2022

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

Gideon A. Lyngdoh  
*University of Rhode Island*

Mohd Zaki

N.M. Anoop Krishnan

Sumanta Das  
*University of Rhode Island, sumanta_das@uri.edu*

Follow this and additional works at: https://digitalcommons.uri.edu/cve_facpubs

The University of Rhode Island Faculty have made this article openly available.  
Please let us know how Open Access to this research benefits you.

Terms of Use
This article is made available under the terms and conditions applicable towards Open Access Policy Articles, as set forth in our Terms of Use.

Citation/Publisher Attribution

Available at: https://doi.org/10.1016/j.cemconcomp.2022.104414

This Article is brought to you for free and open access by the Civil & Environmental Engineering at DigitalCommons@URI. It has been accepted for inclusion in Civil & Environmental Engineering Faculty Publications by an authorized administrator of DigitalCommons@URI. For more information, please contact digitalcommons-group@uri.edu.
Prediction of Concrete Strengths Enabled by Missing Data Imputation and Interpretable Machine Learning

The University of Rhode Island Faculty have made this article openly available. Please let us know how Open Access to this research benefits you.

This is a pre-publication author manuscript of the final, published article.

Terms of Use
This article is made available under the terms and conditions applicable towards Open Access Policy Articles, as set forth in our Terms of Use.

This article is available at DigitalCommons@URI: https://digitalcommons.uri.edu/cve_facpubs/402
Prediction of Concrete Strengths Enabled by Missing Data Imputation and Interpretable Machine Learning

Gideon A. Lyngdoh¹, Mohd Zaki², N.M. Anoop Krishnan²,³, and Sumanta Das¹,*

¹Department of Civil and Environmental Engineering, University of Rhode Island, Kingston, RI, USA 02881
²Department of Civil Engineering, Indian Institute of Technology Delhi, Hauz Khas, New Delhi, 110016, India
³School of Artificial Intelligence, Indian Institute of Technology Delhi, Hauz Khas, New Delhi, 110016, India

*Corresponding author: S. Das (sumanta_das@uri.edu)

Abstract

Machine learning (ML)-based prediction of non-linear composition-strength relationship in concretes requires a large, complete, and consistent dataset. However, the availability of such datasets is limited as the datasets often suffer from incompleteness because of missing data corresponding to different input features, which makes the development of robust ML-based predictive models challenging. Besides, as the degree of complexity in these ML models increases, the interpretation of the results becomes challenging. These interpretations of results are critical towards the development of efficient materials design strategies for enhanced materials performance. To address these challenges, this paper implements different data imputation approaches for enhanced dataset completeness. The imputed dataset is leveraged to predict the compressive and tensile strength of concrete using various hyperparameter-optimized ML approaches. Among all the approaches, Extreme Gradient Boosted Decision Trees (XGBoost) showed the highest prediction efficacy when the dataset is imputed using k-nearest neighbors (kNN) with a 10-neighbor configuration. To interpret the predicted results, SHapley Additive exPlanations (SHAP) is employed. Overall, by implementing efficient combinations of data imputation approach, machine learning, and data interpretation, this paper develops an efficient approach to evaluate the composition-strength relationship in concrete. This work, in turn, can be used as a starting point toward the design and development of various performance-enhanced and sustainable concretes.

Keywords: Machine learning, concrete strength, missing data, Data Imputation, SHAP
1. INTRODUCTION

Concrete is considered the most widely used construction material in the world. The mechanical performance of concrete is primarily characterized in the industry by its 28-days compressive strength. Besides, with the growing focus on fundamental modifications in concrete towards reducing the environmental footprint, the evaluation of flexural and tensile strengths [1] has also become increasingly important. In addition, evaluating strengths has become more critical considering the emergence of concretes with supplementary cementitious materials or alternative binders for multifunctional applications and improved sustainability credentials. Accurate prediction of the concrete strengths significantly impacts the efficiency of the material usage and structural safety in civil infrastructure [2]. Besides, underestimating concrete strengths can lead to excess cement usage, associated with a significant increase in CO₂ emissions [3].

Over the past few decades, enormous efforts have been made to develop predictive models that map the concrete mixing proportion to its associated strength [4–18]. Ideally, a predictive model should provide important insights that lead to improved concrete mixtures with excellent constructability and durability at a lower cost [19–22]. This has led to models driven by physics or chemistry-based relationships [23–25]. While these conventional approaches have played significant roles in extensively linking the parameters such as cement dosage, aggregate fraction, and air void content with the concrete strength, evaluation of the combined effects of these features is still a challenging task. In addition, these conventional approaches ignore the influence of secondary factors such as the nature and dosage of chemical admixtures, aggregate size distribution, and fineness modulus of aggregates [26]. Therefore, it is challenging to seek out a robust and universal concrete strength prediction model using conventional approaches [27].

