



A review of soft computing techniques in predicting the compressive strength of concrete and the future scope

Tanvesh Dabholkar¹ · Harish Narayana² · Prashanth Janardhan³

Received: 12 January 2023 / Accepted: 13 May 2023 / Published online: 24 May 2023
© Springer Nature Switzerland AG 2023

Abstract

Structural design of Reinforced Cement Concrete (RCC) highly depends on the compressive strength of the concrete used. The compressive strength determination techniques are categorized as destructive, non-destructive, and partially destructive. In non-destructive techniques, the equipment is costly and needs expertise. The compressive strength of concrete is influenced by multiple parameters and materials used in making the concrete. Soft computing techniques like Machine learning (ML) and artificial intelligence (AI) have been proven to find hidden relations between multiple parameters and achieve the desired result. The inclusion of AI/ML has enabled the characterization of the strength with advanced techniques based on the individual constituents or images using digital image correlation. Based on the literature reviewed in this study, ML and AI techniques have shown promising outcomes in predicting the compressive strength of concrete. This study systematically examines the contributions made to date in predicting compressive strength utilizing AI-ML-based strategies. It compares and highlights existing literature based on the type of machine learning techniques used, datasets used, evaluation parameters, and performance of different methods. The study does not encompass high strain rate loading or dynamic type of loading. This paper also aims to find the gap in the research conducted and state the potential scope of estimating compressive strength using soft computing techniques.

Keywords Artificial intelligence · Concrete compressive strength · Machine learning · Soft computing

Abbreviations

AI	Artificial intelligence	GA	Genetic algorithm
ANFIS	Adaptive neuro-fuzzy inference system	GEP	Gene expression programming
ANN	Artificial neural network	GP	Genetic programming
BP	Back propagation	GPR	Gaussian process regression
CA	Coarse aggregate	KNN	K nearest neighbour
CNN	Convolutional neural network	LM	Levenberg–Marquardt
DNN	Deep neural network	LR	Linear regression
DT	Decision tree	LSSVM	Least square support vector machine
ELM	Extreme learning machine	MAE	Mean absolute error
ENN	Emotional neural network	MAPE	Mean absolute percentage error
FA	Fine aggregate	MARS	Multivariate adaptive regression spline
FL	Fuzzy logic	ML	Machine learning
		MLP	Multi-layer perceptron
		MLR	Multi-linear regression
		MNLR	Multi non-linear regression
		MPMR	Minimax probability regression
		MSE	Mean squared error
		NDT	Non-destructive testing
		NLR	Non-linear regression
		NN	Neural networks
		R	Regression coefficient
		RBM	Restricted Boltzmann machine

✉ Tanvesh Dabholkar
tanveshd@gmail.com

¹ Department of Civil Engineering, National Institute of Technology, Goa, India

² Visual and Transparent Pvt Ltd, Bengaluru, Karnataka, India

³ Department of Civil Engineering, National Institute of Technology, Silchar, Assam, India

ReLu	Rectified linear unit
RF	Random forest
RMSE	Root mean square error
RSR	RMSE-standard deviation ratio
RVM	Relevance vector machine
SA	Simulated annealing
SVM	Support vector machine
SVR	Support vector regression
VAF	Variance account for
WMAPE	Weighted mean absolute percentage error
WRA	Water-reducing admixture
XGB	Extreme gradient boost

Introduction

As documented by various standard codes, a typical RCC structure design must be checked for satisfying strength and durability or serviceability criteria. Strength criteria must satisfy flexure, compression, shear, and torsion requirements, whereas serviceability criteria must satisfy deflection and cracking [1–4]. Each of these requirements is the function of the compressive strength of the concrete directly or indirectly. This makes it of utmost importance to ensure that the concrete used for construction achieves the desired compressive strength used while designing the structure. For this purpose, testing and estimating the compressive strength of in situ concrete is imperative.

The sampling frequency is defined in several codes, and a general value can be considered at least 1 sample for each 1–5 cubic meters of concreting work. Every batch of concrete that has been prepared is taken for testing [1]. Each concrete sample must be of a minimum of 0.02 cubic meters in volume and be taken from different points from the concreting work [5]; thus, the compressive test must be carried out multiple times throughout the project. The compressive test must be carried out as outlined in several codes [4, 6, 7]. In addition, the compressive strength of concrete can also be determined by non-destructive tests [8]. Although non-destructive tests do not require sampling and are quick to perform, these methods have higher dispersion and deviation from the true compressive strength [9]. Thus, the conventional techniques of compressive testing require expensive equipment, skilled labour, and a dedicated testing facility. These requirements are challenging to follow in a low-cost construction due to the non-availability of funds, critical and short project spans, and the non-availability of a dedicated testing area.

