set of predictors and responses. The support vector regression can be expressed as follows,

$$\underset{\beta,\beta_0}{argmin}\left\{1/2\left|\left|\beta\right|\right|^2 + C\sum\left|\xi_i\right|\right\} \text{ subject to } \left|y_i - \beta^{\intercal}x_i - \beta_0\right| \leq \epsilon + \left|\xi_i\right| \quad [5]$$

Where ϵ is the pre-defined margin size or the maximum error tolerated by the model, ξ_i is the slack variable that accounts for the tolerance of out-of-margin data points, and C is the constraint of overall tolerance of the out-of-margin cases for finding the SVM model. This constraint acts as a regularization term. As C increases, the regression result is less prone to overfitting the given data. In this study, a radial basis function (rbf) kernel is adopted. Figures S4(a) and (b) show the plot of MSE and R^2 of the SVM model (with rbf kernel) with increasing gamma (γ) value for the compressive strength of concrete. It is observed that with low gamma values, the model is underfitting and the error value is comparatively high for both training and validation sets. As the γ value is increased, MSE values drop and R^2 values increase for both training and validation. The trend continues up to a γ value of 0.01 beyond which the MSE starts increasing and R^2decreases. Thus, the optimum order is chosen for the model where minimum MSE and high R^2 for validation set is achieved which corresponds to a γ value of 0.01. The C value adopted in this model is 100. A similar observation is observed for the tensile strength of concrete (Figures S4(c) and (d)).

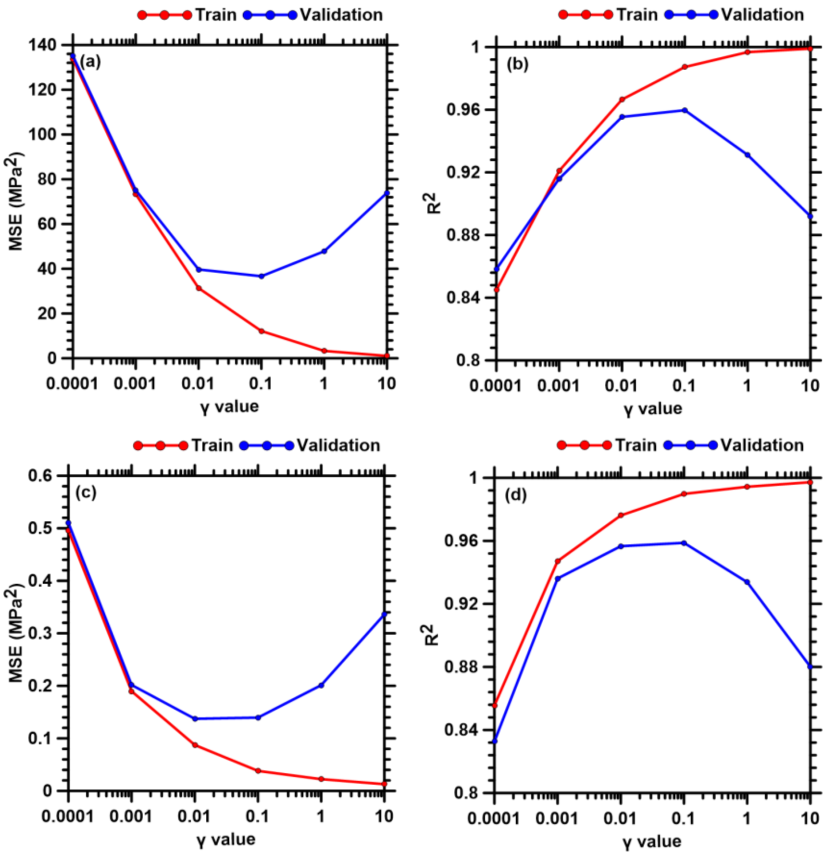

Figure S4. (a) MSE and (b) R^2 values for support vector machine models (with rbf kernel) using the imputed dataset with kNN (k = 10) for compressive strength of concrete. (c) MSE and (d) R^2 values for for support vector machine models (with rbf kernel) using the imputed dataset with kNN (k = 10) for tensile strength of concrete. The optimal gamma value is chosen for such model where minimum MSE and maximum R^2 of the validation set are observed.