Therefore, implementing data-driven approaches to establish the underlying relationships between the inputs and the outputs have been of great interest in recent years. These data-driven models are constructed by employing machine learning (ML) techniques to predict the output based on the features learned from the data. Recently, many ML models have been established to predict concrete strength [5,6,15,28–30]. Artificial Neural Network (ANN)-based ML model to predict the compressive strength of High-Performance concrete was initially demonstrated by Yeh [30] in 1998. It was shown that the ANN model provides better prediction as compared to a model based on regression analysis. Since then, many ML models such as decision trees (DT), random forest (RF), support vector machine (SVM), k-nearest neighbor (k-NN), and other hybrid ML models have been proposed. While training machine learning
models requires huge sets of data, recent studies [6,15] have demonstrated that increasing the dataset will only lower the variation of the predicted value once the model has already mapped the relationship between the inputs and the output. In these studies, ML models such as ANN, SVM, and DT were trained to predict 28-day compressive strength using different mixtures variables for a large data set (>10000 observations) obtained from job-site mixtures. The value of their model was also illustrated in terms of yielding optimal mixes to reduce cost and embodied CO$_2$ impact without hampering the design strength. A boosted tree ML model has also been employed to evaluate the nondestructive concrete compressive strength [7]. It was shown that a good prediction for compressive strength could be achieved using the information from ultrasonic speed and rebound number. Variable performance plots from boosted tree learning were also evaluated. Prediction of compressive strength using gene expression programming (GEP) has also been exploited [8] and compared with other known ML models such as RF, DT, and ANN. It was found that RF provides accurate results compared to other models due to its bagging mechanism. A comparison for model selection methods has also been studied for predicting the compressive strength of high-performance concrete using ANNs [9]. It was shown that the Bayesian approach using Markov Chain Monte Carlo (MCMC) sampling approximation of learning and prediction provides the best accuracy. Moreover, Cook et al. [5] proposed a novel hybrid ML model based on a random forest (RF) model with a firefly algorithm (FFA) to develop a correlation between the mixture design variables of concrete and the age-dependent compressive strength. Their results indicated that the hybrid RF-FFA model outperformed the standalone ML models such as SVM, multilayer perceptron artificial neural network (MLP-ANN), M5Prime model tree algorithm (M5P), and RF. Overall, the ML-based prediction has been shown to outperform the conventional approaches, especially while dealing with highly non-linear problems [30,31]. While all these previous models have been trained using a complete dataset, i.e., no missing values, in reality, most of the data are contaminated with outliers, noise, or missing values. This is a significant challenge for concrete applications, as the availability of large consistent datasets is limited. Therefore, there is a need to assess the reliability of ML approaches for concrete strength prediction applications.

Additionally, the conventional ML models suffer from a lack of interpretability due to their complex nature. The models are treated as black-box, thereby providing little insight into the actual relationship between the input and output features, which is particularly disconcerting for domain experts. Further, many of the input features may have some interdependence in controlling a specific property. The nature of such interdependence also cannot be investigated using the conventional ML methods. With regard to ML-based concrete strength predictions, attempts have been made to incorporate feature importance
values for interpretable insights. In the study by Zhang et al. [32], variable importance is computed using the random forest (RF) algorithm with data imputed by kNN-10. In other studies by Nguyen et al. [33], feature importance for concrete tensile strength generated by XGBoost is implemented. These studies have evaluated feature importance based on direct assessment of the data or the decrease in model performance. However, no attempts have been made to integrate any data interpretation algorithms with ML-based models. Thus, there is a need to efficiently integrate a robust data interpretation algorithm with the ML-based concrete strength prediction models toward developing a synergistic interpretable ML-based predictive tool.

This study aims to address both the concerns mentioned above related to the issue of missing data in the available datasets and the lack of interpretability of the ML approaches toward efficient prediction of compressive and tensile strength of concrete for any given set of performance descriptors. First, various efficient data imputation approaches are implemented to handle missing data that can pose a severe problem if not taken care of while developing a robust predictive model. Moreover, various ML techniques such as polynomial regression (PR) [43], least absolute shrinkage and selection operator (LASSO) [44], support vector machine (SVM) [45], random forest (RF) [46], XGBoost [47] and neural network (NN) [48] are employed to evaluate their performance on imputed data. Second, to address the black-box nature of the traditional ML approaches, recent works on interpretable machine learning have been successful in deciphering complex ML models. Specifically, SHapley Additive exPlanations (SHAP) [38,39] is a model-agnostic post-hoc approach that provides insights into the ML models. SHAP has been successfully used in various fields such as material science to interpret the composition–property relationships [37–39], synthesis of advanced inorganic materials [34], corrosion rate prediction of low-alloy steels [35], the behavior of nanophotonic structures [36], biomedical engineering [37–39], finances [40], and text classification [41]. However, the implementation of interpretable machine learning using SHAP [42] has not been widely realized so far in the concrete community. This study integrates SHAP with ML algorithms to provide critical insights into the underlying complex non-linear behavior of the concrete strength design mixture and to explain the contribution of the input features by assigning an importance value for each input variable. Hence, while data imputation approaches alleviate the issues related to missing data and noise or outliers in the datasets, SHAP provides valuable interpretability features to the ML approaches. Overall, through missing data imputation and implementation of interpretable ML approaches using SHAP, this study establishes a valuable composition-property link that would aid performance tuning and the development of sustainable and durable concretes.
2. DATASET DESCRIPTION

The dataset used in this study comprises 754 uniaxial compressive strength and splitting tensile strength data for manufactured sand concrete (MSC). The data points are assembled from 41 experimental studies [43–45]. The mixture of MSC consists of Grade 42.5 ordinary silicate cement, which is mixed with supplementary cementitious materials (SCMs), crushed stone, and manufactured sand. The SCMs include fly ash, slag, and silica fume. For this particular dataset, the 28-day compressive strength of hardened cement paste ranges from 35.5 to 63.4 MPa, and the 28-day tensile strength ranges from 6.9 to 10.8 MPa. The maximum size of the crushed stone was varied from 12 mm to 120 mm, and the fineness modulus of manufactured sand ranged from 2.2 to 3.55. The mass-based water-binder ratio ranged between 0.24-1.00, while a range of 0.3-1.43 was adopted for the water-cement ratio. The sand ratio considered was 25-54%, and the curing time varied from 3 days to 388 days. Here, the sand ratio is defined as the ratio of the weight of manufactured sand to the total weight of aggregates, including sand and coarse aggregates [44]. While the 28-day uniaxial compressive strength of concrete ranged from 10.1 - 96.3 MPa, the 28-day splitting tensile strength ranged from 0.6 to 6.9 MPa. More information on the mixture proportions and experimental procedures are adequately detailed in [43–45]. It is to be noted that in the experimental dataset, the manufactured sand (MS) is used instead of the river sand or natural sand. MS is produced by crushing the rock where a certain amount of stone powder is also produced depending on factors, such as the lithology of parent rock and crushing craft [46–49]. Recent studies have demonstrated the benefits and effects of using MS as a river (or natural) sand replacement in mechanical properties [45,50–52] and durability [53–55] of the concrete. Table 1 shows the statistical results of the input and output variables and the missing value proportions.