The compressive strength of concrete is not a stochastic property, but it is dependent on various factors like water-cement ratio [10], cement strength class [11], aggregates and admixtures [12], shape and size [9], to name a few factors, apart from these the conditions under which the

concrete is cast like the ambient environment temperature [14], site quality control and various other factors. This makes it a function of some pre-defined parameters between which a relationship can be established. Based on the past literature, it has been seen that soft computing techniques are well suited to finding relationships between parameters to get the desired output in multiple fields [15–19]. It is also seen that conventional modelling techniques of concrete fall short for complex concrete mixes. Soft computing techniques have been proven to predict the compressive strength of a simple concrete mix [20] and a complex of steel fibre-reinforced concrete [21]. The conventional modelling techniques were difficult to model the compressive strength of the complex mix model due to a large number of forecasting variables; soft computing techniques proved useful.

This paper highlights the research in this domain using AI and ML, predominantly bifurcating it into four methods. The first is the conglomerate of algorithms based on rules implemented, the second is the classical ML methods, including the regression and classification techniques, the third is the shallow neural networks, and the fourth is deep machine learning. The approaches reviewed in this study range from simple parametric methods to various AI and ML techniques like shallow neural network models, deep neural networks, and also computer vision-based methods in which the image of the surface of concrete [22] or microtomographic images of the concrete [23]. The computer vision-based methods employ convolutional neural network models to identify patterns in the images and pass them through a deep neural network or classical ML techniques to deduce a satisfactory conclusion.

Qualities a successful prediction should have are scalability of the model, i.e., the results should apply to a large-scale concreting as well; secondly, it should be repeatable; and thirdly, it should conform to evaluation metrics. The performance of soft computing techniques is checked by various evaluation metrics like root mean square error (RMSE), mean absolute error, R^2 value, and mean absolute percentage error (MAPE). The same shall be discussed in brief in the upcoming section.

This paper aims to critically review different soft computing techniques used to predict the compressive strength of concrete. Figure 1 shows the percentage of soft computing techniques utilized by various literature reviewed in this study. The rest of the paper is organized as follows. In section II, the working principles of some of the popular AI algorithms which are encountered in various literature are described. Section III describes the popular evaluation metric used. Section IV describes methodologies used by researchers and their performing capabilities to predict the compressive strength of concrete. Section V describes the conclusion and the future research scope.

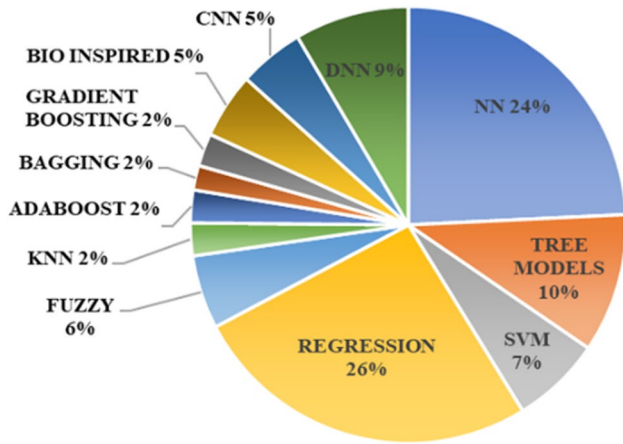


Fig. 1 Pie chart of different algorithms in this study, all regression models are grouped as ‘REGRESSION’, all tree models (DT, RF, Boosting Trees and other similar algorithms) are grouped as ‘TREE MODELS’, neural network models except DNN and CNN are grouped as ‘NN’, Bio-inspired algorithms, GEP, GP are grouped as ‘BIO INSPIRED’