Using the optimal gamma value of 0.01, the predicted results are plotted against experimental values in Figures S5 (a) and (b) for compressive strength and tensile strength of concrete. The R^2 value for both compressive strength and tensile strength of concrete outputs are also computed.

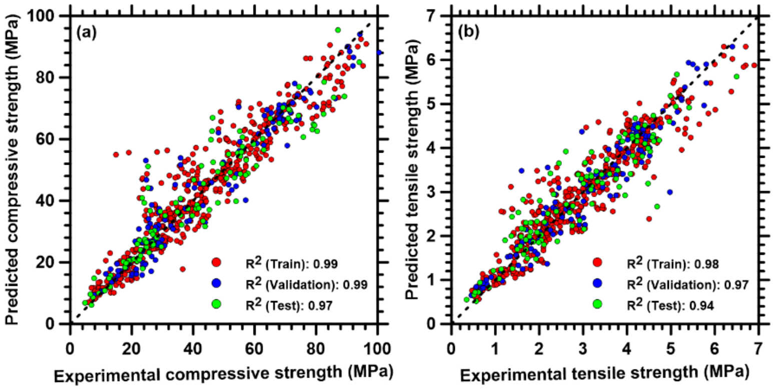

Figure S5. Comparison of (a) the predicted compressive strength and (b) the predicted tensile strength of concrete using SVM (gamma = 0.01) model with the experimental values.

1.5 Random Forest (RF)

Random forest is a decision forest or decision tree method that belongs to ensemble learning. It is an average of a large collection of decorrelated decision trees. Such an ensemble method can both increase the prediction accuracy and reduce over-fitting problems [5]. In this model, a large number of trees is trained individually using only a subset of the input variables [5,6]. In each tree, a bootstrap sample of the training data is used instead of the entire set of training data. This procedure is known as bootstrap aggregation or bagging [5]. The predictions of each individual are then averaged to obtain the prediction of the random forest ensemble. This method is similar to boosting in many aspects but can be easily trained and manipulated. Figures S6(a) and (b) show the plot of MSE and R^2 values respectively offered by random forest algorithm with an increase in the number of trees for the compressive strength of concrete. Here, the number of trees characterizes the complexity of the model. As observed from the Figure S6, minimum MSE for validation set is observed 30 trees and no significant change is observed for R^2. It is noticed that the MSE of the training set and validation set remains almost constant upon increasing number of trees and this indicates that the RF does not yield any noticeable overfitting at high model complexity. Similar justification applies to the tensile strength results shown in Figures S6(c) and (d). To assess the accuracy of the models, Figures S7 shows the predicted values obtained from the best RF model with the number of trees equal to 30 against experimental values.

