Case Studies in Construction Materials 16 (2022) e01059

Contents lists available at ScienceDirect

Case Studies in Construction Materials

journal homepage: www.elsevier.com/locate/cscm

A novel approach to explain the black-box nature of machine learning in compressive strength predictions of concrete using Shapley additive explanations (SHAP)

I.U. Ekanayake [a], D.P.P. Meddage [b,c,*], Upaka Rathnayake [c]

[a] Department of Computer Engineering, Faculty of Engineering, University of Peradeniya, Peradeniya, Sri Lanka
[b] Department of Civil and Environmental Engineering, Faculty of Engineering, University of Ruhuna, Sri Lanka
[c] Department of Civil Engineering, Faculty of Engineering, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka

ARTICLE INFO

Keywords:
Explainable machine learning
Compressive strength
Tree-based regression
SHAP explanation
Laplacian kernel Ridge Regression

ABSTRACT

Machine learning (ML) techniques are often employed for the accurate prediction of the compressive strength of concrete. Despite higher accuracy, previous ML models failed to interpret the rationale behind predictions. Model interpretability is essential to appeal to the interest of domain experts. Therefore, overcoming research gaps identified, this research study proposes a way to predict the compressive strength of concrete using supervised ML algorithms (Decision tree, Extra tree, Adaptive boost (AdaBoost), Extreme gradient boost (XGBoost), Light gradient boosting method (LGBM), and Laplacian Kernel Ridge Regression (LKRR). Alternatively, SHapley Additive exPlainations (SHAP) – a novel black-box interpretation approach - was employed to elucidate the predictions. The comparison revealed that tree-based algorithms and LKRR provide acceptable accuracy for compressive strength predictions. Moreover, XGBoost and LKRR algorithms evinced superior performance (**R = 0.98**). According to SHAP interpretation, XGBoost predictions capture complex relationships among the constituents. On the other hand, SHAP provides unified measures on feature importance and the impact of a variable for a prediction. Interestingly, SHAP interpretations were in accordance with what is generally observed in the compressive behavior of concrete, thus validating the causality of ML predictions.

1. Introduction

Concrete is the most extensively used construction material in the modern era. Given that it is generally a weaker material in

Abbreviations: AdaBoost, Adaptive Boosting; ANFIS, Adaptive Neuro-Fuzzy Inference System; ANN, Artificial Neural Network; BP, Back Propagation; BPNN, Back Propagation Neural Network; CFRP, Carbon Fiber Reinforced Plastic; CNN, Convolutional Neural Network; CoV, Coefficient of Variation; ECSO, Enhanced Cat Swarm Optimization; ELM, Extreme Learning machine; FA, Firefly Algorithm; GEP, Gene Expression Programming; LGBM, Light Gradient Boosting Method; LIME, Local Interpretable Model-Agnostic Explanations; LKRR, Laplacian Kernel Ridge Regression; LSSVR, Least Square Support Vector Regression; MAE, Mean Absolute Error; MAPE, Mean Absolute Percentage Error; MART, Multiple Additive Regression Trees; MFA, Modified Firefly Algorithm; ML, Machine learning; MSE, Mean Square Error; R, Coefficient of Correlation; R^2, Coefficient of Determination; RISE, Real-time Intelligence with Secure Explainable; RMSE, Root Mean Square Error; SHAP, Shapley Additive exPlainations; SVM, Support Vector Machines; SSE, Sum of Square Error; XGBoost, Extreme Gradient Boosting.

* Corresponding author at: Department of Civil and Environmental Engineering, Faculty of Engineering, University of Ruhuna, Sri Lanka.
E-mail address: meddagedpp.20@uom.lk (D.P.P. Meddage).

https://doi.org/10.1016/j.cscm.2022.e01059
Received 13 January 2022; Received in revised form 25 March 2022; Accepted 6 April 2022
Available online 7 April 2022

tension, steel reinforcement can enhance concrete's strength characteristics. Thus, reinforced concrete is suitable for structural applications due to its high strength and ability to cast into a defined shape [1]. In essence, the hardened state of concrete resists water and high temperature, allowing it to be used under different exposure and environmental conditions (e.g. bridge structures, tunnels, and dams). Primary constituents of concrete are cement, coarse aggregates, fine aggregates, and water. Usually, most the constituents are locally available, therefore, the cost of concrete is lower compared to structural steel. In addition to primary materials, nowadays, fly ash, blast furnace slag, and chemical admixtures are used to improve the performance of concrete [2].

However, the presence of additional constituents showed a non-linear behavior in the strength characteristics of concrete [1]. In particular, compressive strength is a dominant factor that reflects the performance of concrete. It strongly depends on the composition of concrete. For instance, high water content compared to cement content tends to drop compressive strength [3,4]. In addition, air content inside the mixture can lower the compressive strength. Besides, different exposure conditions influence the strength characteristics of concrete. Combinations of additional constituents and different exposure conditions result in complex relationships whereas a consistent approach was required to predict compressive strength [5–9]. Despite the omnipresence of civil infrastructure, the compressive strength of concrete is regularly concerned to ensure the safety of each design. Moreover, compressive strength plays a vital role in the strength assessment of old structures as well.

Generally, the compressive strength values are obtained through laboratory experiments [10]. Cubic or cylinder specimens are cast using specific composition and cured. After 28 days, each specimen is subjected to gradual compressive loading until failure to obtain compressive strength. As a result of the dependency on age, more specimens are required to develop an acceptable strength development curve. There, it leads to consuming time and money, as well as decelerates working efficiency [11]. Therefore, the community moved from experimental work to empirical regression models to alter the aforementioned constraints. For example, Feng et al. [11] established several regression models to predict compressive strength. However, the non-linearity of parameters caused complex relationships. Afterward, numerical simulations were proposed to reproduce the behavior of concrete which repeatedly appeared to be inappropriate as a result of complexity.

Later, soft computing techniques such as machine learning (ML) algorithms were proposed to capture complex behavior of predictions. Numerous attempts have been already taken to investigate the possibility of ML for this purpose [10,12–27]. ML techniques can divide into several categories, such as classification, regression, and clustering. The research community observed that they can predict accurate results using ML. On other hand, employing ML has minimized time and cost constraints that come across in traditional approaches. Chaabene et al. [16] provided a comprehensive review of ML-based studies on the mechanical properties of concrete.

Feng et al. [11] described several algorithms that fit well into compressive strength predictions. Among them, Artificial Neural Networks (ANN) and Support Vector Machines (SVM) received major attention [10,28–31]. Siddique et al. [31] formed an ANN to predict the compressive strength of self-compacting concrete containing bottom-ash. Furthermore, Uysal and Tanyildizi, [32] proposed to involve ANN to perform compressive strength predictions under high-temperature exposure. Further, literature showcased that ANN could even manipulate good results for compressive characteristics of concrete which contained recycled aggregate [33]. Both ANN and SVM techniques were acceptable for compressive strength prediction [28]. In addition, Aiyer et al. [34] extended their respective studies using the SVM approach. Gholampour et al. [35] argued that LSSVR models are effective in predicting the compressive strength of concrete with recycled aggregates. Pham et al. [30] stated that the hybrid models with FA and LSSVR enhance the performance of compressive strength predictions with respect to individual ANN and SVM models. Deng et al. [36] proposed that CNNs perform better than conventional neural networks when predicting compressive strength. Topcu and Saridemir [37] developed a method to predict the compressive strength of concrete with recycled aggregates using BPNN on gradient descent and fuzzy logic models. Based on their predictions, BPNN was identified as the superior model. Later, Naderpour et al. [38] assessed BPNN and the corresponding predictions using a sensitivity analysis. They found that BPNN accurately predicts compressive strength. Bui et al. [39] proposed the MFA-ANN approach to predict compressive and flexural strength of high-performance concrete whereas the Pearson correlation of predictions reached 0.95. Dao et al. [40] used ANFIS and ANN models to predict the compressive strength of geopolymer concrete. Proposed methods successfully performed the predictions (ANFIS: $R^2 = 0.879$, and MAE = 1.655, ANN: $R^2 = 0.851$, and MAE = 1.989). Latest studies by Feng et al. [11] and Nguyen et al. [1] claimed that boosting algorithms can strong-arm classical ML models to perform hand in hand with ANNs. Moreover, the research community employed tree-based models to predict compressive strength [41–45]. For example, Han et al. [46] proposed a prediction method using random forests which achieved $R > 0.96$, and RMSE < 4.43 without hyperparameter tuning. M5P algorithm which is a combination of decision tree and linear regression has been often used in compressive strength predictions [47]. The M5P model developed by Behnood and Golshafani [42] achieved a good testing accuracy ($R^2 = 0.932$ and RMSE = 4.715). Besides, evolutionary algorithms have been employed in several studies [47,48]. Evolutionary algorithms generate a category of empirical search methods which are based on biological evolution mechanisms. For instance, Gandomi et al. [49] introduced a method to predict the compressive strength of CFRP confined concrete cylinders based on genetic programming. They argued that the proposed methods obtained an acceptable accuracy (Testing accuracy: $R > 0.86$ and MAPE<14).

In contrast, these studies failed to highlight the interpretability of their models. Despite their higher accuracy, many of these ML applications are not explainable and the user does not aware underlying reasoning of predictions. In terms of the traditional experiments, obtained results are explainable and transparent. In consequence, experts do not entrust ML applications even with higher accuracy. Therefore, model transparency should be increased in order to establish ML in civil engineering applications. Predictions should be interpretable even for a non-technical community. Moreover, a non-technical community generally has zero knowledge of the ML-based methodology, which is unexplainable (*black-box*). Hence, it further leads the implementation of ML methods impractical. The absence of vital information such as feature importance, the relationship between inputs and outputs, the reasoning behind the predictions diminishes the confidence of end-user on ML approaches.

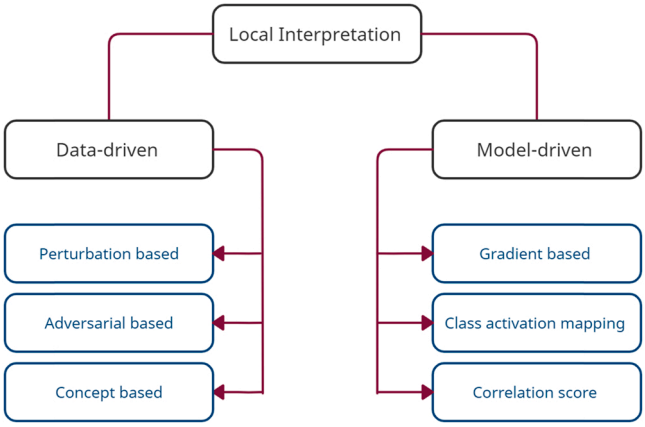


Fig. 1. Classification of local interpretation methods.

As a novel method, explainable ML proposes to eliminate earlier issues with model interpretation. It provides a better overview of how a particular prediction is made. Therefore, explainable ML holds due attention among data analysts, the business community, data scientists, and consumers [50]. With advanced features, explainable ML converts *black-box* type models to *glass-box* (transparent) models, exposing the underlying reasoning. These interpretable models elucidate how ML model provide their predictions, revealing the feature importance and dependency of variables [51,52]. Recent studies have introduced post-hoc explanations to be used for compressive strength predictions [53,54]. However, the studies did not explicitly highlight how a particular prediction was obtained. In contrast, they only provided insights using global feature importance. On the other hand, explaining the reasoning behind a prediction (instance) is imperative to cross-validate ML predictions, improve domain experts' and end-users' confidence in predictions, and explain a complex instance. Therefore, the authors believe that the present study is novel as it not only elucidates the model (*global explanation*), but also provides underlying reasoning on how a particular prediction was obtained (*instance-based explanations*).

In this workflow, we proposed five decision-tree-based algorithms (Decision tree, Extra tree, AdaBoost regressor, Light Gradient Boosting Machine (LGBM), and Extreme Gradient Boosting (XGBoost)) and Laplacian Kernel Ridge Regression (LKRR) to predict the compressive strength of concrete. The main reason to employ a decision tree is that it is self-explainable at each depth. A classic decision tree mimics the human thinking process therefore, the predictions are transparent [55,56]. However, other complex tree formations are not simply explainable compared to a decision tree. Therefore, we require a separate post-hoc explanation to interpret complex models. For the model building, experimental data from UCI (University of California machine learning repository) were employed (https://archive.ics.uci.edu/ML/datasets/Concrete+Compressive+Strength [57]).