Artificial intelligence algorithms

Algorithms based on rules

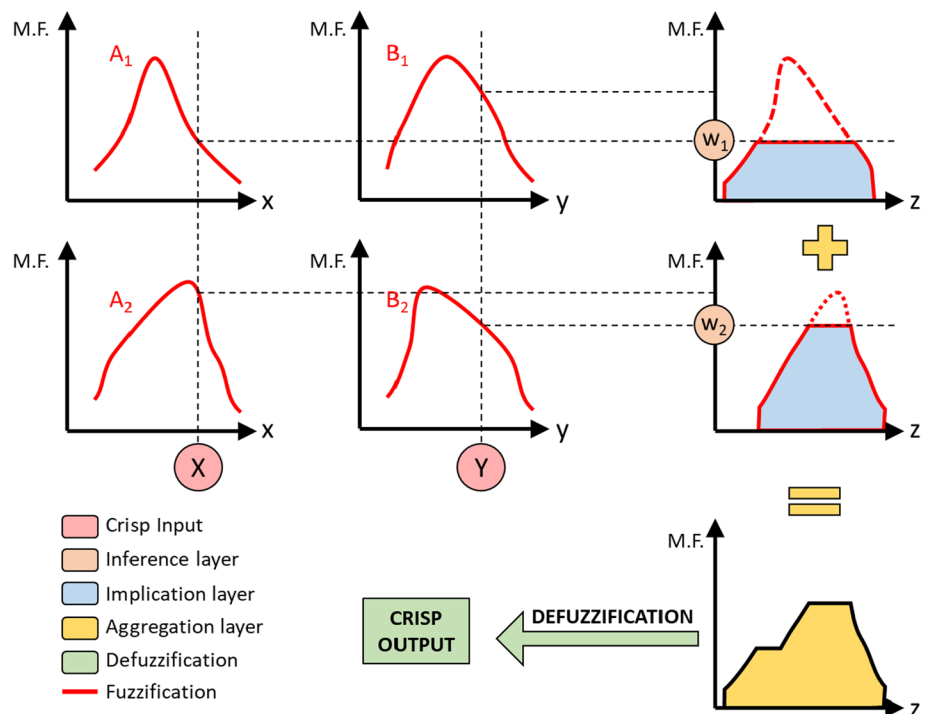
Adaptive neuro-fuzzy inference system

Fuzzy networks can be optimized using backpropagation and other training functions implemented in a neural network

system. This system is called a neuro-fuzzy system. One of the methods for neuro-fuzzy inference is the Mamdani inference system, as shown in Fig. 2. In the Mamdani inference system, a crisp input is applied to a fuzzy set, resulting in a fuzzy output set, which results in a crisp output by de-fuzzification. Mamdani fuzzy inference system has five layers, namely:

- i. Fuzzification layer: This layer is used for calculating the membership functions. There are different types of membership functions like trapezoidal function, bell function, triangular function, Gaussian function, and left–right function, to list a few. For most cases, the Gaussian membership function is used.
- ii. Inference layer: The inference layer is the reasoning mechanism of the neuro-fuzzy system. A fuzzy rule can be defined as a relation between its antecedent (input) and the consequent (output). Based on this, fuzzy reasoning can be defined as a single rule with a single antecedent, a single rule with multiple antecedents, or multiple rules with numerous antecedents. Weights in the inference system are found using the T-norm operator.
- iii. Implication layer: The consequent membership functions are calculated based on the inference layer.
- iv. Aggregation layer: The implied results are summed up to give a fuzzy output.
- v. Defuzzification layer: Defuzzification is the process of converting fuzzy output to a crisp output. In Mamdani

Fig. 2 Mamdani Fuzzy Inference System



Inference System, the centroid of the fuzzy output is calculated, and that value is the crisp output.

Fuzzy logic

A Boolean logic decision-making system takes only two values of true and false or 1 and 0 [24]. These values can be put through several logic gates to get the desired results. However, this logic system does not encompass any condition between true and false, for example, partially true, partially false, or true to some extent, and various other linguistic syntaxes like negations, hedges, and connectives. These scenarios are well captured by fuzzy logic. The decision making by fuzzy logic includes a fuzzification module in which membership values are assigned to specific criteria or conditions. Membership values are like weights that describe to what extent a particular condition is valid. A relationship is formed between membership values using IF-THEN rules, linguistic rules, or conditional rules, which give fuzzy logic an outlook from the perspective of humans. An inference system provides a de-fuzzified output from the relation established to get a crisp value.

Classical ML and clustering

Support vector machine (SVM)

Support vector machine is a classification algorithm used for applications like the classification of images, pattern recognition, and text categorization. The data is classified by constructing a hyperplane of (n-1) dimensions, where 'n' is an n-dimensional vector used to describe a data point.

The best hyperplane is the one that ensures maximum margin distance from the data points involved in classification. Figure 3a shows a hyperplane and margin distance in SVM for classification purposes. The data points closest to the hyperplane are called support vectors in this process. The kernel or classifier selection is essential for precise classification using SVM. Different types of kernels in SVM are the polynomial kernel, sigmoid kernel, and radial basis kernel. These three are the non-linear kernels, and the fourth is the

linear kernel classifier. SVM has the capability of working with linear as well as non-linear data.