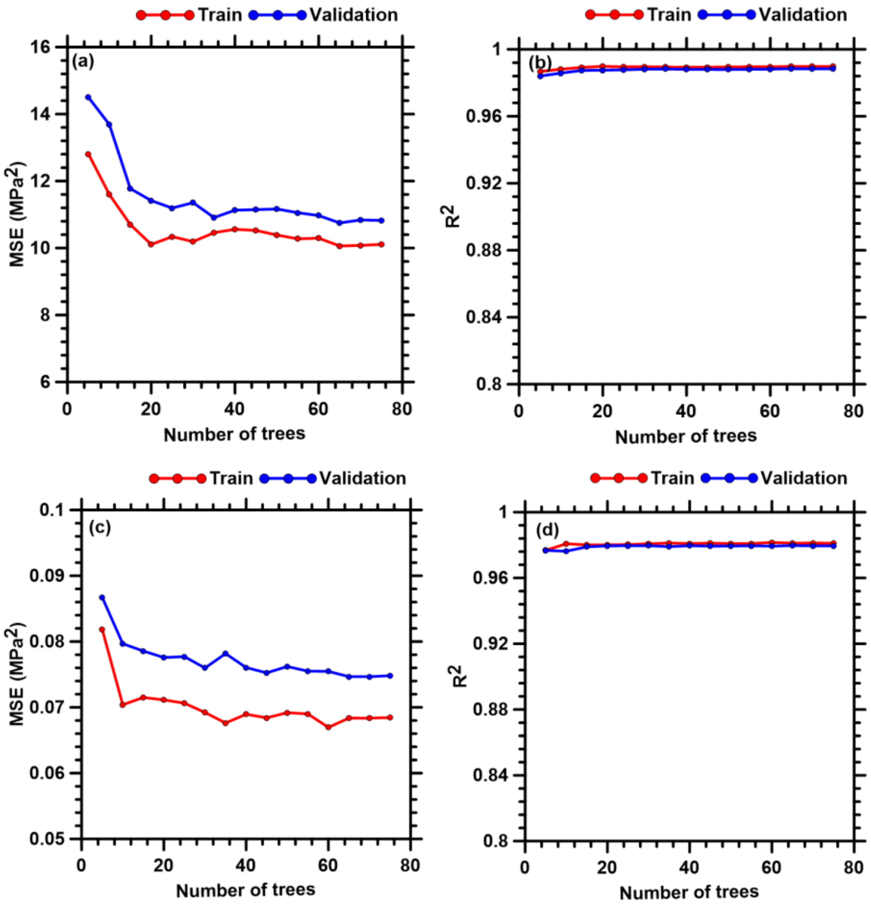

Figure S6. (a) MSE and (b) R^2 values for Random Forest models using the imputed dataset with kNN (k = 10) for compressive strength of concrete. (c) MSE and (d) R^2 values for Random Forest models using the imputed dataset with kNN (k = 10) for tensile strength of concrete. The optimal number of trees is chosen for such model where minimum MSE and maximum R^2 of the validation set are observed.

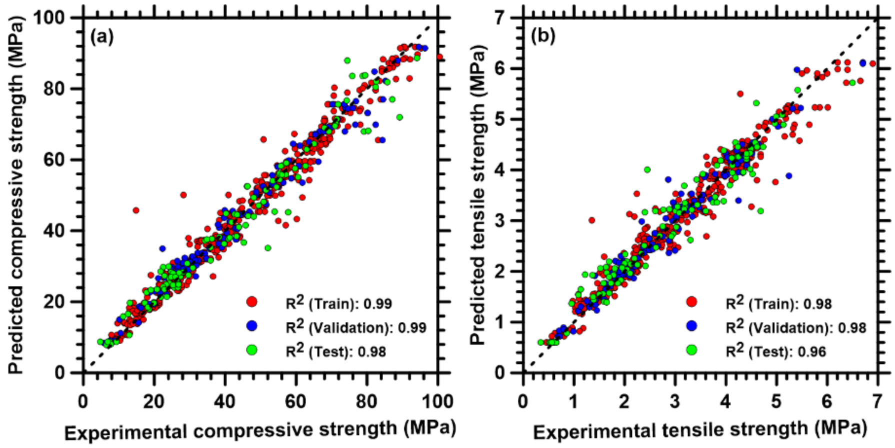

Figure S7. Comparison of (a) the predicted compressive strength and (b) the predicted tensile strength of concrete by random forest (with the number of trees equals to 30) with the experimental values.

1.6 Multilayer perceptron-based Neural Network (NN)

Neural networks (NN) draw inspiration from a human neural system where the information is stored and processed in the brain [7]. The map between the predictor and the response comprises multiple perceptron layers (such as the input layer, hidden layers, and output layer) and activation functions. It is called the feed-forward neural network [8,9]. It is a mathematical model which maps a given set of predictors, x, to a set of the desired response, y and is expressed as follows,

$$\boldsymbol{y} = \boldsymbol{f_N}\left(\boldsymbol{A_N}, \dots \boldsymbol{f_2}\left(\boldsymbol{A_2}, \boldsymbol{f_1}(\boldsymbol{A_1}, \boldsymbol{x})\right) \dots\right) \quad [6]$$

where $\boldsymbol{f_N}(\cdot)\colon \mathbb{R} \to \mathbb{R}$ is a continuous bounded function which is usually referred to as the activation function, $\boldsymbol{A_i}\colon \mathbb{R}^{\boldsymbol{d_i}} \to \mathbb{R}^{\boldsymbol{d_{i+1}}}$ is the transformation matrix that contains weights between two layers of perceptrons [8]. The NN has gained tremendous attention in academia and applications in engineering due to the proven universal approximation property that states that the feed-forward NN architectures with a sigmoid activation function can approximate any set of functions between two Euclidean spaces for the canonical topology [10].