Addressing the previous research gaps, this research work aimed to interpret the black-box nature of ML in predicting the compressive strength of concrete. More importantly, the study intends to elucidate how a particular prediction was obtained. The novel study involves Shapley additive explanations (SHAP) explanations that reflect model and instance-based resolutions. Interestingly, it emphasizes the contribution of each feature to the corresponding prediction and global and local behavior of the model to obtain end-users trust in the ML approach. Further, this study stresses using explainable ML does not essentially sacrifice precision or complexity but rather reinforces a model's predictions by affording human-intelligible explanations.

The main deliverables of this research include (a) directing compressive strength predictions using Tree-based ML and Laplacian-Kernel Ridge regression; (b) involving novel ML interpretable models to explain the black-box nature of predictions and interaction of variables, (c) establishing end-users' and domain experts' trust in ML techniques, confirming ML models perceive physical engineering characteristics.

2. Explainable ML

Conventional ML-based predictions require post-hoc interpretations to advance decision-making sequences. Interpretable models allow the community to understand the rationale behind the prediction. The ubiquitous presence of ML applications has emphasized interpretability is likewise important as precision [58,59]. Explainable ML provides necessary add-in to ML models by improving transparency of automatically obtained predictions . Fig. 1 illustrates the classification of local interpretation methods.

Foremost, such models are clustered into two groups namely, model-driven interpretation and data-driven interpretation [61]. Model-driven interpretations examine inner components of ML models and it does not specifically require a global explanation of the working methodology of the entire ML model. Instead, the interpretation depends on a specific category to elucidate how ML performs a given task. Model-driven methods are appeared to be low computationally expensive and easy to implement. Further, the cluster is divided into three sub-sections such as gradient-based interpretation [62,63], class activation map [64–66], and correlation-score interpretation [67–70].

Data-driven explanations rely on inputs for interpretation sequence, however, it does not require understanding working methodology inside an ML model, as well. It examines the effect of distinct input-data deviations on the ML model in terms of quantification

sequence. There are three sub-sections under data-driven interpretations such as adversarial-based interpretations [71,72], concept-based interpretations [73,74], and perturbation-based interpretations [63,75,76]. For the present study, we have employed perturbation-based interpretations to improve the transparency of ML models.

Perturbation works by masking a fraction of input data of the ML model. Masking different regions assist to obtain a set of disturbances. Subsequently, the disturbed set is fed into the model to acquire a new set of predictions. Because the disturbed-set corresponds to another set of predictions, that can be compared with original predictions. There, it leads to estimate the importance of various positions of input data set through a set of postulated rules. Evidently, strategy and interpretation rules reflect the contrast between each perturbation method. In essence, it includes some explanatory models such as LIME [77], CXplain, RISE [78], and SHAP [79]. SHAP and LIME are widely used in the applied ML research fields. A major difference between SHAP and LIME is the method followed to generate weights. LIME weighs an instance following similarity to the corresponding original instance whereas SHAP determines Shapley value to estimate weight corresponds to a sampling instance. Further, Moradi and Samwald [80] described that LIME creates dummy instances considering the neighborhood of a particular instance where it does not provide the actual feature value of the instance. In contrast, the SHAP value is recognized as a unified measure of feature importance. Hence we have used SHAP explainable model to interpret ML-based predictions.

2.1. SHAP (SHapley Additive exPlainations)

Lundberg and Lee [79] suggested using SHAP to elucidate the ML predictions based on game theory. For example, inputs are referred to as players while the prediction becomes the payout. SHAP determines the contribution of each player to the game. Lundberg and Lee [79] have introduced several versions of SHAP (e.g. DeepSHAP, Kernel SHAP, LinearSHAP, and TreeSHAP) for specific ML model categories. For example, Tree-SHAP is used in the present study to explain the ML predictions. It uses a linear explanatory model and Shapley values (Eq. 1) to estimate the initial prediction model.

$$h(z^{'}) = \varnothing_o + \sum_{i=1}^{N} \varnothing_i z^{'}_i \tag{1}$$

In Eq. 1, 'h' represents the explanation model whereas $z^{'}$ denotes the basic features. N is the maximum size of collation and $\varnothing$ denotes the feature attribution. Lundberg and Lee [79] recommend Eqs. 2 and 3 to compute the attribution of each feature.

$$\varnothing_{\mathrm{i}} = \sum_{\mathrm{K}\subseteq \mathrm{M}\{\mathrm{i}\}} \frac{|\mathrm{K}|!(\mathrm{N}-|\mathrm{K}|-1)!}{\mathrm{N}!}[\mathrm{g_x}(\mathrm{K}\cup\{\mathrm{i}\}) - \mathrm{g_x}(\mathrm{K})] \tag{2}$$

$$\mathrm{g_x}(\mathrm{K}) = \mathrm{E}[\mathrm{g}(\mathrm{x})|\mathrm{x_K}] \tag{3}$$

The term 'K' represents a subset of the features (input). 'M' denotes the set of all inputs. $E[g(x)|x_K]$ represents the expected value of the function on subset K. In addition, this study employs Scikit-learn, NumPy, matplot, pandas, and Shap libraries for the implementation.

3. ML Algorithms

Fundamentals of ML algorithms used in this study are briefly presented here.

3.1. Decision tree regressor

A decision tree can be interpreted as the basic form of tree-based algorithms. It is an effectual algorithm that serves classification and regression. Breaking a complicated task into several simpler tasks is the fundamental process of a decision tree. Thus, it can assist to interpret solutions more conveniently [81]. The tree structure is hierarchical from root to leaves [82]. Decision tree possesses considerable transparency and is formed based on rational postulations [83]. The training process of a decision tree is performed based on multiple regression and recursive splits. These splits take place until the end criteria are achieved [84]. Each leaf node represents a simple regression model. The pruning sequence is performed subsequently to improve generalization while reducing the complexity of the model. The sum of square error (SSE) is generally considered to perform recursive splits (refer to Eq. 4). For instance, per each split, the response (z) is categorized into two groups, G_1 and G_2. Later, the tree examines whether the predictor variable x is lesser or higher than the split threshold value.

$$SSE = \sum_{i\in \quad G_1} (z_i - \overline{z_1})^2 + \sum_{i\in G_2} (z_i - \overline{z_2})^2 \tag{4}$$

$\overline{z_1}$ and $\overline{z_2}$ denotes the average values of the response variable of each group (G1, and G2). Similar to classification, the tree generally intends to perform the split, minimizing the SSE value for a split.

3.2. Extra tree regressor

The extra tree is a meta-estimator, which can form numerous disordered decision trees on sub-samples. It is an extended version of random forest. According to related work, the extra tree is less likely to be over-fitted to a sample [85]. Similar to the random forest, it

employs random sub-samples of features to the training process of the base estimator [86]. The whole data set is simultaneously employed for the training process of an extra tree. However, the bootstrap replica is used to train a random forest model.

Despite the size of the tree, Geurts [85] reports that the extra tree is less complex compared to a random forest. Besides, the extra tree is faster than both tree-bagging and random forest methods. Random cut points used in the extra tree make the variance lower in contrast to the weak randomization sequence of other approaches.

For regression, relative variance reduction is used whereas the score is expressed in Eq. 5. The terms T_i and T_j represent subsets of cases from T that correspond to the outcome of a split s.

$$Score_R(s,T) = \frac{var\{z|T\} - \frac{|T_i|}{|T|}var\{z|T_i\} - \frac{|T_j|}{|T|}var\{z|T_j\}}{var\{z|T\}} \tag{5}$$

3.3. AdaBoost regressor

Optimally combining weak learners to obtain a precise prediction can be referred to as boosting [87]. Among boosting techniques, AdaBoost (Adaptive boosting) is an efficient learning technique that was applied for generic regression algorithms, improving performance [88,89]. It adopts an iterative sequence such that a new regressor is formed during each step on the training sample. Weights associated with data points are modified using the resolution of the previous learning sequence. AdaBoost forms numerical models by changing the variation of a dataset. When samples are selected based on their precision, each misprediction is boosted by an identical amount.

3.4. Extreme gradient boosting regressor (XGBoost)

Extreme Gradient Boosting regressor (XGBoost) is a gradient boosting integration algorithm [90]. It performs as a second-order Taylor series expansion over objective function and combines an additional regularization term that leads to achieving a prime solution. The objective function in terms of loss and regularization can be expressed as given in Eq. 6.

$$\text{Objective} = \sum_{j=1}^{N} l(y_j, \widehat{y_j}) + \sum_{j=1}^{K} \Omega(f_j) \tag{6}$$

The first term inside sigma operator ($\sum_{j=1}^{N} l(y_j, \widehat{y_j})$) is a loss function that represents the difference between predicted and actual values. The term inside second sigma operator ($\sum_{j=1}^{K} \Omega(f_j)$) is a regularization term that decides the complexity of the XGBoost model.

3.5. LightGBM regressor (LGBM)

LGBM is a decision tree formed on a gradient training structure. It also involves boosting [91]. Employing histogram-based algorithms is observed to be a unique feature of LGBM with respect to XGBoost. Thus, it can accelerate training sequence, while reducing memory utilization. Given that LGBM is based on a decision tree (moderately weaker model), segmentation precision is not quite an important aspect [92]. A coarse regularization process is present to avoid the chance of over-fitting. LGBM consists of some improved features compared to the conventional decision tree format. For instance, an inefficient strategy of level-wise growth is usually adopted in decision trees which results in additional memory consumption. However, leaf-wise growth in LGBM is a prominent strategy to reduce error, employing the same segmentation period.

3.6. Laplacian kernel ridge regression (LKRR)

Kernel ridge regression substantially accounts for the non-linear behavior of a system. Similar to Euclidean space, a distance measure is used to define similarity between any two points. Mannodi-Kanakkithodi et al. [87] provide a detailed description of basic formulation in kernel ridge regression. Several kinds of kernels are available such as Gaussian, Laplacian, and Polynomial. Laplacian, which is a radial basis function kernel was employed in this research. In addition, Mannodi-Kanakkithodi et al. [87] explained several similarities between Gaussian kernel and Laplacian kernel, and more information on Laplacian kernel can be found in their work.

4. Performance measurements

It is important to perform an uncertainty analysis to evaluate the performance of ML regression models [93]. Four commonly used indices are used in this study, including Mean Absolute Error (*MAE*), Coefficient of Correlation (*R*), Coefficient of Determination (R^2), and Root Mean Square Error (*RMSE*). Mathematical explanations for these four indices are given in Eqs. 7–10. The coefficient of determination and correlation coefficient decide how well predictions fit experimental data and linearity between predictions and experimental values, respectively. Both indices being closer to 1 explicate a strong relationship. In addition, MAE quantifies absolute residual and RMSE determines the standard deviation of the error component. Both training and testing sequences are performed based on R^2, and overall predictions are verified using all indices to confirm the consistency of regression models.

Table 1
Mixed design data (Source: UCI repository).

Type	Variable	Maximum value	Minimum value	Mean	Standard deviation	CoV (%)
Independent	Cement	540	102	281.1	104.5	37
	Blast Furnace slag	359.4	0	73.8	86.2	117
	Fly ash	200.1	0	54.1	63.9	118
	Water	247	121.8	181.5	21.3	12
	Super plasticizer	32.2	0	6.2	5.9	95
	Coarse aggregate	1145	801	972.9	77.7	8
	Fine aggregate	992.6	594	773.5	80.1	10
	Age (Days)	365	1	45.6	63.1	138
Dependent	Compressive strength (MPa)	82.6	2.33	35.8	16.7	47

$$R^2 = \frac{\sum_{i=1}^{N}(CS_P - \overline{CS}_E)^2}{\sum_{i=1}^{N}(CS_E - \overline{CS}_E)^2} = \frac{Model\ \ Sum\ \ of\ \ Squares}{Total\ \ Sum\ \ of\ \ Squares} \tag{7}$$

$$R = \frac{N\sum_{i=1}^{N}(CS_P.CS_E) - (\sum_{i=1}^{N}CS_P.\sum_{i=1}^{N}CS_E)}{\sqrt{(N\sum_{i=1}^{N}{CS_E}^2 - (\sum_{i=1}^{N}CS_E)^2).\sqrt{N\sum_{i=1}^{N}{CS_P}^2 - (\sum_{i=1}^{N}CS_P)^2)}}} \tag{8}$$

$$MAE = \frac{\sum_{i=1}^{N}|CS_P - CS_E|}{N} \tag{9}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(CS_P - CS_E)^2}{N}} \tag{10}$$

(Subscripts '*P*' and '*E*' denotes predicted and experimental, respectively. $\overline{CS}_E$ refers to mean value of compressive strength of parent sample.).