Table 1: Statistical results of the input/output variables and the missing value proportions in the dataset

<table>
<thead>
<tr>
<th>Variables</th>
<th>Min</th>
<th>Max</th>
<th>Mean</th>
<th>SD</th>
<th>Missing value (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Inputs</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compressive strength of cement binder (MPa)</td>
<td>35.50</td>
<td>63.40</td>
<td>48.32</td>
<td>4.32</td>
<td>35.20</td>
</tr>
<tr>
<td>Tensile strength of cement binder (MPa)</td>
<td>6.90</td>
<td>10.20</td>
<td>8.27</td>
<td>0.58</td>
<td>38.57</td>
</tr>
<tr>
<td>Curing time (days)</td>
<td>3.00</td>
<td>388.00</td>
<td>75.81</td>
<td>98.39</td>
<td>0.00</td>
</tr>
<tr>
<td>Crushed stone size (mm)</td>
<td>16.00</td>
<td>80.00</td>
<td>30.74</td>
<td>11.62</td>
<td>8.70</td>
</tr>
<tr>
<td>Stone powder content</td>
<td>0.00</td>
<td>0.20</td>
<td>0.08</td>
<td>0.05</td>
<td>5.61</td>
</tr>
<tr>
<td>Input</td>
<td>2.20</td>
<td>3.50</td>
<td>3.04</td>
<td>0.27</td>
<td>16.13</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>-------</td>
</tr>
<tr>
<td>Fineness modulus (GPa)</td>
<td>2.20</td>
<td>3.50</td>
<td>3.04</td>
<td>0.27</td>
<td>16.13</td>
</tr>
<tr>
<td>Water-binder ratio</td>
<td>0.25</td>
<td>0.69</td>
<td>0.43</td>
<td>0.10</td>
<td>0.56</td>
</tr>
<tr>
<td>Water-cement ratio</td>
<td>0.31</td>
<td>1.01</td>
<td>0.47</td>
<td>0.08</td>
<td>1.40</td>
</tr>
<tr>
<td>Water (kg/m³)</td>
<td>104.00</td>
<td>291.00</td>
<td>172.68</td>
<td>20.12</td>
<td>1.82</td>
</tr>
<tr>
<td>Sand ratio</td>
<td>0.28</td>
<td>0.45</td>
<td>0.37</td>
<td>0.04</td>
<td>14.87</td>
</tr>
<tr>
<td>Outputs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compressive strength of concrete (MPa)</td>
<td>14.90</td>
<td>96.30</td>
<td>53.79</td>
<td>17.43</td>
<td>0.00</td>
</tr>
<tr>
<td>Tensile strength of concrete (MPa)</td>
<td>1.13</td>
<td>6.40</td>
<td>3.60</td>
<td>1.04</td>
<td>0.00</td>
</tr>
</tbody>
</table>

3. METHODOLOGY

The general pipeline of the ML analysis adopted in this study is summarized in Figure 1. To train ML models, an adequate dataset should be provided. In this study, the experimental dataset is collected from the existing literatures [44]. Since the models are trained on the data, the quality of the data determines the model accuracy. It is observed that the present dataset has missing data. Therefore, it is necessary to bridge the gap of the missing data. Here, data imputation techniques are implemented to handle the missing values. The best method with the least error is selected and used for training the models. After training the model, the efficacy of the ML-based predictive model is evaluated by applying it toward the unseen test dataset. Moreover, to interpret the predicted strength responses and to alleviate the black-box nature of the ML-based algorithms, the ML approaches are efficiently integrated with a SHAP-based model interpretation approach for verification of the fundamental physics-based realistic trends.

Figure 1: A schematic overview of the general pipeline for the ML analysis to predict the strength of concrete
3.1 Missing data imputation

To build a robust ML model, the data provided should be free from missing values, duplicate entries, and outliers. However, in reality, the data are often embedded with uncertainties due to how the data were procured. One such prevalent case is the missing data. Training of the model on a dataset that has multiples missing values can significantly impact the quality of prediction. Toward this, the imputation of missing values can be adopted as an efficient alternative approach. In this study, the various data imputation methods are adopted to handle the missing values. These include removing the missing values, imputation using mean/median values, imputation using k-Nearest Neighbors (k-NN), and multivariate imputation by chained equations (MICE). These techniques are compared and contrasted to shed light on their relative efficiency in missing data management and overall performance prediction. The adopted techniques are detailed in the forthcoming sub-sections:

3.1.1 Removing the missing values

In this method, the missing values are deleted from the dataset. In general, this method is adopted only when the proportion of missing values is significantly small (<5%) [56]. In this study, the method of removing the missing value serves as a benchmark to compare the performance of other imputation methods.

3.1.2 Imputation using mean/median

In this method, the missing value for each feature is replaced by the mean or median of the non-missing values of the respective features. Such a method is easy to implement and works well with small numerical datasets. However, this method does not consider the correlation between features and does not account for the uncertainty in the imputations [57].

3.1.3 Imputation using k-NN

The k-NN is an algorithm that uses the similarity in the features to predict the values of any new data points. In other words, the new point is assigned based on how closely it resembles the points in the training set. In k-NN based method [58], the missing value is imputed by taking the weighted average of the feature associated with the missing value from the k-closest sets. For example, for the missing value in the water-cement ratio in experiment 1, the k-NN method would first find the nearest Euclidean distance between experiment 1 and the other experiments ranging from 2 to N (where N is the total number of experiments). The set that has the minimum nearest Euclidean distance indicates the highest
similarity index. The missing value in the water-cement ratio is then estimated by taking the weighted average of values of water-cement ratio with k-closet sets that have a high similarity index from experiments 2 to N (1 ≤ k ≤ N − 1). In the weighted average, the contribution of each feature is weighted by evaluating the similarity of its expression to that of the water-cement ratio. In this study, the feature similarity or the nearest distance is computed by using Euclidean distance. The drawback of k-NN is that it is sensitive to an outlier in the data. For more information on the methodology, the readers can refer to [58]. The values of k considered in this study are 5 and 10, denoted as k-NN (k=5) and k-NN (k=10), respectively.