In the NN, the neurons in the input layer do not participate in the computation but serve to pass the input vectors to the hidden layers. Each neuron in the layers other than the input layer performs a simple non-linear transformation using an activation function such as rectified linear units (ReLU) or sigmoid function [11]. The neurons in two consecutive layers are connected by weights, which are learned through a training process to approximate the mapping function from the input to output vectors. The weights can be solved by formulating the above mapping into a constrained optimization problem as stated below,

$$\boldsymbol{argmin}_{A_j}\left\{f_N\left(\boldsymbol{A_N}, \dots \boldsymbol{f_2}\left(\boldsymbol{A_2}, \left(\boldsymbol{f_1}(\boldsymbol{A_1}, \boldsymbol{x})\right) \dots\right) + \boldsymbol{\lambda}\mathbf{g}\left(\mathbf{A_j}\right)\right)\right\} \quad [7]$$

where λ is the regularization intensity constant and $g(\cdot)$ is a functional form of the weights to be regularized. This optimization problem is usually solved by stochastic gradient descent or backward propagation algorithm. However, because of the non-convex nature of the neural network, the solution to this optimization problem is not unique. Moreover, the selection of the number of layers and the number of perceptrons in each layer affects the result of the regression, and high variances are observed when large numbers of neurons and layers are used which necessitates an efficient regularization approach. In this study, while training an NN model, a rectified linear unit (ReLU) is implemented as an activation function due to its superior performance [12].

For the NN model, the hyperparameters include the number of hidden nodes, size of hidden layers, optimizer function, learning rate, epoch, and batch size. In this study, the NN model is trained using the back-propagation algorithm [13]. Also, the NN model implements Adam optimizer [14] with a learning rate of 0.001, an epoch equals 400, and a batch size of 32. The size of hidden layers is set to two as no improvement is observed with any further increase in the number of hidden layers. Figures S8(a) and (b) represent the MSE and R^2 values respectively for concrete compressive strength whereas Figures S8(c) and (d) show the same for the concrete tensile strength from the NN model with respect to an increasing number of neurons.