5. Materials and methodology

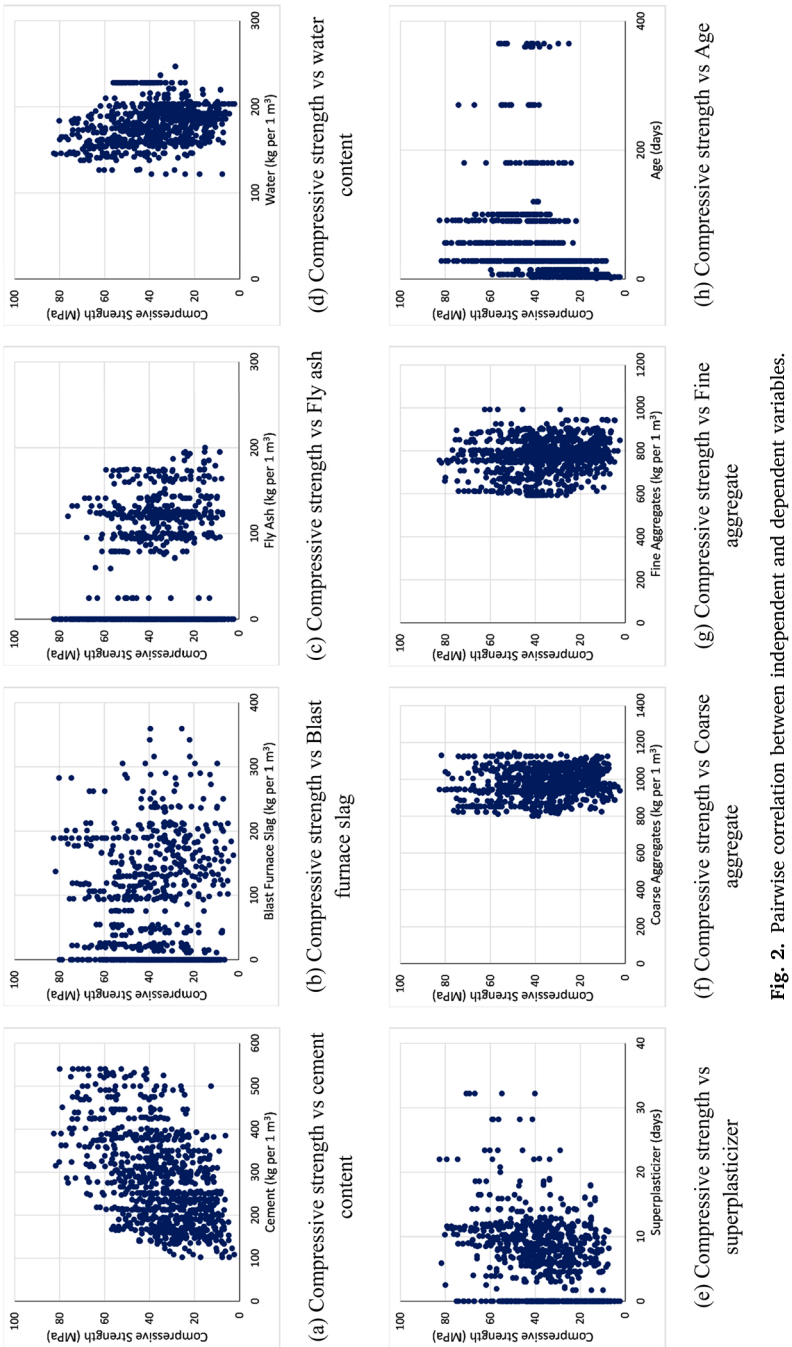
5.1. Mix design data

Public repositories assist to advance the involvement of ML in various disciplines. However, the reliability of the source should be validated before the usage of these data. The mixed design data from the UCI (University of California Institute) ML repository were used for this study [57,94–96]. Macia and Bernado-Mansilla [97] claim that it is one of the trusted data sources among the ML community. The UCI database was established in the early nineties and contains a broad range of data (e.g. categorical, numerical). In addition, it provides data for several types of analysis including regression, classification, clustering, etc. The mixed design dataset, which consists of 1030 instances was chosen for the ML process. Latter sections critically analyze the resolution of the dataset. It consists of eight independent variables and one dependent variable (See Table 1). Mass of all constituents is given in kilograms in 1 m^3 of the mixture.

According to Table 1, mix design data has various additives other than conventional components of concrete mix. Coefficient of Variation (CoV%) values show that variation of coarse aggregates, water content, and fine aggregates are notably lower than other materials. However, the mean of coarse aggregate is significant. Blast Furnace Slag, Fly Ash, and Age display moderately higher dispersion among the whole dataset. It is noted that compressive strength corresponds from 1 day to 365 days are available in values ranging from 2 Mpa to 83 Mpa.

It is required to justify the need for ML algorithms to exploit relationships. Hence, relationships between various independent variables and compressive strength were plotted and showcased in Fig. 2. Two linear relationships can be assumed for Fig. 2a (compressive strength vs the cement content) and Fig. 2e (compressive strength vs superplasticizer). However, they are not solid as there are many outliers and the coefficient of determinations for trend lines are 0.25 and 0.13, respectively. However, all other plots showcase a non-linear behavior. Hence, a complex relationship rather than linear mapping is highly required to capture variation and interaction. Even though many variables exhibit non-linear behaviour with compressive strength, they can have significant feature importance to the prediction. For example, water content holds greater importance on compressive strength whereas an increase in water content reduces the compressive strength. However, Fig. 2 provides a poor relationship between water content and compressive strength. Therefore, we suggested taking all eight features as respective input features. Because SHAP will provide original feature importance regardless of the linearity observed in Fig. 2.

Subsequently, the data set was randomly separated into 60% and 40% for the training and testing (*out-of-bag data*) process. Each ML algorithm was locally optimized using respective hyper-parameters through a grid search.



(a) Compressive strength vs cement content

(b) Compressive strength vs Blast furnace slag

(c) Compressive strength vs Fly ash

(d) Compressive strength vs water content

(e) Compressive strength vs superplasticizer

(f) Compressive strength vs Coarse aggregate

(g) Compressive strength vs Fine aggregate

(h) Compressive strength vs Age

Fig. 2. Pairwise correlation between independent and dependent variables.

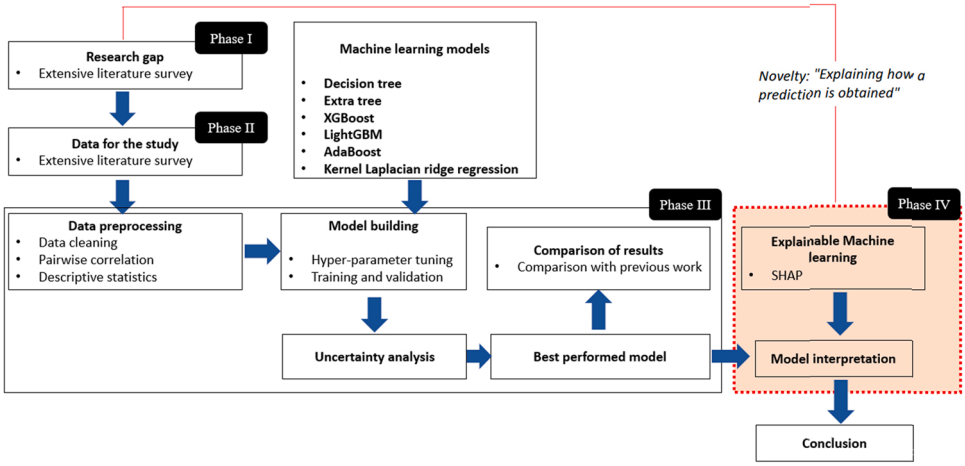


Fig. 3. Methodology for the research study.

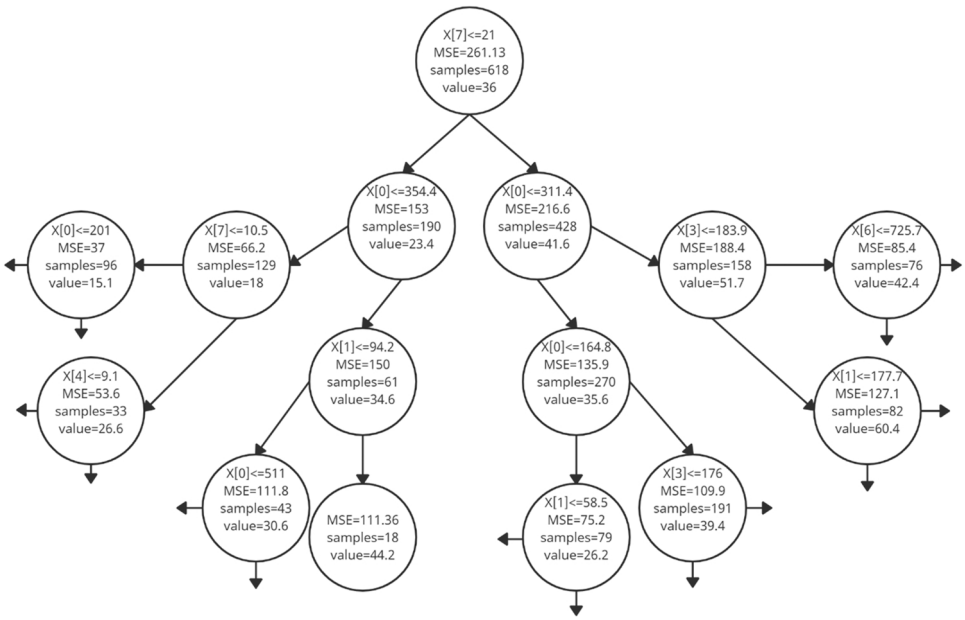


Fig. 4. First four layers of decision tree structure formed at depth = 8.

5.2. *Workflow of the study*

Fig. 3 illustrates the workflow of this study. In contrast to previous studies, we included phase IV as the novelty that explicitly describes how a specific prediction was obtained (underlying reasoning).

5.3. *Functioning of a decision tree regressor*

The use of a decision tree-based structure provides explicitness in contrast to ANN's. Since the decision tree clustering process mimics the human thinking process, it is convenient for even a non-technical community to understand the behavior. However, many advanced tree-based structures are moderately complex compared to the basic decision tree. Nevertheless, all tree-based models used in this study are decision tree-based models. Therefore, it is imperative to describe the operating sequence of a decision tree regressor. Fig. 4 depicts the first four layers formed in the decision tree of the current study. The number in front of the letter X represents the variable assigned in the following order. (0 – Cement content; 1- Blast Furnace Slag; 2-Fly Ash; 3- Water; 4 - Superplasticizer; 5 – Coarse Aggregate; 6 – Fine aggregate; 7 - Age).