3.1.4 Multivariate imputation by chained equations (MICE)

MICE is an iterative approach where the missing value in a dataset is imputed based on error-minimization. MICE-based imputation approach adopts a series of estimations where each missing variable is imputed by regressing on the other variables. This is achieved by running through an iterative process. In the first iteration, the missing value is initially imputed by taking the mean of the observed values for that feature in the dataset. In the second iteration, the feature with the fewest missing values is selected, and the corresponding imputed mean values are removed from the dataset. In this iteration, the removed set is now considered as the test set, and the remaining dataset is considered as the training set. The first missing value in the test set is the dependent variable, whereas all the other values in the test set are considered independent variables. The independent variables from this test set are then imputed based on the regression model, which is developed based on the training set. In the third iteration, the previous steps are repeated for the remaining missing features, where the predicted feature from the previous iteration is also included as the independent variable. This entire process of iterating through the missing features is repeated until convergence is reached. Thus, the whole process is run multiple times to get multiple imputations. Thus, instead of a single imputation, this method uses multiple imputations and measures the uncertainty of the missing values systematically [59]. The flexibility and ability to handle different data types such as continuous or binary make the chained equations approach more attractive. Further details on MICE are available in [59].

3.2 Extreme Gradient Boosted Decision Trees (XGBoost)

After missing data imputation, the concrete strengths are predicted using ML techniques. While this section details XGBoost, all other ML techniques are adequately detailed in the supplementary material. XGBoost [60] is an extreme gradient boosted and tree-based machine learning model [61–64]. It is an iterative approach where the models are trained in succession to minimize the errors. The iteration
continues till no further improvements can be made. It is also considered an ensemble technique in which many tree models are combined to obtain the final one. XGBoost contains a toolset for scalable end-to-end tree boosting systems, sparsity-based algorithms, and justified weighted sketch for efficient proposal calculation [65]. XGBoost uses K additive function to predict the output, which is described as

$$\hat{y}_o = \sum_{k=1}^{K} f_k(x_i), \ f_k \in F$$

[1]

Where K is the number of regression trees algorithm (such as CART [66]) in a tree ensemble, f is a function in functional space (F), F is set of all possible CARTs and \(x_i\) is the input feature vector for \(i^{th}\) data point in the given dataset \(D = \{x_i, y_i\}, i = 1,2,3, \ldots, n\) (n is the total number of data points). As described above, the tree ensemble is created by iteratively adding new regression trees (CARTs) to improve the model accuracy from the prior models. However, it is impractical to find all the possible regression trees to improve the model accuracy substantially. Therefore, an optimal regression tree is created from a single node by iteratively adding branches in practice. This addition of the branches ceases when the allowable depth of a tree is reached, or the number of samples at a splitting node reaches the minimum number of samples.

### 3.3 Performance evaluation

The predictive capability of the ML techniques is evaluated using both mean square error (MSE) and coefficient of determination \((R^2)\). The MSE measures the average Euclidean distance between the predicted and true or measured values and is expressed as [64]:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_p(i) - y_t(i))(y_p(i) - y_t(i))$$

[2]

where \(y_p(i)\) and \(y_t(i)\) are the \(i^{th}\) predicted output and the \(i^{th}\) true output, respectively. MSE can estimate the accuracy of the predicted values from each model, where lower MSE indicates higher accuracy.

The coefficient of determination is used to quantify the variation of the predicted values from the observed values. In this study, the Pearson correlation coefficient [67] is used to indicate the accuracy of the predicted results. In the case of the sampled data, the Pearson correlation coefficient can be determined as follows:

$$R^2 = \frac{\sum_{i=1}^{n}(y_p(i) - \bar{y}_p)(y_t(i) - \bar{y}_t)}{\sqrt{\sum_{i=1}^{n}(y_p(i) - \bar{y}_p)^2 \sqrt{\sum_{i=1}^{n}(y_t(i) - \bar{y}_t)^2}}}$$

[3]

Where \(y_p(i)\) and \(y_t(i)\) are the \(i^{th}\) predicted output and the \(i^{th}\) true output, respectively.
In this study, both $MSE$ and $R^2$ of the train and test data are used to obtain the optimal degree of complexity and performance for each ML model, as explained in the forthcoming section.

### 3.4 Hyperparameter tuning

To increase the accuracy of the ML models and to avoid any potential overfitting, a fraction of the data points is kept fully hidden from the models and used as a “test set.” The test set is then used to evaluate the accuracy and performance of each model on these unseen data. To this end, a $k$-fold cross-validation (CV) technique [68] is adopted in this study. In the CV technique, the dataset is split into $k$ number of smaller sets, wherein at each fold, the model is trained on a fraction of data (training set) and tested on the remaining data. The final value obtained is the average value which is iteratively run on each of the $k$-folds. This study implements a nested two-level CV approach. The nested two-level cross-validation (CV) is implemented to avoid any risk of over-fitting. First, in the outer CV (5-fold), 15% of the data is first randomly split from the total dataset to be used as the test set, which is kept completely hidden from the model training and validation. Next, in the inner CV (5-fold), the remaining 85% of the data is further split into training and validation sets at an 85:15 ratio. Thus, the nested CV technique discards any arbitrary choice of the test set and partially mitigates issues arising from the limited number of data points. It needs to be noted that the entire dataset was randomly split into training, validation, and testing dataset. Besides, the test dataset was kept unseen by the model during the training and validation process. Such random selection of training, validation, test datasets, and complete separation of the test dataset during the model training and validation process helps avoid any data bias.