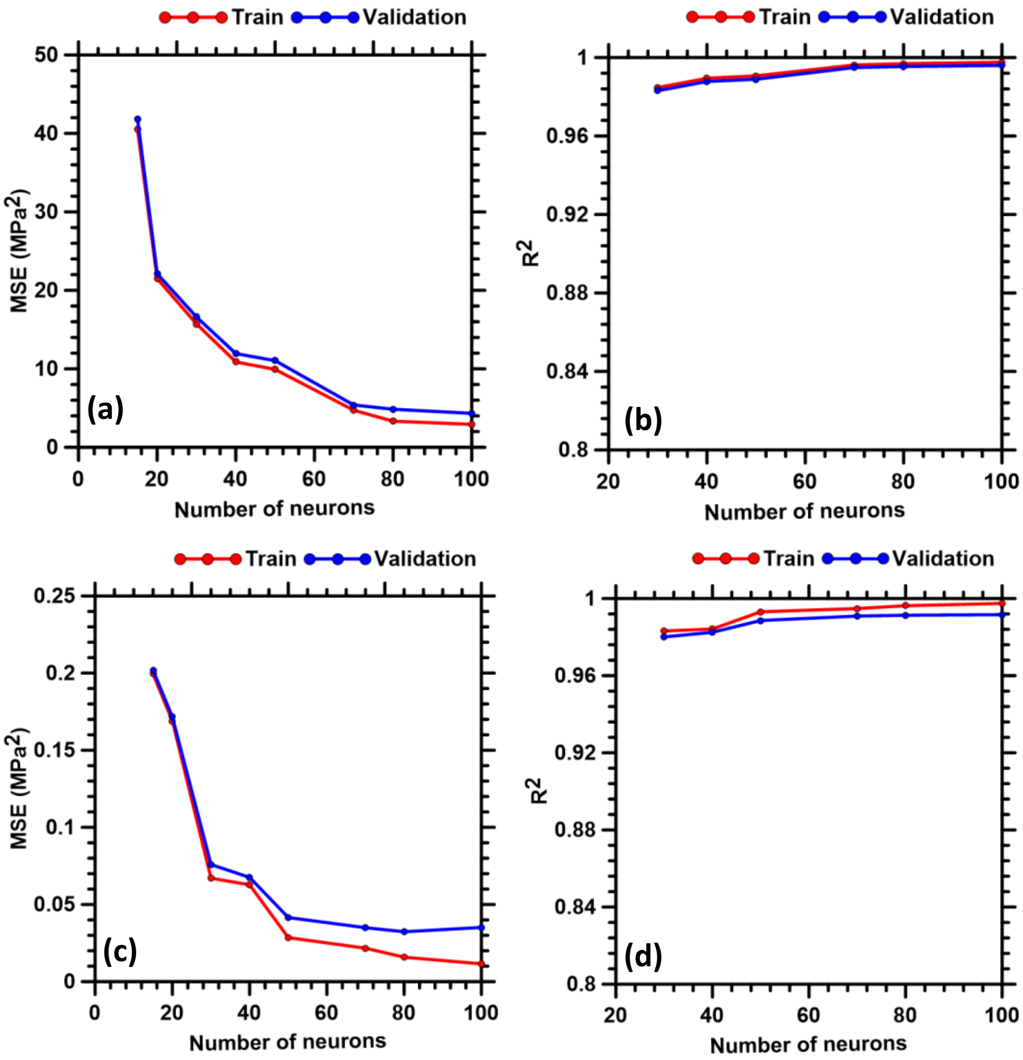

Figure S8. (a) MSE and (b) R^2 values for NN with 2 hidden layers using the dataset imputed with kNN (k = 10) for compressive strength of concrete. (c) MSE and (d) R^2 values for NN with 2 hidden layers using the dataset imputed with kNN (k = 10) for tensile strength of concrete. From these relationships, an optimized number of neurons with minimum MSE and maximum R^2 values are obtained.

It is observed that with a low number of neurons, the model is yet to learn, which is demonstrated with a high MSE value. In contrast to a low number of neurons, the model with a high number of neurons has saturated where the MSE value for the validation set either starts increasing or shows a slight improvement. The optimum neurons are chosen where minimum MSE value and high R^2 value for the validation set is achieved which is at 70 neurons for compressive strength prediction and 50 neurons for tensile strength prediction.

In this section, predictions based on the hyperparameter-optimized NN model are reported for the k-NN (k=10)-imputed data. The test set, which is hidden from the model during the training, is used to evaluate the model prediction using the trained NN model with the optimized hyperparameters. Figure S9(a) shows the predicted compressive strength of the concrete using the trained NN model with R^2 values from the training set, validation set, and test set.

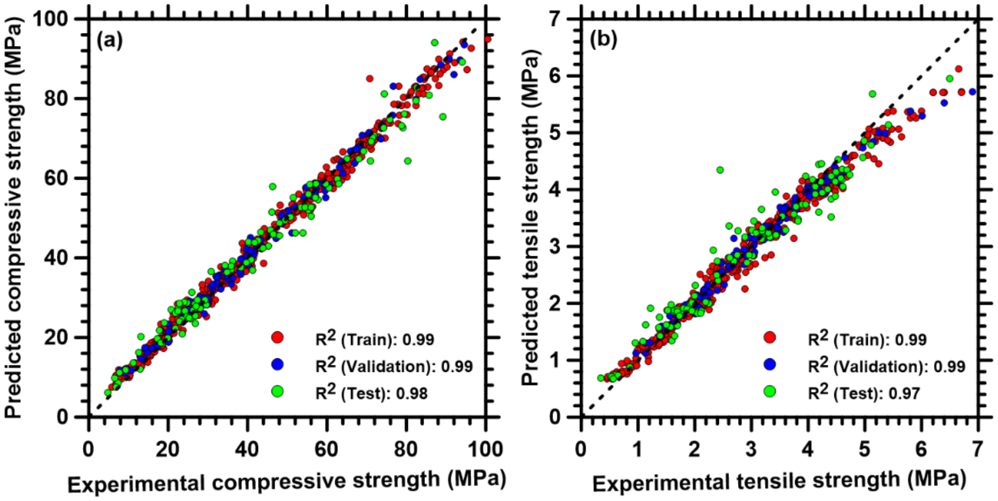

Figure S9. Comparison of (a) the predicted compressive strength and (b) the predicted tensile strength of concrete using the NN model with the experimental values.