Initially, the tree decided, age is the dominant parameter. Because, compressive strength values had a significant range of values

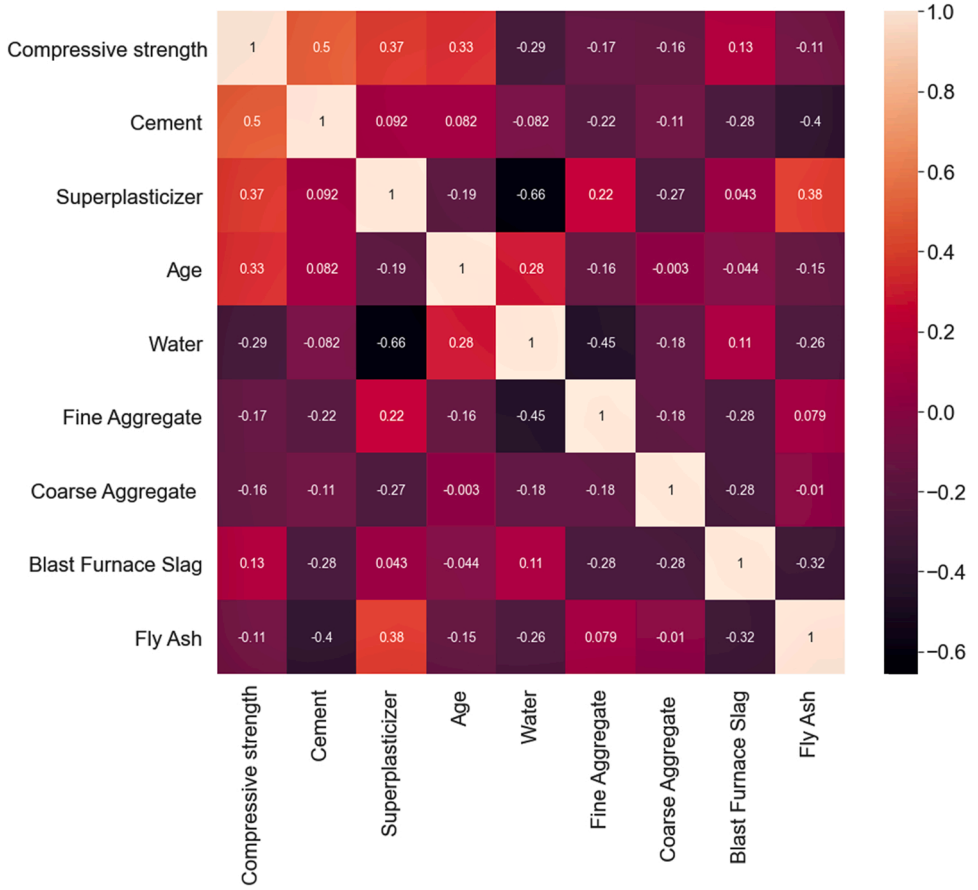


Fig. 5. Pairwise correlation matrix.

after 14 days going up to 365 days, according to UCI data. Accordingly, the decision tree separated 618 samples into two groups at the subsequent layer. Afterward, the tree considered cement content has a considerable variation in subsample and it divided each sample according to limiting values provided in the second layer. Progressively, tree structure became more expanded at the third layer, tree concerned several variables (Cement, Water, Blast Furnace Slag, and Age) as dominant variables within subsamples. Splitting was performed based on mean squared error (MSE). Interestingly, the tree separates a large variation from the sample by gradually reducing MSE at each layer. Thus, it continues splitting samples until end criteria are met (no significant discrepancies exist within the cluster). All tree-based algorithms employed in this study are decision tree-based architectures. However, each model possesses various unique implementations such as bagging and boosting, generating a unique output.

6. Results and discussion

6.1. Correlation results of the pairwise matrix of data sample

Fig. 5 depicts the correlation matrix of the data set. Accordingly, Superplasticizer, Cement, Blast furnace slag, and Age have a positive correlation with the compressive strength of concrete, while cement content shows the highest positive correlation of 0.5. Negative correlation values between compressive strength and variable show moderately lower values compared to positive values. The highest negative value observed between compressive strength and water content is − 0.29. In addition, there are several notable correlation values found between independent variables. For instance, the correlation between fly-ash and superplasticizer content is 0.4 and it becomes − 0.4 between fly ash and cement. In addition, a fairly good negative correlation between water content and superplasticizer (−0.66) can be observed. Furthermore, a negative correlation exists between fine aggregates and water content.

6.2. Hyperparameter optimization

Hyperparameters should be tuned in order to optimize the performance of ML models. The authors obtained default hyperparameter values from the Scikit library. Subsequently, the grid search method was used to optimize hyperparameters for each model. Grid search will separate the domain of hyperparameters from a discrete grid. For each ML model, grid search creates numerous

Table 2
Summary of hyperparameter optimization (Highlighted parameters were optimized using a grid search method whereas remaining parameters are default values).

Indice/ Hyperparameter	XGBoost	AdaBoost	Extra tree	Decision tree	LKRR	LGBM
Tree depth	5	9	9	9	-	9
Estimators	-	20	100	-	-	-
Alpha	-	-	-	-	0.0051	-
Gamma	-	-	-	-	0.0009	-
Minimum samples leaf	-	-	2	2	-	-
Minimum sample split	-	-	2	2	-	-
Gamma	0.0001	-	-	-	-	-
Learning rate	0.1	-	-	-	-	0.1
Reg_Alpha	0.0001	-	-	-	-	0.0001

models by using combinations of hyperparameters. Finally, those models will be evaluated to obtain the optimum value for hyper-parameters. It is noteworthy that the accuracy of tree-based models is strongly dependent on the depth of the tree and the number of estimators. In such a case, other parameters can be omitted from grid search since it is an exhaustive process [98]. The optimized and default hyperparameters are mentioned in Table 2.

Fig. 6 showcases the hyper-parameter optimization (tree-depth, and the number of estimators) of tree-based models. The simplest form of a tree-based algorithm, the classic decision tree was not efficient at lower depths as shown in Fig. 6. However, beyond the depth of 5, R^2 exceeds 0.7 for both testing and training phases. Beyond a depth of 8, R^2 values and the gap between training and testing accuracies remained constant. A maximum R^2 value of 0.9 was obtained at a depth of 8 and the corresponding testing R^2 value was 0.8.

Subsequently, training and testing of extra tree regressor appeared to be better compared to decision tree regressor, achieving higher stalled R^2 values beyond 0.85. However, a direct comparison between depth and the number of estimators is impractical as a result of the unique working principle. At a low number of estimators, R^2 has exceeded 0.7 for both phases. The gap observed between phases was visible for extra tree regression as well. Beyond estimators = 10, extra tree yields stable values of 0.97 and 0.87 for learning and testing, respectively.

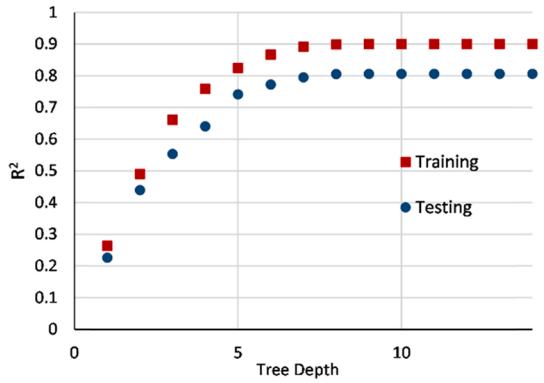
In addition, the same approach was relatable for AdaBoost regressor. Boosting algorithm has obtained moderately good learning and testing processes in contrast to decision tree algorithm. However, when the number of estimators increased beyond 20, training and testing accuracies tend to descend and the gap between the two phases has reduced comparatively. Generally, training accuracy and testing accuracies are acceptable, when they are nearly equal. Higher training accuracy might lead to data memorization, whereas higher testing accuracies lead to under-fitting occasions [99]. However, in terms of accuracy, the extra tree does predictions better than the decision tree and AdaBoost algorithm. Here, the base of AdaBoost algorithms is a decision tree. It does not rule out employing AdaBoost to enhance the performance of the remaining algorithms. More importantly, AdaBoost has notably improved classical decision tree predictions.

Fig. 6d and e show the training and testing accuracies for the XGBoost and LGBM algorithms. The gap between training and testing has reduced compared to the decision tree and extra tree regressor. Moreover, XGBoost reaches R^2 beyond 0.9 after tree depth = 2. Thus, it explains the superior performance of the XGBoost algorithm to handle data even at lower depths.

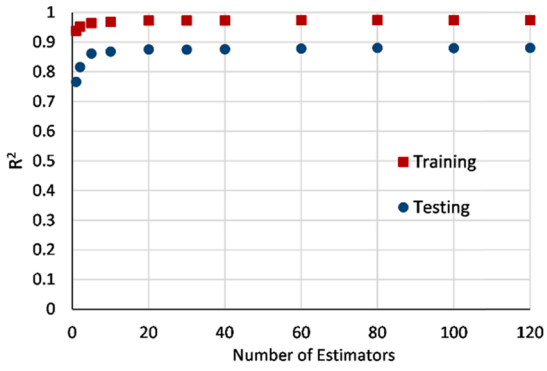
The optimization process of the Laplacian kernel was reasonably different from tree-based ML models. There are two hyper-parameters (called 'α' and 'ϒ') whereas training and testing accuracies are functions of those two estimators. Fig. 7 illustrates the spatial variation of R^2 of each phase with the above estimators. Accordingly, at lower values of both alpha (α) and gamma (ϒ), The Laplacian kernel has obtained acceptable accuracies, which are comparable with classical tree-based regressors. Interestingly, the training process easily reaches an R^2 of 0.99 for a certain range of α and ϒ. Parameters α and ϒ were selected as 0.0051 and 0.0009 for overall predictions, respectively.

6.3. *Comparison of overall predictions*

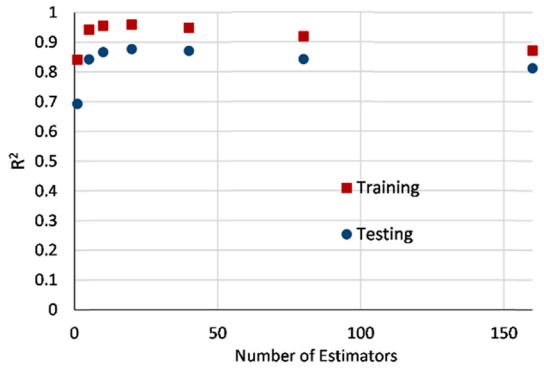
Fig. 8 shows overall predictions obtained from the previous regression methods. Evidently, the Pearson correlation coefficient (R) of predictions was acceptable ($R > 0.9$ for each model). Since the decision tree itself is a weak learner, it provides relatively weaker predictions compared to the rest of the ML models. The decision tree seems to have more inconsistencies compared to experimental results. The deviations greater than 20% are present for lower and higher values of compressive strength.



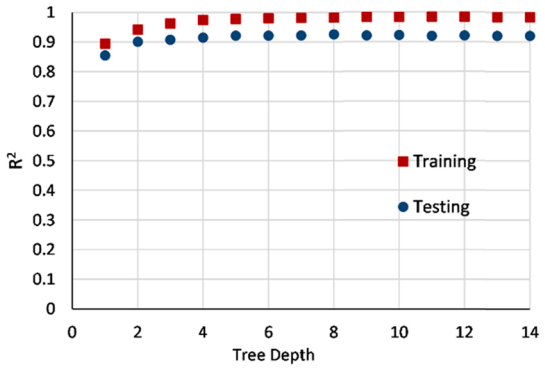
(a) For Decision tree regressor



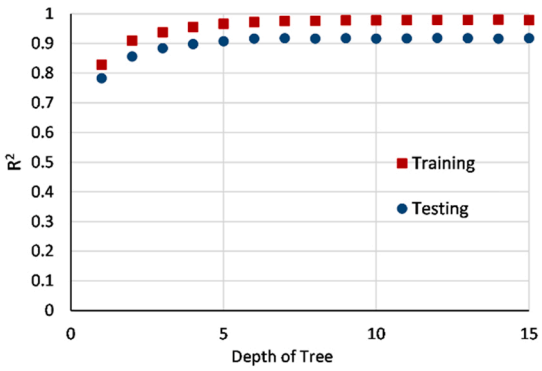
(b) For Extra tree regressor (Tree depth = 9)



(c) For AdaBoost regressor (Tree depth = 9)



(d) For XGBoost regressor



(e) For LGBM regressor

Fig. 6. Hyperparameter optimization of tree-based models.

The AdaBoost algorithm could boost the performance of the decision tree, enhancing the Pearson correlation coefficient from 0.93 to 0.96. Thus, boosting algorithms provided an improved version of the classic decision tree. Linearity was acceptable and AdaBoost predicted higher range compressive strength values almost within the 10% error limit. When extra tree predictions are considered, it has acquired an R of 0.97, which was comparable with XGBoost, LGBM, and LKRR. Extra tree improved both lower range and higher range of compressive strength predictions compared to Adaboost.

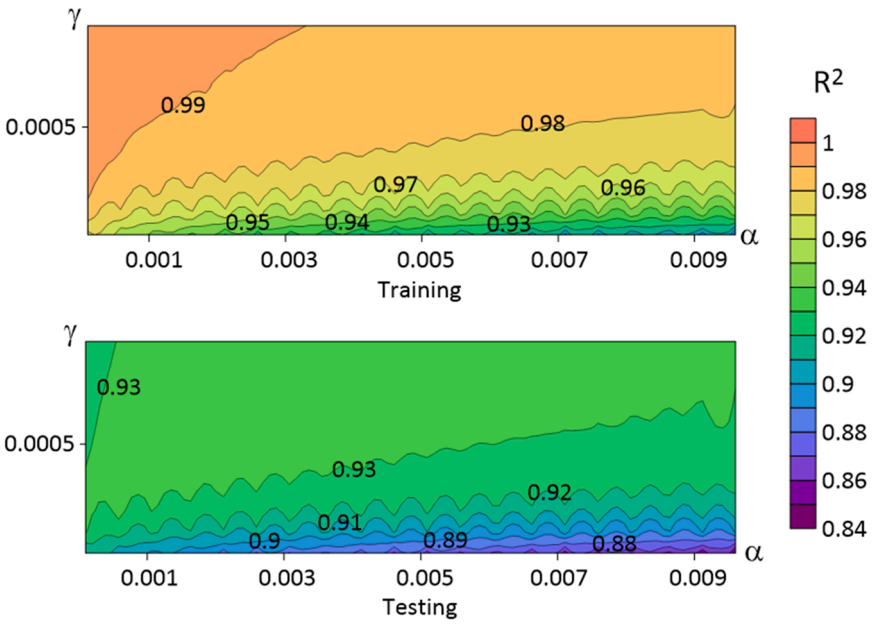


Fig. 7. Hyper-parameter optimization of LKRR.