Developing a model that is accurate and simple at the same time is challenging. Therefore, there is a trade-off between accuracy and simplicity. While simple models often perform poorly or are under-fitted, overly complex models often show reduced performance prediction efficiency on the test data or unknown data sets. Such overly complex models can capture perfect trends on the training dataset but show poor transferability to unknown data sets and suffer from overfitting. Hence, to avoid such instances, models need to be optimized by tuning the hyperparameters to reach an ideal trade-off between accuracy and computational demand. This is achieved in this study by gradually increasing the degree of complexity (e.g., number of neurons or number of trees) and tracking the model prediction accuracy for both the training and validation datasets.
3.5 Model interpretability using SHAP

The ability of the ML models to learn from the known data and to predict the responses in the unseen or unexplored domains has led to the development of robust prediction tools. However, most of these ML models suffer from high complexity and low interpretability. SHAP, which is known as Shapley Additive Explanations (SHAP), is derived from Shapley values in game theory [69,70] and is employed to measure the importance of various features within the models [42,71]. As per SHAP, the importance of feature $j$ for the output of model $f$, $\phi^j(f)$, is a weighted sum of the feature's contribution to the model's output $f(x_i)$ over all possible feature combinations [72]. $\phi^j(f)$ is expressed as:

$$\phi^j(f) = \sum_{S \subseteq \{x^1, ..., x^p\} \setminus \{x^j\}} \frac{|S|!(p-|S|-1)!}{p!} (f(S \cup \{x^j\}) - f(S))$$

[4]

Where $x^j$ is feature $j$, $S$ is a subset of features, and $p$ is the number of features in the model. In SHAP, feature importance is measured by quantifying the prediction error while perturbing a given feature value. The importance of the feature is weighted by the sensitivity of the prediction error when perturbing the feature value. SHAP is also used to describe the performance of a trained ML model. SHAP uses an additive feature attribution method, i.e., an output model is considered a linear addition of the input variables to describe an interpretable model. For example, a model with input variables $x^i$ where $i$ varies from 1 to $k$, and $k$ is the number of input variables, the explanation model $h(x_s)$ with simplified input $x_s$ for an original model $f(x)$ is expressed as

$$f(x) = h(x_s) = \phi_0 + \sum_{i=1}^p \phi_i x^i_s$$

[5]

Where $p$ signifies the number of input features, and $\phi_0$ represents the constant value when no information is provided (or no inputs used). Inputs $x$ and $x_s$ are correlated through a mapping function, $x = m_x(x_s)$. Equation 5 is further demonstrated in Figure 2, where the terms $\phi^0, \phi^1, \phi^2$ and $\phi^3$ increase the predicted value of $h()$, while $\phi^4$ decreases the value of $h()$. As indicated in [42], there exists a single solution for Equation 5, which has three desirable properties: local accuracy, missingness, and consistency. The local accuracy ensures that the output of the function is the sum of the features attributions where the model is required to match the output of $f$ for the simplified input $x_s$. The local accuracy is obtained when $x = m_x(x_s)$. Missingness ensures that for missing features, no importance is assigned, i.e., for $x^i_s = 0$ implies $\phi^i = 0$. Lastly, through consistency, changing a larger impact feature does not reduce the attribution assigned to that feature.
4. RESULTS AND DISCUSSION

4.1 Performance evaluation for different missing data imputation techniques

In this section, the performance of imputation approaches is evaluated by comparing the MSE values computed from respective ML models. The imputation methods implemented in this study are: (1) complete removal of the missing data, which is denoted as ‘Data elimination’, (2) missing value imputation based on the mean of the non-missing value of the respective features, which is denoted as ‘Mean’, (3) imputation based on the median of the non-missing value of the respective features, denoted as ‘Median’, (4) k-NN with five nearest neighbors, denoted as k-NN (k=5), (5) k-NN with ten nearest neighbors, denoted by k-NN (k=10) and (6) ‘MICE’, in which the missing data is imputed multiple times through an iterative series of predictive models until the convergence is reached. While XGBoost is detailed earlier in this paper, other approaches such as NN, LASSO, and random forest are adequately shown in the supplementary material. Figures 3(a) and (b) show the MSE values corresponding to different ML techniques on the imputed datasets for the prediction of compressive strength and tensile strength of concrete, respectively. It is to be noted that the MSE value presented in Figure 3 is based on the training accuracy.
Figure 3. MSE values obtained from the optimized ML models after data imputation method for predicting (a) compressive strength and (b) tensile strength of concrete

From Figures 3(a) and (b), it can be clearly observed that XGBoost and NN show better performance as compared to LASSO and RF for all the imputation techniques. It can also be seen in Figure 3(a) and (b) that the minimum MSE value is obtained when the missing value is imputed using k-NN (k=10), and XGBoost is used for performance prediction. While XGBoost shows better performance as compared to NN for k-NN (k=10) imputation approach for compressive strength predictions (Figure 3(a)), NN shows lower MSE values than those obtained for XGBoost for other imputation approaches such as Mean, Median, and k-NN (k=5). In the case of tensile strength predictions, a comparison between NN and XGBoost reveals that XGBoost performs better for k-NN (k=10), Median, and Mean, whereas NN shows lower MSE values for Data elimination, k-NN (k=5), and MICE. From Figure 3(a), it can be observed that the SVM model for compressive strength of concrete with data elimination is associated highest error value, whereas, for k-NN, SVM’s performance is comparable to RF. For mean and median, SVM shows slightly reduced MSE values as compared to RF. In the case of tensile strength of concrete (Figure 3(b)), SVM underperforms for data elimination but outperforms LASSO for all other imputation approaches, and overall, SVM’s performance is comparable to that of RF for all the imputation approaches except the data elimination approach. Since Lasso is a linear model and concrete’s tensile and compressive strength have a non-linear relationship with the input features, the error is quite high for all the imputation cases when this model is used. RF, which is an ensemble-based method, learns multiple regressors for the given dataset, and the output predictions are obtained by utilizing the prediction of all the models present in the ensemble.
Therefore, the error is again high for RF models. On the other hand, XGBoost is based on a gradient boosting approach, and it utilizes the ensemble of multiple weak models to arrive at the final predictions. Since each model in the XGBoost ensemble is created on the basis of data points, the iterative process of training yields a rational XGBoost model which may overfit. Therefore, using intensive cross-validation procedure and testing on the hidden test set, we were able to obtain a generalized model which performs well on all the splits of the data. From Figure 3, it can be easily observed that XGBoost is able to learn the underlying distribution and is providing lower errors, and the performance is significantly enhanced after data imputations. The same is the case of NNs due to their complex architecture and the ability to learn the non-linear relationships between inputs and output. Nevertheless, based on the broad observations, both XGBoost and NN show lower levels of MSE values. Overall, from Figures 3(a) and (b), it can be inferred that the selection of the best ML approach depends on the type of missing data imputation technique adopted. Since XGBoost with k-NN (k=10) shows the best performance, for the brevity of discussions, this paper primarily details the model performance for XGBoost, whereas the rest of the ML techniques are provided in the supplementary material.