Overall, excellent prediction efficacy is obtained as indicated by the R^2 values. Figure S9(b) represents the predicted tensile strength of the concrete using the trained NN model. At lower tensile strength values, the NN model shows excellent predictions. However, the prediction efficacy of the NN model reduces at higher tensile strengths which could be attributed to a lower number of available high tensile strength data points for training. Nevertheless, an overall high R^2 value of 0.97 is obtained for the test set.

2. SIGNIFICANCE OF SHAP RIVER FLOW PLOT

To explain the significance of SHAP river flow plot (as shown in Figure 8 in the main article), we have taken couple of data points. The first data point corresponds to a low output compressive strength of 9.49 MPa which is obtained for a specific combination of input feature values (stone powder content = 0.06, sand ratio = 0.36, fineness modulus = 2.92, water content = 81, binder tensile strength = 8.4 MPa, binder compressive strength 62.4 MPa, crushed stone size = 80, water-binder ratio = 0.5, curing age = 7 days and water-cement ratio = 1.0). The second data point corresponds to a higher output compressive strength value of 84.36 MPa which is also obtained for a specific combination of input features (stone powder content = 0.04, sand ratio = 0.41, fineness modulus = 3.0, water content = 150, binder tensile strength = 8.0 MPa, binder compressive strength 47.84 MPa, crushed stone size = 30, water-binder ratio = 0.25, curing age = 90 days and water-cement ratio = 0.33).

Now, for the first case, if we only consider water-cement ratio as the input feature and eliminate all other input features, the mean value of compressive strength is 29.58 MPa corresponding to a water-cement ratio of 1.0. Similarly, if curing age is considered as the only input feature, a mean compressive strength of 33.31 MPa is obtained corresponding to curing age of 7 days. Using this approach, all the mean values corresponding to each input features are obtained. The values are reported in Figure S10 (a) and this line corresponds to blue color as the corresponding output value is lower. Similarly, for the second case (output compressive strength of 84.36 MPa), the mean compressive strength of individual input features are shown in Figure S10 (b). This line corresponds to blue color as the corresponding output compressive strength value is on the higher side.

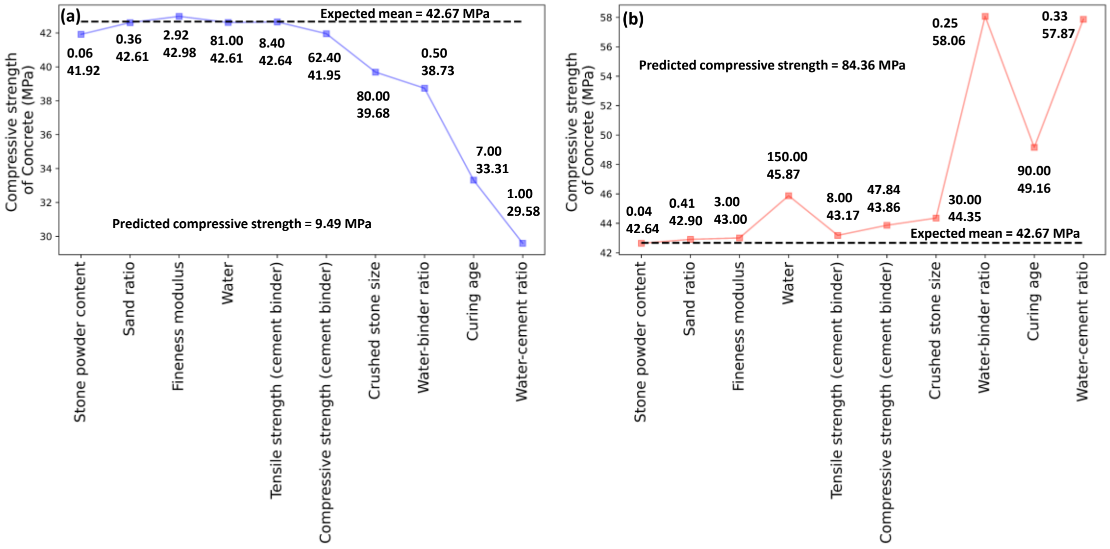

Figure S10. SHAP river flow plot for concrete with (a) low compressive strength and (b) high compressive strength using the XGBoost model.