Interestingly, the LKRR efficiently predicted compressive strength values at an R = 0.98. Hence, LKRR possesses the ability to compete with conventional ML algorithms. It was observed that the predictions lie within the 20% error range for most of the time. Except for some slight inconsistencies, prediction accuracy was slightly higher than XGBoost, thus, showing consistent linearity towards a higher range of compressive strength.

The remaining tree-based boosting algorithms XGBoost and LGBM showcased their potential to predict the compressive strength of concrete at an R of 0.98. It appeared, both models had error limits of less than 20% error limit. Hence, it convinced the ability of tree-based algorithms and LKRR to be applied for compressive strength predictions. However, the former illustration (Fig. 8) cannot explain the overall uncertainty of the model. Therefore, Table 3 provides the summary of uncertainty analysis for optimized models.

Accordingly, all models perform predictions with acceptable accuracy. AdaBoost possessed greater adaptability at a lower number of estimators compared to the extra tree regressor. Corresponding MAE and RMSE values were obtained as 3.44, and 4.61 for AdaBoost and 2.87, and 4.3 for extra-tree, which are comparable to each other. The highest RMSE and MAE were recorded for decision tree regressor were 6.26, and 4.58, respectively. MAE and RMSE indices that have been obtained for LGBM and XGBoost were nearly equal and lower in contrast to AdaBoost and Extra-tree regressor. Fascinatingly, LKRR exhibited the lowest error indices (MAE=1.83, RMSE=3.15), showcasing superior flexibility with lower inconsistencies. However, equal R values were computed for LKRR, XGBoost, and LGBM models.

XGBoost model formed on a lower tree depth reached higher accuracy. In terms of optimization, XGBoost consists of superior performance over the LGBM model. Hence, XGBoost and LKRR appeared to be suitable for compressive strength predictions from the present analysis. However, during the process, it was identified that XGBoost assists model interpretability sequence compared to LKRR. Therefore, we continued the explanation using the XGBoost model as the best-performed model.

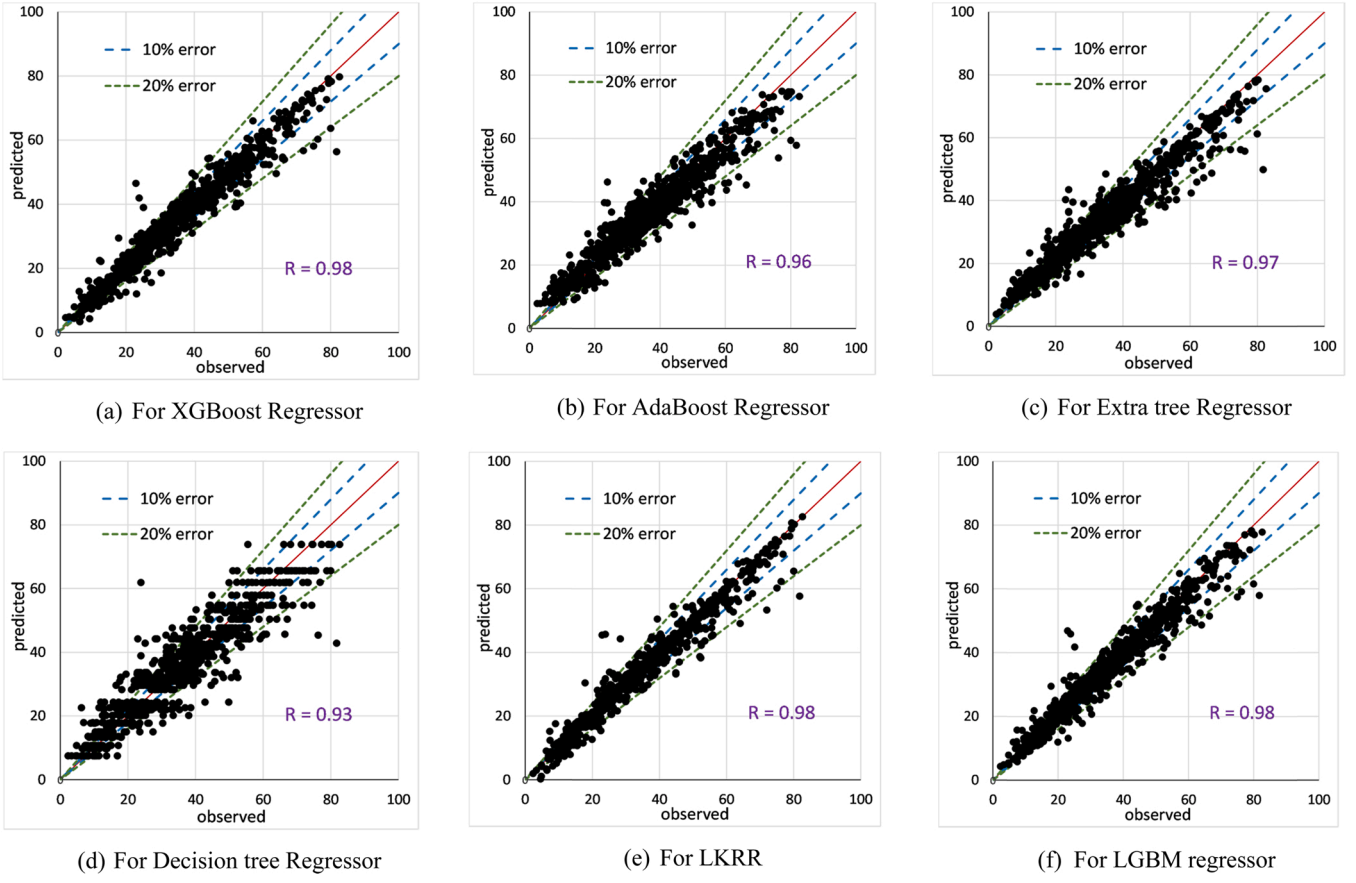
6.4. *Comparison with related work*

Table 4 provides a summary of several studies conducted on ML involvement in predicting compressive characteristics of concrete. Many previous studies involved ANN and SVM methods. Accordingly, in the present study, a better accuracy was achieved compared to many previous studies. For example, the XGBoost algorithm reached an R of 0.98, exhibiting the astonishing performance of gradient boosting trees. As well, ridge regression with LKRR had a similar performance, despite its inferior explainability. Moreover, the decision tree has reached R= 0.93, even though it is considered a weaker learner (See Fig. 9).

In terms of optimization, Tree-based algorithms are user-friendly compared to ANN. The choice of optimum architecture for an ANN is a complicated task that requires significant expertise. Besides, tree-based algorithms themselves form the evolved structure that appeared to be convenient in contrast to an ANN.

6.5. *SHAP interpretations*

Evidently, previous studies were unable to involve ML interpretability methods to explain how a prediction is obtained. On the other hand, when the models become complex, a post-hoc explanation is required. Therefore, we involved SHAP explanations to reveal



(a) For XGBoost Regressor (b) For AdaBoost Regressor (c) For Extra tree Regressor

(d) For Decision tree Regressor (e) For LKRR (f) For LGBM regressor

Fig. 8. Comparison of overall ML predictions with respect to UCI data.

Table 3
Uncertainty analysis of ML models.

Indice	XGBoost	AdaBoost	Extra tree	Decision tree	LKRR	LGBM
R	0.98	0.96	0.97	0.93	0.98	0.98
R^2	0.95	0.93	0.94	0.86	0.96	0.95
MAE	2.38	3.44	2.87	4.58	1.83	2.32
RMSE	3.60	4.61	4.30	6.26	3.15	3.60

Table 4
Summary of several related works on ML interference on compressive strength predictions.

Citation	Sample size	ML models	Performance indices					
[28]	1030	ANN	R	0.95	RMSE	5.0	MAPE	10.9
		SVM		0.94		5.6		12.8
		MART		0.95		4.9		13.9
[100]	1133	GEP	R	0.91			MAE	5.2
		FA-LSSVR		0.94				3.9
		MFA-ANN		0.95				3.4
[101]	210	ANN	R	0.92	RMSE	2.4	MAE	2.0
		ANFIS		0.94		2.3		1.7
[102]	1761	BPNN	R	0.98			MAE	5.0
		ELM		0.97				4.3
		ANFIS		0.95				4.2
		SVM		0.89				6.0
		M5		0.97				4.0
		ECSO-SVM		0.97				4.0
[30]	239	ANN	R	0.87	RMSE	6.7	MAPE	13.4
		SVM		0.89		6.1		12.0
		FA-LSSVR		0.93		4.9		9.8
[103]	324	ELM	R	0.99	RMSE	0.8	MAE	0.6
		BP		0.99		0.9		0.7
[104]	180	ANN	R	0.82	RMSE	3.2		
		GA-ANN		0.90		2.2		
		ANFIS		0.97		1.5		
Present study	**1030**	**XGBoost**	**R**	**0.98**	**RMSE**	**3.6**	**MAE**	**2.4**
		AdaBoost		**0.96**		**4.6**		**3.4**
		Extra tree		**0.97**		**4.3**		**2.9**
		Decision tree		**0.93**		**6.3**		**4.6**
		LKRR		**0.98**		**3.2**		**1.8**
		LGBM		**0.98**		**3.6**		**2.3**

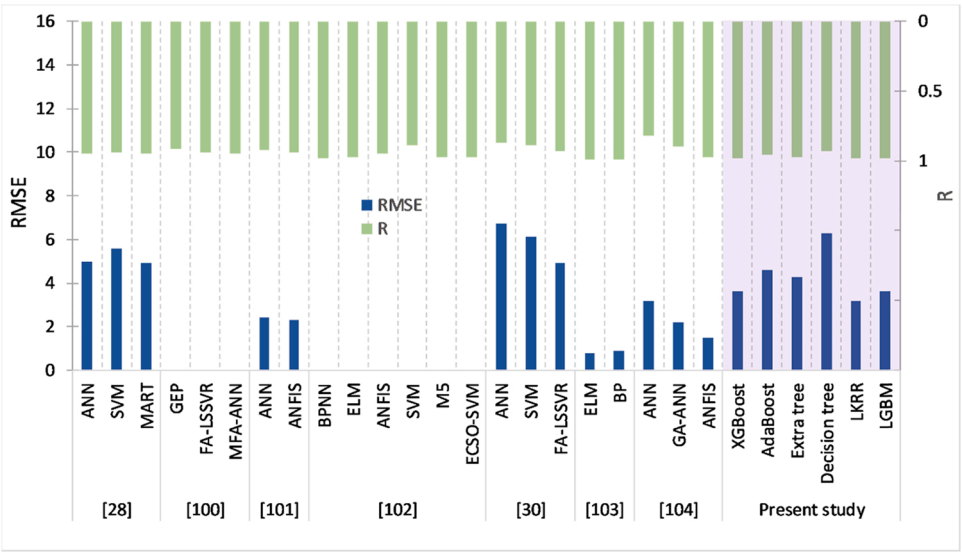


Fig. 9. Comparison of R, and RMSE of predictions with related work.

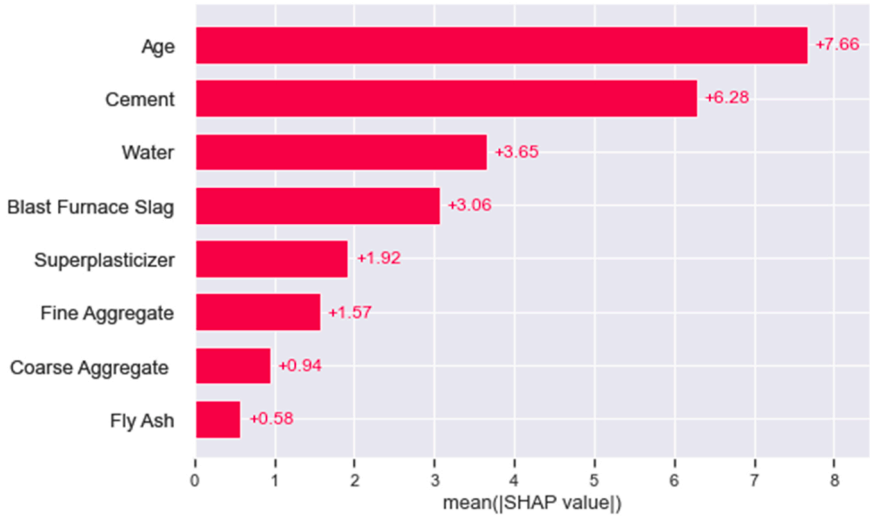


Fig. 10. Mean absolute SHAP values of XGBoost model.