### 4.2 Hyperparameter optimization and training process

This section describes the training and hyperparameter optimization results for XGBoost. For the XGBoost tree ensemble model, the maximum depth, the minimum child weight, and the learning rate are set to 9, 5, and 0.09, respectively. Figure 4(a) and (b) correspond to the MSE and $R^2$ values of the compressive strength predictions with an increasing number of trees (considered as the model complexity). Figures 4(c) and (d) depict the MSE and $R^2$ values, respectively, with increasing model complexity for tensile strength predictions. It is observed that as the model complexity is increased, the model shows a saturation behavior beyond a certain number of trees for the validation set while the MSE value keeps on reducing for the training set. Thus, the optimized number of trees was found to be 55 for both compressive and tensile strength predictions. It is worth mentioning that although the $R^2$ value for all the model cases is above 0.9, the MSE value provides a better understanding of the model training and validation performance. The optimized hyperparameters obtained in this section are leveraged to evaluate the performance prediction efficacy as explained hereafter. Please refer to Table S1 in the supplementary material for the optimized values of the hyperparameters corresponding to different ML approaches.
4.3 Prediction of concrete strengths using XGBoost tree ensemble

In this section, the prediction of the compressive and tensile strength of the concrete is presented when the k-NN (k=10) imputation method is used, and the hyperparameter-optimized XGBoost, tree ensemble model, is implemented. While XGBoost is primarily focused here for the brevity of discussions, all the other predictive models implemented in this study are provided in the supplementary material. Moreover, a performance comparison between these ML techniques is also detailed in the forthcoming section.
Figure 5(a) shows the prediction of concrete compressive strength using XGBoost. Similar high $R^2$ values are obtained for all the sets, indicating that the XGBoost tree ensemble model can interpolate well with the unseen data, as shown in Figure 5(a). Similar overall trends are observed for tensile strength predictions, as shown in Figure 5(b). However, at high tensile strengths, the prediction efficacy reduces. This could be due to the fewer data points available for training corresponding high tensile strength values. Overall, such excellent correlations between the experimental and predicted strengths establish the exceptional prediction efficacy of the missing data imputation techniques and the ML model considered here.

![Graphs showing predicted vs. experimental values for compressive and tensile strengths](image)

**Figure 5.** Comparison of (a) the predicted compressive strength and (b) the predicted tensile strength of concrete using XGBoost tree ensemble model with the experimental values.

### 4.4 Comparison of the predictive performance of different ML approaches

For a comparison between different ML techniques implemented in this study, MSE and $R^2$ values (both training set and test set) obtained for compressive strength and tensile strength of concrete are shown in Table 2. The details on LASSO, SVM, RF, and NN are provided in the supplementary material for ease of reference. While the accuracy of the model to interpolate on the known data can be characterized by evaluating the model on the training set, the accuracy of model prediction for the unknown data can be captured using the test set. For all the ML approaches, it is observed that the $R^2$ is close to 1. However, the MSE values for the different models show high variations. For instance, the $R^2$ value for the compressive strength using SVM is 0.99 (train set), but it is associated with a high MSE value of 11.87 MPa$^2$ (train set) compared to other models. This signifies that the predictive capability of ML models should be ascertained by both high $R^2$ values and low MSE values [73].
In general, it is required for the model to have low complexity and high interpretability. However, in most cases, there is a trade-off between model complexity and interpretability to achieve a high level of accuracy. For example, LASSO has high interpretability, and it reduces the complexity of the model by adding weights to the model coefficients. From Table 2, it is observed that the MSE offered by LASSO for the train set is low compared to SVM and RF. However, the prediction error of the LASSO on the test set is relatively higher than SVM and RF. Whereas in the case of tensile strength, the MSE value offered by LASSO from both training set and test set is higher than SVM and RF. Nevertheless, LASSO with a polynomial degree of 3 signifies that the composition-property relationship of the concrete is not linearly correlated, which is vital to developing a predictive model. On the other hand, RF, which is based on the ensemble of decision trees where large trees are trained individually, shows acceptable accuracy on the train set and good predictability on the test set on the compressive strength of concrete. A similar case is also observed for the tensile strength. NN also provides good prediction efficacy for both tensile and compressive strengths. Overall, XGBoost offers the best performance among all the ML techniques with the least MSE value and high $R^2$ value for both compressive strength and tensile strength predictions. The forthcoming section implements SHAP to shed more light on the interpretability of the predicted responses obtained from XGBoost.

**Table 2:** Performance comparison for compressive strength prediction of concretes from various ML Techniques

<table>
<thead>
<tr>
<th>Models</th>
<th>MSE</th>
<th>R²</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train</td>
<td>Test</td>
</tr>
<tr>
<td>Lasso</td>
<td>8.65</td>
<td>37.65</td>
</tr>
<tr>
<td>SVM</td>
<td>11.87</td>
<td>28.32</td>
</tr>
<tr>
<td>RF</td>
<td>10.07</td>
<td>21.96</td>
</tr>
<tr>
<td>XGBoost</td>
<td>1.98</td>
<td>14.57</td>
</tr>
<tr>
<td>NN</td>
<td>4.72</td>
<td>15.56</td>
</tr>
</tbody>
</table>

**Table 3:** Performance comparison for tensile strength prediction of concretes from various ML Techniques

<table>
<thead>
<tr>
<th>Models</th>
<th>MSE</th>
<th>R²</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train</td>
<td>Test</td>
</tr>
<tr>
<td>Lasso</td>
<td>0.14</td>
<td>0.27</td>
</tr>
<tr>
<td>SVM</td>
<td>0.08</td>
<td>0.20</td>
</tr>
<tr>
<td>RF</td>
<td>0.07</td>
<td>0.13</td>
</tr>
<tr>
<td>XGBoost</td>
<td>0.02</td>
<td>0.11</td>
</tr>
<tr>
<td>NN</td>
<td>0.03</td>
<td>0.12</td>
</tr>
</tbody>
</table>
4.5 Interpretability of the trained ML models

ML models such as XGBoost have a highly complex and non-linear architecture, due to which they tend to behave as black-box models. While the tree models are explainable, thanks to their hierarchical structure, the visualizations of these models may not be easy to decipher. Here, we show that SHAP can be a very useful model agnostic tool to interpret complex machine learning models with a large number of parameters. Since the XGBoost exhibits the best performance in predicting compressive and tensile strengths of concrete, we interpret these models in this section for the tensile and compressive strength of concrete using SHAP.