Table S1. Hyperparameters values corresponding to different ML approaches

Models	**Hyperparameters**	**Range/List/Values**
Lasso	λ	0.01
	degree	1 to 4
SVM	γ	0.001,0.001,0.01,0.1,1,10
	C	10,100,1000
RF	num_estimator	1 to 80
	min_samples_leaf	5
NN	learning rate	0.001
	epoch	400
	batch_size	32
	hidden_neurons	2 to 100
	hidden_layer	1 to 5
XGBoost	num_estimators (number of trees)	5 to 500
	max_depth	9
	minimum child weight	7
	learning rate	0.09
	objective	reg:squarederror

3. INFLUENCE OF INSIGNIFICANT FEATURE REMOVAL ON THE COMPRESSIVE STRENGTH PREDICTIVE EFFICACY

To elucidate and quantify the influence of insignificant features on the predictive efficacy, Table S2 shows the comparison for the compressive strength of concrete using XGBoost when (i) full features are used, (ii) only stone powder content is removed, and (iii) stone powder content and sand ratio are removed.

Table S2: Performance measures for analysis of different input variable combinations

Combinations	Train MSE (MPa^2)	Validation MSE (MPa^2)
No features removed	1.98	5.54
Stone powder content removed	6.37	7.97
Stone powder content and sand ratio removed	2.92	6.15

REFERENCES

[1] S. Zhao, F. Hu, X. Ding, M. Zhao, C. Li, S. Pei, Dataset of tensile strength development of concrete with manufactured sand, Data in Brief. 11 (2017) 469–472. https://doi.org/10.1016/j.dib.2017.02.043.

[2] K. Yang, X. Xu, B. Yang, B. Cook, H. Ramos, N.M.A. Krishnan, M.M. Smedskjaer, C. Hoover, M. Bauchy, Predicting the Young's Modulus of Silicate Glasses using High-Throughput Molecular Dynamics Simulations and Machine Learning, Sci Rep. 9 (2019) 8739.

[3] R. Tibshirani, Regression Shrinkage and Selection Via the Lasso, Journal of the Royal Statistical Society: Series B (Methodological). 58 (1996) 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

[4] V.N. Vapnik, Statistical learning theory, Wiley, New York, USA, 1998.

[5] Z.Q.J. Lu, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Journal of the Royal Statistical Society: Series A (Statistics in Society). 173 (2010) 693–694. https://doi.org/10.1111/j.1467-985X.2010.00646_6.x.

[6] M. Kuhn, K. Johnson, Applied Predictive Modeling, Springer Science & Business Media, 2013.

[7] F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain, Psychological Review. 65 (1958) 386–408. https://doi.org/10.1037/h0042519.

[8] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, The MIT Press, Cambridge, Massachusetts, 2016.

[9] N. Siddique, H. Adeli, Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing, John Wiley & Sons, 2013.

[10] A. Kratsios, Characterizing the Universal Approximation Property, ArXiv:1910.03344 [Cs, Math, Stat]. (2020). http://arxiv.org/abs/1910.03344 (accessed August 30, 2020).

[11] M.W. Gardner, S.R. Dorling, Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences, Atmospheric Environment. 32 (1998) 2627–2636. https://doi.org/10.1016/S1352-2310(97)00447-0.

[12] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM. 60 (2017) 84–90. https://doi.org/10.1145/3065386.

[13] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature. 323 (1986) 533–536. https://doi.org/10.1038/323533a0.

[14] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, ArXiv:1412.6980 [Cs]. (2017). http://arxiv.org/abs/1412.6980 (accessed May 10, 2021).