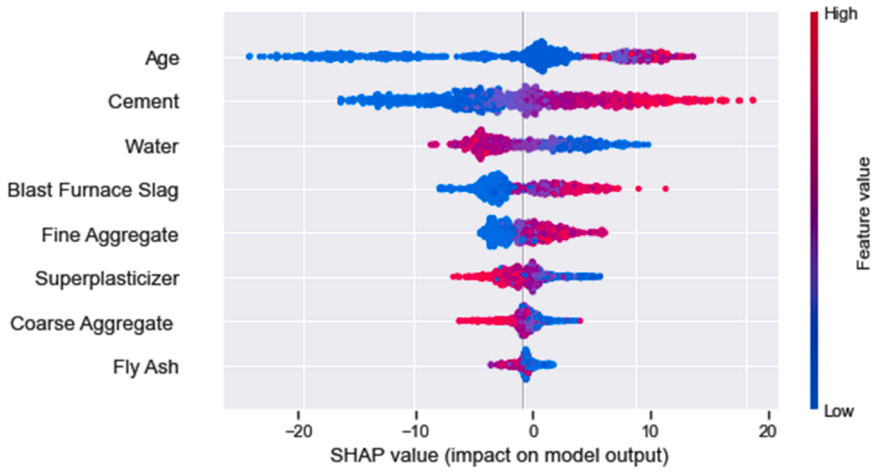


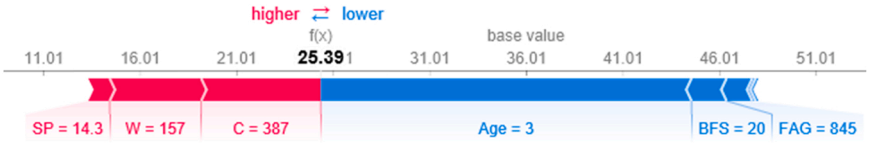
Fig. 11. SHAP global explanation on XGBoost model.

the underlying reasoning behind predictions, the interaction of each variable towards the outcome, and explain a given instance.

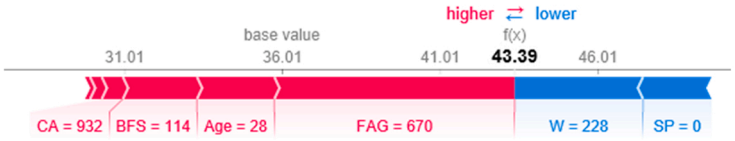
As previously mentioned, the XGBoost model was continued for SHAP explanation. Figs. 10 and 11 depict mean absolute SHAP values and global explanations of the XGBoost model, respectively. Feature values are mainly illustrated using a color code such that blue colors for lower feature values and red for higher feature values.

Age showed the highest mean SHAP value on the output which is 7.66. Next, cement content appeared to be the dominant parameter that had obtained 6.28. Surprisingly, the feature importance of age was not noticed from the pairwise correlation (refer to Fig. 2). Accordingly, SHAP provided the mean SHAP values according to their impact on the model output within a few seconds. The effect of fly ash had less influence on the overall output (0.58). SHAP confirmed that the effect of superplasticizer is greater compared to the effects of fine and coarse aggregates. Further, the mean SHAP values can be described using their original feature importance as shown in Fig. 11.

According to Fig. 11, age has a much influence on compressive strength as XGBoost identified. Lower feature values of age decrease compressive strength and higher values have a comparable positive effect on compressive strength. Secondly, cement content appeared to be a dominant variable which is similar to the influence of age. When the cement content is low, compressive strength decreases and vice versa. Higher cement content leads to a considerable positive impact on the compressive strength of concrete. However, water content had a completely reversed influence compared to former variables. Similar to experimental observations, a higher water content drops the compressive strength of concrete. Both blast furnace slag and superplasticizer had a similar influence on compressive strength despite their magnitudes. Lower fine aggregate content has a notable effect on compressive strength while lower content has a concatenated negative effect on the outcome. The effects of blast furnace slag, fine aggregates, and superplasticizer are comparable with each other. A slightly pronounced variation was observed from coarse aggregates and fly ash content on compressive strength.



(a) Instance force plot 1: Superplasticizer (SP); Water (W); Cement (C); Blast furnace slag (BFS); Fine aggregates (FAG)



(b) Instance force plot 2: Superplasticizer (SP); Water (W); Coarse aggregates (CA); Blast furnace slag (BFS); Fine aggregates (FAG)

Fig. 12. SHAP force plot on selected instances for local interpretation.

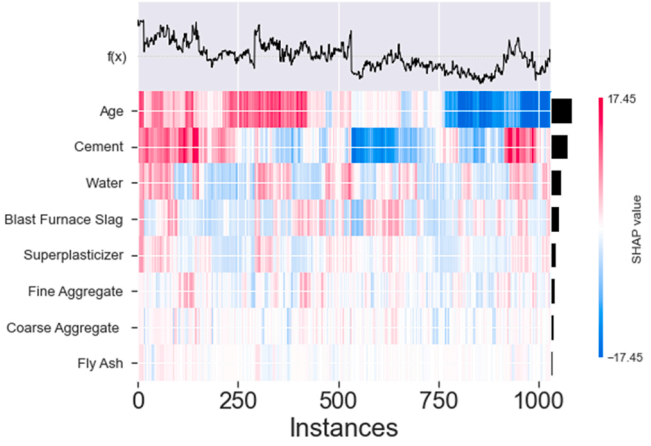


Fig. 13. SHAP heat map plot.

However, we noticed, coarse aggregate content varies from 801 to 1000 on provided data set (lovest CoV%). Therefore, SHAP labels values near 800 as low values even if it is a considerable value. Hence, the importance of feature range is also an important factor in model explanation.

Moreover, SHAP explained adheres to the relationship observed in the data preprocessing stage as well. For example, according to the correlation matrix of the parent dataset shown in Fig. 5, water, fine aggregates, fly ash, and coarse aggregate content have a negative correlation with compressive strength. Thus, it validates ML interpretation on XGBoost decisions, complying with experimental observations. Further, the SHAP explanation quantifies interference of each variable within the whole range which was absent in the correlation matrix. Apart from global interpretation, Fig. 12 illustrates selected two instances for local explanations (particular prediction) obtained from the SHAP force plot.

According to Fig. 12a, lower feature values of age decrease the compressive strength. The bolded value (base value + SHAP values) represents the prediction obtained during the ML training sequence. Thus, it explains variables have unique contributions and interactions towards outcome at any given point. For instance, cement content (387) has a positive impact on the compressive strength of concrete. Further, the water content of 157 has also a positive impact on compressive strength compared to Fig. 12b whereas the water content of 228 has a negative influence on compressive strength output. It was interesting to observe that SHAP identified that the strength decreases without any superplasticizer as depicted in the right-hand corner of Fig. 12b. Complying with Fig. 12b, a elucidates that the superplasticizer content has a positive effect on compressive strength at 3 days. Comparing these two instances, lower blast furnace slag content (20) pushes compressive strength towards lower values while increased blast furnace slag content (114) forces

compressive strength base values towards the positive side. As well, increased fine aggregates content (845) has a negative influence on the compressive strength of concrete.

The heat map (refer to Fig. 13) helps to identify the SHAP value behind the same prediction at different instances. It uses hierarchical clustering based on the explanation similarity. Accordingly, lower features (from superplasticizer to fly-ash) have an inferior SHAP value on the model output. In contrast, the f(x) plot shows the instances where different SHAP values are observed for the same f (x) value. Compared to the global interpretation, cement content shows a higher negative SHAP value for the instances from 500 to 750. For example, the effect of age is positive for the instances from 0 to 500 and the effect becomes reversed from instance 750. Such strongly highlighted variation was observed for cement as well.

Overall, SHAP provided various detailed resolutions on each instance and global interpretation of the model. SHAP interpreted the inner-working approach of the ML model, without expecting a user to involve. Hence it assists in decision-making even for a non-technical audience. Impressively, the local explanations achieved the goal of explaining the reasoning behind a prediction. Therefore, we suggest engaging ML interpretability methods to support decision-making criteria in civil engineering applications. Eventually, it will build trust between domain experts on ML techniques.

7. Limitation of the study

- Although the study involved SHAP for the interpretations, there are many post-hoc explanatory models which are designed for the same task. Therefore, we suggest comparing the interpretations obtained through different explanation methods. For example, the feature importance values calculated using SHAP might not the same as other explanation methods.
- Furthermore, the study focussed on the compressive strength of concrete. However, the study is applicable for other strength characteristics as well. ML together with post-hoc explainable methods will help to predict other strength characteristics and explain the underlying reasoning. Accordingly, the influencing parameters can be identified which are required for the design stage.

8. Conclusions

This research investigated the use of interpretable (glass box) ML to predict the behavior of compressive strength of concrete varying its constituents. The lack of explainability of traditional ML has been successfully overcome by employing the ML interpretability method. In addition, we proposed tree-based and ridge regression models instead of ANN and SVM approaches. The comparison revealed Laplacian Kernel Ridge Regression (LKRR), Light Gradient Boosting Method (LGBM), and Extreme Gradient Boosting (XGBoost) provide superior performance in terms of predictions ($\mathbf{R = 0.98}$). XGBoost possesses better adaptability to prediction criteria even at lower tree-depth. Fascinatingly, SHAP interpretations confirmed XGBoost predictions follow the theoretical behavior of compressive strength of concrete. SHAP quantifies interaction between each parameter and contribution towards an overall outcome, convincing the underlying nature of the ML model. Besides, SHAP provides the causality and the inner-working approach of ML models to advance the end-user's trust in predicting the compressive strength of concrete. Finally, the results of this research offer new knowledge to the research community and in addition, showcase its applicability in civil engineering applications.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

We thank the Department of Civil Engineering, University of Moratuwa for facilitating initiative studies related to research work.

References

[1] H. Nguyen, T. Vu, T.P. Vo, H.-T. Thai, Efficient machine learning models for prediction of concrete strengths, Constr. Build. Mater. 266 (2021), 120950, https://doi.org/10.1016/j.conbuildmat.2020.120950.
[2] J.-S. Chou, W.K. Chong, D.-K. Bui, Nature-inspired metaheuristic regression system: programming and implementation for civil engineering applications, J. Comput. Civ. Eng. 30 (5) (2016), 04016007, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000561.
[3] D.A. Abrams, Water-cement ratio as a basis of concrete quality, J. Proc. 23 (2) (1927) 452–457, https://doi.org/10.14359/15426.
[4] S. Popovics, Analysis of concrete strength versus water-cement ratio relationship, Mater. J. 87 (5) (1990) 517–529, https://doi.org/10.14359/1944.
[5] O.Z. Cebeci, Strength of concrete in warm and dry environment, Mater. Struct. 20 (4) (1987) 270–272, https://doi.org/10.1007/BF02485923.
[6] M.A. DeRousseau, E. Laftchiev, J.R. Kasprzyk, B. Rajagopalan, W.V. Srubar, A comparison of machine learning methods for predicting the compressive strength of field-placed concrete, Constr. Build. Mater. 228 (2019), 116661, https://doi.org/10.1016/j.conbuildmat.2019.08.042.
[7] A. Fernández-Jiménez, A. Palomo, Characterisation of fly ashes. Potential reactivity as alkaline cements☆, Fuel 82 (18) (2003) 2259–2265, https://doi.org/10.1016/S0016-2361(03)00194-7.
[8] A.A. Ramezanianpour, V.M. Malhotra, Effect of curing on the compressive strength, resistance to chloride-ion penetration and porosity of concretes incorporating slag, fly ash or silica fume, Cem. Concr. Compos. 17 (2) (. 1995) 125–133, https://doi.org/10.1016/0958-9465(95)00005-W.
[9] A.M. Zeyad, Effect of curing methods in hot weather on the properties of high-strength concretes, J. King Saud. Univ. - Eng. Sci. 31 (3) (2019) 218–223, https://doi.org/10.1016/j.jksues.2017.04.004.