For XGBoost, a tree-specific SHAP approximation method, namely TreeExplainer [74] is used. TreeExplainer exploits the internal structure of tree-based models, which is summed over to a set of calculations specific to the leaf node of a tree model that led to low order polynomial complexity [74]. Figure 6(a) demonstrates the mean SHAP values corresponding to various features for the compressive strength predictions as obtained from the XGBoost tree ensemble model.

![Figure 6](image)

**Figure 6.** SHAP summary plot for concrete (a) compressive strength and (b) tensile strengths obtained using XGBoost

It is observed that for the prediction of compressive strength, the water-to-cement ratio has the maximum SHAP value. The feature ‘curing age’ has the second maximum SHAP value. Overall, the water-to-cement ratio, water-to-binder ratio, and curing age are the most dominant features for predicting the compressive strength of concrete, as observed from XGBoost.

Figure 6(b) demonstrates the mean SHAP values corresponding to different features for the tensile strength predictions as obtained from the XGBoost tree ensemble model. Here, the water-to-cement ratio shows the most dominant SHAP value, followed by curing age and water-to-binder ratio. It is interesting...
to observe in Figure 6(b) that the tensile strength of cement binder exerts a low impact on the overall prediction of the tensile strength of concrete. However, in reality, the tensile strength of binders should have a significant influence on the tensile strength of concretes. This can be explained from the tensile strength dataset inconsistencies arising from the fact that the tensile strength test in binders and concretes is very sensitive to the type of test (whether a uniaxial tensile test or split tensile test) used across the dataset and to the presence of cracks compared to the compression test [75–79]. Besides, between the tensile strength of the cement binder and concrete, a high variation of the tensile strength of concrete is expected for the same binder features owing to the presence of aggregates which induce cracks or other defects that make the tensile strength results very sensitive. Thus, it needs to be noted that because of the lack of good correlation between the tensile strengths of the binders and concretes in the dataset, the SHAP analysis downgrades the importance of the tensile strength of binder in dictating the tensile strength of concrete, which needs further investigation.

It can be observed from Figure 6(a) that features such as stone powder content and the sand ratio has a lower influence on the compressive strength predictions. To evaluate their influence, two different models were constructed. While for the first case, stone powder content was removed from the model, the second model was constructed by removing both stone powder content and sand ratio. It is observed that the MSE value increases significantly when only the stone powder content is removed. On the other hand, when both stone powder and sand ratios are removed from the model, the MSE values are reduced as compared to the case with only sand power content being removed. However, the MSE values are still higher compared to the model with all the features. Such a trend is expected as XGBoost is a tree-based ensemble approach where each decision tree is created based on the dataset. Similar observations are also reported in [80], where the RMSE value is higher when removing one feature compared to removing 2 and 3 features. Please refer to section 3 of the supplementary material for the quantified results.

Figure 7 shows the violin plot of the SHAP values for each feature used to predict the compressive and tensile strengths of concrete using the XGBoost model. For each feature, the color represents the feature value, and its corresponding x-axis (SHAP) value represents the contribution to the output property. For instance, for the input feature water-cement ratio, blue-colored points on the right represent low values, and red-colored values on the left represent high values of the water-cement ratio. Correspondingly, a SHAP value of 20 for the rightmost point suggests that a low water-cement ratio can increase the compressive strength by 20 MPa from the mean value. This is subjected to the dataset utilized in the current study, and more accurate results can be obtained with a greater number of data points. On the
other hand, the corresponding SHAP value for tensile strength (Figure 7(b)) is 1.5 MPa. Thus, the increase in tensile strength due to low values of water-cement ratio is only 1.5 MPa from the mean value. Thus, the main interpretations from these observations are that: (i) both tensile and compressive strengths decrease with increasing water-cement ratio. This can be explained by the fact that the high water to cement ratio is often connected to high porosity due to the high volume of voids that are not filled by hydration products [81,82]. This, in turn, degrades the strength of concrete [83,84]. (ii) the increase in compressive strength due to a decrease in the water-cement ratio is almost an order magnitude higher than the corresponding increase in the tensile strength. A similar trend was reported by Chen et al. [85] for cement mortars, where a significant decrease in the compressive-to-tensile strength ratio was observed with increasing porosity. Such interpretations allow an expert to ensure that the functions learned by the ML models are reasonable and physically sensible.

From Figure 7, it is evident that the water-cement ratio, curing age, and water-binder ratio are the top three features contributing to both tensile and compressive strength predictions. Among these, curing age seems to have a positive impact on the compressive and tensile strengths, while water-cement and water-binder ratios seem to have a negative impact. It should be noted that this observation is very well consistent with the present understanding of the development of compressive and tensile strengths in concrete upon curing. For other features, the SHAP values are mostly centered around zero. These low values signify their lower importance towards the model output. In other words, these parameters do not affect the tensile and compressive strength significantly for the feature range observed in the dataset. It needs to be noted that Figures 6(a) and (b) show the global importance factor of the input variables, and such global importance values are obtained as the average of the absolute SHAP value per feature across the data. As such, these average SHAP values in Figures 6(a) and (b) are path independent [72,86]. While Figures 6(a) and (b) provide the path-independent global importance of the input variables, Figures 7(a) and (b) reveal the range and distribution of impacts of input variables on the prediction of compressive strength and tensile strength of concrete.
Figures 8(a) and (b) show the river flow plots for compressive strength and tensile strength of concrete, respectively, using the XGBoost model. The expected model value is the model output when values of any of the input features are not available. In other words, when no information about the input features is available, the best estimate of the output value will be the mean of the dataset used for training the model. Adding the information about each of the input features one-by-one, followed by all possible combinations, provides insight into the individual and the collective roles played by each of the input features.
features in governing the output property. To demonstrate this effect, each line in the SHAP river flow plot (see Figure 8) corresponds to one data point.