[10] A.T.A. Dantas, M. Batista Leite, K. de Jesus Nagahama, Prediction of compressive strength of concrete containing construction and demolition waste using artificial neural networks, Constr. Build. Mater. 38 (2013) 717–722, https://doi.org/10.1016/j.conbuildmat.2012.09.026.
[11] D.-C. Feng, et al., Machine learning-based compressive strength prediction for concrete: an adaptive boosting approach, Constr. Build. Mater. 230 (2020), 117000, https://doi.org/10.1016/j.conbuildmat.2019.117000.
[12] O.R. Abuodeh, J.A. Abdalla, R.A. Hawileh, Assessment of compressive strength of Ultra-high Performance Concrete using deep machine learning techniques, Appl. Soft Comput. 95 (2020), 106552, https://doi.org/10.1016/j.asoc.2020.106552.
[13] P.G. Asteris, V.G. Mokos, Concrete compressive strength using artificial neural networks, Neural Comput. Appl. 32 (15) (2020) 11807–11826, https://doi.org/10.1007/s00521-019-04663-2.
[14] A. Behnood, V. Behnood, M. Modiri Gharehveran, K.E. Alyamac, Prediction of the compressive strength of normal and high-performance concretes using M5P model tree algorithm, Constr. Build. Mater. 142 (2017) 199–207, https://doi.org/10.1016/j.conbuildmat.2017.03.061.
[15] A. Behnood, E.M. Golafshani, Machine learning study of the mechanical properties of concretes containing waste foundry sand, Constr. Build. Mater. 243 (2020), 118152, https://doi.org/10.1016/j.conbuildmat.2020.118152.
[16] W. Ben Chaabene, M. Flah, M.L. Nehdi, Machine learning prediction of mechanical properties of concrete:critical review, Constr. Build. Mater. 260 (2020), 119889, https://doi.org/10.1016/j.conbuildmat.2020.119889.
[17] M. Castelli, L. Vanneschi, S. Silva, Prediction of high performance concrete strength using genetic programming with geometric semantic genetic operators, Expert Syst. Appl. v40 (17) (2013) 6856–6862, https://doi.org/10.1016/j.eswa.2013.06.037.
[18] A.M. Diab, H.E. Elyamany, A.E.M. Abd Elmoaty, A.H. Shalan, Prediction of concrete compressive strength due to long term sulfate attack using neural network, Alex. Eng. J. 53 (3) (2014) 627–642, https://doi.org/10.1016/j.aej.2014.04.002.
[19] E.M. Golafshani, A. Behnood, M. Arashpour, Predicting the compressive strength of normal and high-performance concretes using ANN and ANFIS hybridized with grey wolf optimizer, Constr. Build. Mater. 232 (2020), 117266, https://doi.org/10.1016/j.conbuildmat.2019.117266.
[20] K. Güçlüer, A. Özbeyaz, S. Göymen, O. Günaydın, A comparative investigation using machine learning methods for concrete compressive strength estimation, Mater. Today Commun. 27 (2021), 102278, https://doi.org/10.1016/j.mtcomm.2021.102278.
[21] Q. Han, C. Gui, J. Xu, G. Lacidogna, A generalized method to predict the compressive strength of high-performance concrete by improved random forest algorithm, Constr. Build. Mater. 226 (2019) 734–742, https://doi.org/10.1016/j.conbuildmat.2019.07.315.
[22] M.-C. Kang, D.-Y. Yoo, R. Gupta, Machine learning-based prediction for compressive and flexural strengths of steel fiber-reinforced concrete, Constr. Build. Mater. 266 (2021), 121117, https://doi.org/10.1016/j.conbuildmat.2020.121117.
[23] B.A. Salami, T. Olayiwola, T.A. Oyehan, I.A. Raji, Data-driven model for ternary-blend concrete compressive strength prediction using machine learning approach, Constr. Build. Mater. 301 (2021), 124152, https://doi.org/10.1016/j.conbuildmat.2021.124152.
[24] U.K. Sevim, H.H. Bilgic, O.F. Cansiz, M. Ozturk, C.D. Atis, Compressive strength prediction models for cementitious composites with fly ash using machine learning techniques, Constr. Build. Mater. 271 (2021), 121584, https://doi.org/10.1016/j.conbuildmat.2020.121584.
[25] H. Song, et al., Predicting the compressive strength of concrete with fly ash admixture using machine learning algorithms, Constr. Build. Mater. 308 (2021), 125021, https://doi.org/10.1016/j.conbuildmat.2021.125021.
[26] A. Ahmad, W. Ahmad, F. Aslam, P. Joyklad, Compressive strength prediction of fly ash-based geopolymer concrete via advanced machine learning techniques, Case Stud. Constr. Mater. 16 (2022), e00840, https://doi.org/10.1016/j.cscm.2021.e00840.
[27] J. Xu, X. Zhao, Y. Yu, T. Xie, G. Yang, J. Xue, Parametric sensitivity analysis and modelling of mechanical properties of normal- and high-strength recycled aggregate concrete using grey theory, multiple nonlinear regression and artificial neural networks, Constr. Build. Mater. 211 (2019) 479–491, https://doi.org/10.1016/j.conbuildmat.2019.03.234.
[28] J.-S. Chou, C.-K. Chiu, M. Farfoura, I. Al-Taharwa, Optimizing the prediction accuracy of concrete compressive strength based on a comparison of data-mining techniques, J. Comput. Civ. Eng. 25 (3) (2011) 242–253, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000088.
[29] J. Duan, H. Zuo, Y. Bai, J. Duan, M. Chang, B. Chen, Short-term wind speed forecasting using recurrent neural networks with error correction, Energy 217 (2021), 119397, https://doi.org/10.1016/j.energy.2020.119397.
[30] A.-D. Pham, N.-D. Hoang, Q.-T. Nguyen, Predicting compressive strength of high-performance concrete using metaheuristic-optimized least squares support vector regression, J. Comput. Civ. Eng. 30 (3) (2016), 06015002, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000506.
[31] R. Siddique, P. Aggarwal, Y. Aggarwal, Prediction of compressive strength of self-compacting concrete containing bottom ash using artificial neural networks, Adv. Eng. Softw. 42 (10) (2011) 780–786.
[32] M. Uysal, H. Tanyildizi, Estimation of compressive strength of self compacting concrete containing polypropylene fiber and mineral additives exposed to high temperature using artificial neural network, Constr. Build. Mater. 27 (1) (2012) 404–414.
[33] S.R. Salimbahrami, R. Shakeri, Experimental investigation and comparative machine-learning prediction of compressive strength of recycled aggregate concrete, Soft Comput. 25 (2) (2021) 919–932, https://doi.org/10.1007/s00500-021-05571-1.
[34] B.G. Aiyer, D. Kim, N. Karingattikkal, P. Samui, P.R. Rao, Prediction of compressive strength of self-compacting concrete using least square support vector machine and relevance vector machine, KSCE J. Civ. Eng. 18 (6) (2014) 1753–1758.
[35] A. Gholampour, I. Mansouri, O. Kisi, T. Ozbakkaloglu, Evaluation of mechanical properties of concretes containing coarse recycled concrete aggregates using multivariate adaptive regression splines (MARS), M5 model tree (M5Tree), and least squares support vector regression (LSSVR) models, Neural Comput. Appl. 32 (1) (2020) 295–308, https://doi.org/10.1007/s00521-018-3630-y.
[36] F. Deng, Y. He, S. Zhou, Y. Yu, H. Cheng, X. Wu, Compressive strength prediction of recycled concrete based on deep learning, Constr. Build. Mater. 175 (2018) 562–569, https://doi.org/10.1016/j.conbuildmat.2018.04.169.
[37] İ.B. Topçu, M. Sarıdemir, Prediction of mechanical properties of recycled aggregate concretes containing silica fume using artificial neural networks and fuzzy logic, Comput. Mater. Sci. 42 (1) (2008) 74–82, https://doi.org/10.1016/j.commatsci.2007.06.011.
[38] H. Naderpour, A.H. Rafiean, P. Fakharian, Compressive strength prediction of environmentally friendly concrete using artificial neural networks, J. Build. Eng. 16 (2018) 213–219, https://doi.org/10.1016/j.jobe.2018.01.007.
[39] D.-K. Bui, T. Nguyen, J.-S. Chou, H. Nguyen-Xuan, T.D. Ngo, A modified firefly algorithm-artificial neural network expert system for predicting compressive and tensile strength of high-performance concrete, Constr. Build. Mater. 180 (2018) 320–333, https://doi.org/10.1016/j.conbuildmat.2018.05.201.
[40] D.V. Dao, H.-B. Ly, S.H. Trinh, T.-T. Le, B.T. Pham, Artificial intelligence approaches for prediction of compressive strength of geopolymer concrete, Art. no. 6, Materials 12 (6) (2019), https://doi.org/10.3390/ma12060983.
[41] A. Behnood, V. Behnood, M. Modiri Gharehveran, K.E. Alyamac, Prediction of the compressive strength of normal and high-performance concretes using M5P model tree algorithm, Constr. Build. Mater. 142 (2017) 199–207, https://doi.org/10.1016/j.conbuildmat.2017.03.061.
[42] A. Behnood, E.M. Golafshani, Machine learning study of the mechanical properties of concretes containing waste foundry sand, Constr. Build. Mater. 243 (2020), 118152, https://doi.org/10.1016/j.conbuildmat.2020.118152.
[43] B.A. Omran, Q. Chen, R. Jin, Comparison of data mining techniques for predicting compressive strength of environmentally friendly concrete, J. Comput. Civ. Eng. 30 (6) (2016), 04016029, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000596.
[44] V. Quan Tran, V. Quoc Dang, L. Si Ho, Evaluating compressive strength of concrete made with recycled concrete aggregates using machine learning approach, Constr. Build. Mater. 323 (2022), 126578, https://doi.org/10.1016/j.conbuildmat.2022.126578.
[45] Y. Ayaz, A.F. Kocamaz, M.B. Karakoç, Modeling of compressive strength and UPV of high-volume mineral-admixtured concrete using rule-based M5 rule and tree model M5P classifiers, Constr. Build. Mater. 94 (2015) 235–240, https://doi.org/10.1016/j.conbuildmat.2015.06.029.
[46] Q. Han, C. Gui, J. Xu, G. Lacidogna, A generalized method to predict the compressive strength of high-performance concrete by improved random forest algorithm, Constr. Build. Mater. 226 (2019) 734–742, https://doi.org/10.1016/j.conbuildmat.2019.07.315.
[47] W. Ben Chaabene, M. Flah, M.L. Nehdi, Machine learning prediction of mechanical properties of concrete: critical review, Constr. Build. Mater. 260 (2020), 119889, https://doi.org/10.1016/j.conbuildmat.2020.119889.

[48] M. Castelli, L. Vanneschi, S. Silva, Prediction of high performance concrete strength using genetic programming with geometric semantic genetic operators, Expert Syst. Appl. 40 (17) (2013) 6856–6862, https://doi.org/10.1016/j.eswa.2013.06.037.

[49] A.H. Gandomi, A.H. Alavi, M.G. Sahab, New formulation for compressive strength of CFRP confined concrete cylinders using linear genetic programming, Mater. Struct. 43 (7) (2010) 963–983, https://doi.org/10.1617/s11527-009-9559-y.

[50] V. Belle, I. Papantonis, Principles and practice of explainable machine learning, Front. Big Data 4 (2021), 688969, https://doi.org/10.3389/fdata.2021.688969.

[51] R. Roscher, B. Bohn, M.F. Duarte, J. Garcke, Explainable machine learning for scientific insights and discoveries, IEEE Access 8 (2020) 42200–42216, https://doi.org/10.1109/ACCESS.2020.2976199.

[52] F. Xu, H. Uszkoreit, Y. Du, W. Fan, D. Zhao, J. Zhu, Explainable AI: a brief survey on history, research areas, approaches and challenges, Nat. Lang. Process. Chin. Comput. (2019) 563–574, https://doi.org/10.1007/978-3-030-32236-6_51.

[53] D. Chakraborty, I. Awolusi, L. Gutierrez, An explainable machine learning model to predict and elucidate the compressive behavior of high-performance concrete, Results Eng. 11 (2021), 100245, https://doi.org/10.1016/j.rineng.2021.100245.

[54] A. Tahmassebi, M. Motamedi, A.H. Alavi, A.H. Gandomi, An explainable prediction framework for engineering problems: case studies in reinforced concrete members modeling, Eng. Comput. 39 (2) (2021) 609–626, https://doi.org/10.1108/EC-02-2021-0096.

[55] K. Broelemann and G. Kasneci, A gradient-based split criterion for highly accurate and transperent model trees, in *Proceedings of the twenty eighth international joint conference on artificial intellignece*, Germany, pp. 2030–2037.

[56] H.H. Patel, P. Prajapati, Study and analysis of decision tree based classification algorithms (doi: https://doi.org/), Int. J. Comput. Sci. Eng. 6 (10) (2018) 74–78, https://doi.org/10.26438/ijcse/v6i10.7478.

[57] I.-C. Yeh, Modeling of strength of high-performance concrete using artificial neural networks, Cem. Concr. Res. 28 (12) (1998) 1797–1808, https://doi.org/10.1016/S0008-8846(98)00165-3.

[58] M.A. Ahmad, C. Eckert, and A. Teredesai, Interpretable Machine Learning in Healthcare, in *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informa*tics, New York, NY, USA, Aug. 2018, pp. 559–560. doi: 10.1145/3233547.3233667.

[59] O. Sagi, L. Rokach, Explainable decision forest: transforming a decision forest into an interpretable tree, Inf. Fusion 61 (2020) 124–138, https://doi.org/10.1016/j.inffus.2020.03.013.

[61] Y. Liang, S. Li, C. Yan, M. Li, C. Jiang, Explaining the black-box model: a survey of local interpretation methods for deep neural networks, Neurocomputing 419 (2021) 168–182, https://doi.org/10.1016/j.neucom.2020.08.011.

[62] A. Ross and F. Doshi velez, Improving the Adversarial Robustness and Interpretability of Deep Neural Networks by Regularizing their Input Gradients, Nov. 2017.

[63] M.D. Zeiler, R. Fergus, Vis. Underst. Convolutional Netw., *Comput. Vis. – ECCV 2014* Sep. 2014 818 833 doi: 10.1007/978-3-319-10590-1_53.

[64] B.N. Patro, M. Lunayach, S. Patel, V.P. Namboodiri, U-CAM: Vis. Explan. Using Uncertain. Based Cl. Act. Maps 2019 7444 7453.Accessed: Jun. 17, 2021. [Online]. Available⟨https://openaccess.thecvf.com/content_ICCV_2019/html/Patro_U-CAM_Visual_Explanation_Using_Uncertainty_Based_Class_Activation_Maps_ICCV_2019_paper.html⟩.

[65] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization, in 2017 *IEEE* International Conference on C*omputer Vision (ICCV)*, Oct. 2017, pp. 618–626. doi: 10.1109/ICCV.2017.74.

[66] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, Learning Deep Features for Discriminative Localization, in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2016, pp. 2921–2929. doi: 10.1109/CVPR.2016.319.

[67] A. Binder, G. Montavon, S. Lapuschkin, K.-R. Müller, W. Samek, Layer-wise relevance propagation for neural networks with local renormalization layers, Artif. Neural Netw. Mach. Learn. (2016) 63–71, https://doi.org/10.1007/978-3-319-44781-0_8.

[68] M. Sundararajan, A. Taly, and Q. Yan, Axiomatic attribution for deep networks, in *Proceedings of the 34th International Conference on Machine Learning - Volume 70* , Sydney, NSW, Australia, Aug. 2017, pp. 3319–3328.

[69] J. Zhang, S.A. Bargal, Z. Lin, J. Brandt, X. Shen, S. Sclaroff, Top-down neural attention by excitation backprop, Int. J. Comput. Vis. 126 (10) (2018) 1084–1102, https://doi.org/10.1007/s11263-017-1059-x.

[70] Q. Zhang, Y.N. Wu, S.-C. Zhu, Interpret. Convolutional Neural Netw. 2018 8827 8836.Accessed: Jun. 17, 2021. [Online]. Available⟨https://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_Interpretable_Convolutional_Neural_CVPR_2018_paper.html⟩.

[71] C. Etmann, S. Lunz, P. Maass, and C. Schoenlieb, On the Connection Between Adversarial Robustness and Saliency Map Interpretability, in I*nternational Conference on Machine Learning*, May 2019, pp. 1823–1832. Accessed: Jun. 17, 2021. [Online]. Available: ⟨http://proceedings.mlr.press/v97/etmann19a.html⟩.

[72] G. Tao, S. Ma, Y. Liu, and X. Zhang, Attacks Meet Interpretability: Attribute-steered Detection of Adversarial Samples, *ArXiv181011580 Cs Stat*, Oct. 2018, Accessed: Jun. 17, 2021. [Online]. Available: http://arxiv.org/abs/1810.11580.

[73] A. Ghorbani, J. Wexler, J. Zou, and B. Kim, Towards Automatic Concept-based Explanations, *ArXiv190203129 Cs Stat*, Oct. 2019, Accessed: Jun. 17, 2021. [Online]. Available: http://arxiv.org/abs/1902.03129.

[74] B. Zhou, Y. Sun, D. Bau, A. Torralba, Interpretable basis decomposition for visual explanation, Comput. Vis. (2018) 122–138, https://doi.org/10.1007/978-3-030-01237-3_8.

[75] Y. Aydin, B. Dizdaroğlu, Blotch detection in archive films based on visual saliency map, Complexity (2020), https://doi.org/10.1155/2020/5965387.

[76] R. Fong A. Vedaldi, Interpretable Explanations of Black Boxes by Meaningful Perturbation, *20*17 IEEE Int. Conf. Comput. Vis. *ICCV*, pp. 3449–3457, Oct. 2017, doi: 10.1109/ICCV.2017.371.

[77] M.T. Ribeiro, S. Singh, C. Guestrin, 'Why Should I Trust You?': Explaining the Predictions of Any Classifier HLT-NAACL Demos 2016 doi: 10.1145/2939672.2939778.

[78] V. Petsiuk, A. Das, and K. Saenko, RISE: Randomized Input Sampling for Explanation of Black-box Models, ArXiv180607421 Cs, Jun. 2018, Accessed: Apr. 11, 2021. [Online]. Available:⟨http://arxiv.org/abs/1806.07421⟩,

[79] S.M. Lundberg and S.-I. Lee, A unified approach to interpreting model predictions, in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, Dec. 2017, pp. 4768–4777.

[80] M. Moradi, M. Samwald, Post-hoc explanation of black-box classifiers using confident itemsets (doi: DOI), Expert Syst. Appl. (2021), https://doi.org/10.1016/j.eswa.2020.113941.

[81] M. Xu, P. Watanachaturaporn, P.K. Varshney, M.K. Arora, Decision tree regression for soft classification of remote sensing data, Remote Sens. Environ. 97 (3) (2005) 322–336, https://doi.org/10.1016/j.rse.2005.05.008.

[82] L. Breiman, J. Friedman, R. Olshen, C.J. Stone, Classif. Regres. Trees 1983 doi: 10.2307/2530946.

[83] M.W. Ahmad, J. Reynolds Y. Rezgui Predict. Model. Sol. Therm. Energy Syst.: A Comp. Support Vector Regres., Random For., Extra trees Regres. trees vol., 203, 1, 2018, pp. 810–821 doi: 10.1016/j.jclepro.2018.08.207.

[84] V. Rodriguez-Galiano, M. Sanchez-Castillo, M. Chica-Olmo, M. Chica-Rivas, Machine learning predictive models for mineral prospectivity: an evaluation of neural networks, random forest, regression trees and support vector machines, Ore Geol. Rev. 71 (2015) 804–818, https://doi.org/10.1016/j.oregeorev.2015.01.001.

[85] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees (doi: DOI), Mach. Learn. 63 (2005), https://doi.org/10.1007/s10994-006-6226-1.

[86] V. John, Z. Liu, C. Guo, S. Mita, K. Kidono, Real-time lane estimation using deep features and extra trees regression, Image Video Technol. (2015) 721–733, https://doi.org/10.1007/978-3-319-29451-3_57.

[87] A. Mannodi-Kanakkithodi, G. Pilania, R. Ramprasad, Critical assessment of regression-based machine learning methods for polymer dielectrics, Comput. Mater. Sci. 125 (2016) 123–135, https://doi.org/10.1016/j.commatsci.2016.08.039.

[88] Y. Freund and R.E. Schapire, Experiments with a New Boosting Algorithm, in *In Proceedings of the Thirteenth International Conference on Machine Learning*, 1996, pp. 148–156.

[89] D.P. Solomaitine and D.L. Shresthra, AdaBoost RT: A boosting algorithm for regression problems, in *IEEE international joint conference oon neural networks*, 2004, vol. 2, pp. 1163–1168. doi: 10.1109/IJCNN.2004.1380102.
[90] C. Xu, et al., A study of predicting irradiation-induced transition temperature shift for RPV steels with XGBoost modeling, Nucl. Eng. Technol. (2021), https://doi.org/10.1016/j.net.2021.02.015.
[91] G. Ke et al., LightGBM: A Highly Efficient Gradient Boosting Decision Tree, Nov. 2017, Accessed: Apr. 11, 2021. [Online]. Available: ⟨https://www.microsoft.com/en-us/research/publication/lightgbm-a-highly-efficient-gradient-boosting-decision-tree/⟩.
[92] J. Fan, X. Ma, L. Wu, F. Zhang, X. Yu, W. Zeng, Light gradient boosting machine: an efficient soft computing model for estimating daily reference evapotranspiration with local and external meteorological data, Agric. Water Manag. 225 (2019), 105758, https://doi.org/10.1016/j.agwat.2019.105758.
[93] I.H. Ebtehaj, H. Bonakdari, A.H. Zaji, H. Azimi, F. Khoshbin, GMDH-type neural network approach for modeling the discharge coefficient of rectangular sharp-crested side weirs, Int. J. Eng. Sci. Technol. 18 (4) (2015) 746–757, https://doi.org/10.1016/j.jestch.2015.04.012.
[94] I.-C. Yeh, Modeling concrete strength with augment-neuron networks, J. Mater. Civ. Eng. 10 (4) (1998) 263–268, https://doi.org/10.1061/(ASCE)0899-1561(1998)10:4(263).
[95] I.-C. Yeh, Design of high-performance concrete mixture using neural networks and nonlinear programming, J. Comput. Civ. Eng. 13 (1) (1999) 36–42, https://doi.org/10.1061/(ASCE)0887-3801(1999)13:1(36).
[96] I.-C. Yeh, Analysis of strength of concrete using design of experiments and neural networks, J. Mater. Civ. Eng. 18 (4) (2006) 597–604, https://doi.org/10.1061/(ASCE)0899-1561(2006)18:4(597).
[97] N. Macià, E. Bernadó-Mansilla, Towards UCI+: a mindful repository design, Inf. Sci. 261 (2014) 237–262, https://doi.org/10.1016/j.ins.2013.08.059.
[98] I. Syarif, A. Prugel-Bennett, G. Wills, SVM parameter optimization using grid search and genetic algorithm to improve classification performance, Art. no. 4, TELKOMNIKA Telecommun. Comput. Electron. Control 14 (4) (2016), https://doi.org/10.12928/telkomnika.v14i4.3956.
[99] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.
[100] D.-K. Bui, T. Nguyen, J.-S. Chou, H. Nguyen-Xuan, T.D. Ngo, A modified firefly algorithm-artificial neural network expert system for predicting compressive and tensile strength of high-performance concrete, Constr. Build. Mater. 180 (2018) 320–333, https://doi.org/10.1016/j.conbuildmat.2018.05.201.
[101] D.V. Dao, H. Ly, S.H. Trinh, T. Le, B.T. Phan, Artificial intelligence approaches for prediction of compressive strength of geopolymer concrete, Materials 12 (6) (2019), https://doi.org/10.3390/ma12060983.
[102] Y. Yu, W. Li, J. Li, T.N. Nguyen, A novel optimised self-learning method for compressive strength prediction of high performance concrete, Constr. Build. Mater. 184 (2018) 229–247, https://doi.org/10.1016/j.conbuildmat.2018.06.219.
[103] A.K. Al-Shamiri, J.H. Kim, T.-F. Yuan, Y.S. Yoon, Modeling the compressive strength of high-strength concrete: an extreme learning approach, Constr. Build. Mater. 208 (2019) 204–219, https://doi.org/10.1016/j.conbuildmat.2019.02.165.
[104] Z. Yuan, L.-N. Wang, X. Ji, Prediction of concrete compressive strength: research on hybrid models genetic based algorithms and ANFIS, Adv. Eng. Softw. 67 (2014) 156–163, https://doi.org/10.1016/j.advengsoft.2013.09.004.