Figure 8. SHAP river flow plot for concrete (a) compressive and (b) tensile strengths using the XGBoost model. Here the colors represent the output values of compressive/tensile strength predicted by the model considering all the input features. The blue color represents lower values of the output strengths, and the red color indicates higher output strengths (for additional information, please refer to the supplementary material).

The line color represents the property value corresponding to the given data point, the red color is associated with higher values and blue with the lower values. In other words, a red line represents a high compressive strength concrete, and blue represents a low compressive strength one. The rise and fall of
the line corresponding to each of the features with respect to the expected value shows how these respective features control the final property value. The features are arranged in increasing order of average SHAP value from left to right of the x-axis. Interestingly, we observe that the curing age shows a mixed effect. Specifically, for both compressive and tensile strengths, the curing age seems to both increase as well as decrease the model predictions for concretes having high compressive and tensile strengths (red lines). Similar trends are observed for concrete with low tensile and compressive strengths as well (blue lines). Note that water alone as a feature is not having a significant impact on any property. However, it becomes the top contributing feature when it is used as a ratio with cement and binder.

While making model predictions, it is essential to understand how different features in the model are dependent on each other. Many times, such dependencies may not be revealed from simple correlations. Figures 9(a) and (b) show the interaction values plot for compressive and tensile strengths. The interaction value plots reveal the interdependencies between the input features for a given output. Note that the diagonal value of the SHAP interaction plot corresponds to the mean SHAP value of the feature. Here, for the sake of visualization, the values have been normalized between 0 to 1. As expected, both water-binder and water-cement ratios exhibit strong dependency. Interestingly, curing age also exhibits a dependency on these ratios. Among others, the water-cement ratio exhibits a dependency on the crushed stone size, while the curing age and water-binder ratio exhibit little or no dependency on it. Another similar feature that seems to have a stronger dependency on the water-cement ratio is the sand ratio, although it has little or no dependency on the water-binder ratio. Indeed, water exhibits dependency with both water-cement ratio and water-binder ratio in the case of both tensile and compressive strengths. In Figure 9, it can be observed that the water-binder ratio and water-cement ratio exhibit strong dependency. As such, evaluating the influence of handcrafting of the input variables such as decoupling the water-cement ratio and using feature engineering to compare the overall predictive performance of the models with decoupled or independent variables can lead to an interesting future study. However, for the context of this paper, since the dataset is experimentally retrieved from the literature, the ML models are implemented here using all the original input features reported in the dataset to avoid any preprocessing or add any prior knowledge by decorrelating the input variables in the dataset. Moreover, it is the core of the machine learning algorithms such as Neural Network to learn the pattern and find any correlation and dependencies from the data which are shown in this paper.

Overall, the relative importance of different features is found to be similar between compressive and tensile strength predictions using XGBoost. The visualizations generated using SHAP values provide
comprehensible insights into model predictions. The interactions among different features reveal how a combination of the factors influences the compressive and tensile strengths of concrete. An evaluation of such relative importance of different features through interpretable ML can provide efficient means to optimize the material quantity and quality for desired performance while also gaining fundamental insights into the material behavior.
Figure 9. SHAP interaction plot for concrete (a) compressive and (b) tensile strengths using XGBoost model.
5. CONCLUSION

This paper presents the strength prediction for concretes using various ML approaches, including PR, LASSO, SVM, RF, XGBoost, and NN. The dataset used in this study corresponds to concrete mixtures with manufactured sand used as fine aggregates, which makes it different from conventional concrete mixtures. Such modifications of the components in concretes add more complexity to its composition-property relationship. Besides, the dataset contains a significantly higher degree of missing data, which makes it challenging to obtain good performance predictions. Using the ML technique, the non-linear relationships between the inputs and the outputs are assessed, which are overlooked by the physics or chemistry-based models. The model performance of various ML techniques is evaluated based on the MSE and $R^2$. Among all the trained models, XGBoost and NN exhibit excellent prediction efficacy when kNN with ten neighbors ($k=10$) was leveraged for imputation of the missing data. Moreover, the SHAP-based model interpretability technique is leveraged for the interpretation of the predicted results in terms of the relative importance of different input features. Based on SHAP values, it was found that the compressive strength and the tensile strength of concrete are dominantly influenced by the water-to-cement ratio, water-to-binder ratio, and curing age. Besides, the relative importance of all other features such as sand ratio, fineness modulus of the manufactured sand, curing age, the maximum size of the crushed stone, and tensile/compressive strength of the binder phase are also reported. Such evaluation of the relative importance of different features and their influence on the concrete strengths will potentially enable materials designers and decision-makers to make informed decisions regarding the selection of appropriate raw materials so as to obtain the desired strength performance. Moreover, the assessment of relative feature importance will potentially enable the development of better physics and chemistry-based predictive models for the emerging sustainable alternative concretes. Overall, the results in this paper demonstrate machine learning as a promising tool to predict the strength of concrete based on the knowledge of their mixture proportions. In the future, it is desirable to develop a more generalized model using various feature engineering techniques, which can open up various promising pathways for more efficient input feature selections by omitting the redundant input features, thereby reducing the time for model training without sacrificing the model’s accuracy.

6. ACKNOWLEDGEMENTS

This research was conducted in the Multiscale & Multiphysics Mechanics of Materials Research Laboratory (M4RL) at the University of Rhode Island and Multiphysics & Multiscale Mechanics Research Group (M3RG) at IIT Delhi. The authors acknowledge Bluewaves and Andromeda High-Performance Research
Computing at the University of Rhode Island (URI) for providing computer clusters and data storage resources that have contributed to the research results reported within this paper. The authors also thank the IIT Delhi HPC facility for providing the computational and storage resources.

7. REFERENCES