Support vector regression (SVR)

In simple regression problems, the final aim is to minimize the square of the error to ensure the best fit; however, in support vector regression, the main concern is with error reduction only to a certain degree. It is accepted if the predicted value, including the error, falls within the acceptable range. In SVR, the hyperplane is constructed as the best fit line, and the hyperplane is offset by an error tolerance 'ε' on both sides. The aim of SVR is that all data points should fall within this tolerance region. Introducing a slack variable minimizes their deviation for values outside the tolerance region. A slack variable 'ξ' denotes the deviation of a data point from the margins of the tolerance region. Figure 3b shows a schematic diagram for SVR. SVR and SVM work on similar principles of creating hyperplanes using the kernel; SVM is used for classification, whereas SVR gives a numerical answer.

Multi-linear regression

Multi-linear regression is a multivariate algorithm for supervised machine learning. It aims at establishing the relationship between multiple input variables and a dependent output variable. This algorithm is validated by the least square error between the actual and predicted values. A general form of multi-linear regression is given in equation (1) [25].

$$y = \beta_0 + \sum_{i=1}^n \beta_i x_i + \varepsilon \quad (1)$$

where 'y' is the dependent variable on 'n' different independent variables, β_0 is the constant, β_i is the regression vector, and ε is the error.

Non-linear regression

Non-linear regression is a multivariate higher degree algorithm for supervised machine learning. This algorithm

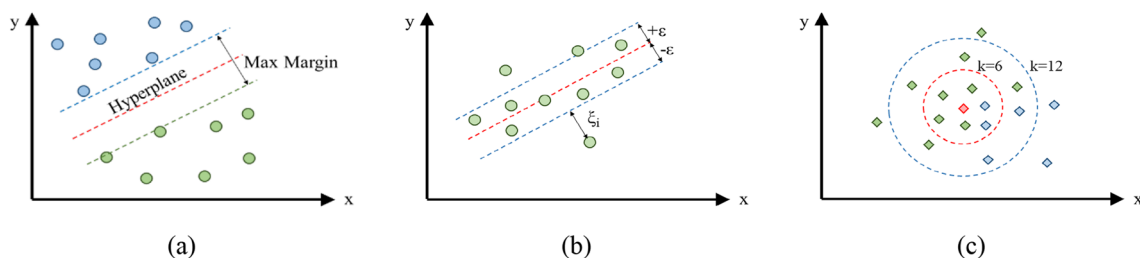


Fig. 3 **a** Support Vector Machine (SVM) algorithm. **b** Support Vector Regression (SVR) algorithm. **c** K-nearest neighbour (KNN) algorithm

establishes a relationship between the dependent variable and a polynomial multi-variable equation. Non-linear regression is used when the relationship between a dependent and independent variable is complex and not linear. A general form of non-linear regression is given in equation (2) [25].

$$y = \beta_0 + \sum_{\substack{i=1 \\ m \in \mathbb{R}}}^n \beta_i x_n^m + \varepsilon \tag{2}$$

where ‘y’ is the dependent variable on ‘n’ independent variables with a degree ‘m’, m belongs to real number \mathbb{R} , β_0 is the constant, and ε is the error.

Gaussian process regression

Gaussian process regression is based on the fact that infinite functions could fit the control data points for a particular distribution of data points. A non-parametric regression model calculates the probability distribution for all functions that fit the data points. The Gaussian process uses a kernel-based probabilistic model. The Gaussian process for an input ‘x’ and corresponding output ‘y’ is given by Eq. (3) [25].

$$y = g(x)^T \alpha + f(x) + \varepsilon f(x) \sim N(\mu, K) \tag{3}$$

where g(x) is a vector function that relates the input variables to a multi-dimensional space, α is coefficient to g(x), f(x) constitutes for Gaussian process with μ being the mean vector matrix and K being the covariance matrix. The covariance matrix mathematically can be expressed as in equation (4) [25].

$$K = \begin{bmatrix} K(x_1, x_1) & \dots & K(x_1, x_n) \\ \vdots & \ddots & \vdots \\ K(x_n, x_1) & \dots & K(x_n, x_n) \end{bmatrix} = \begin{bmatrix} 1 & \dots & K(x_1, x_n) \\ \vdots & \ddots & \vdots \\ K(x_n, x_1) & \dots & 1 \end{bmatrix} \tag{4}$$

Extreme gradient boosting (XGBoost)

Extreme Gradient boosting is a decision tree based supervised learning technique. It can be used for classification and regression purposes. It employs a gradient boosting framework along with distributed processing or parallel processing. Gradient boosting alone suffers from problems like overfitting and more computation time; hence the extreme Gradient boosting method was evolved as a modified gradient boosting method. The training time of XGBoost is lesser than the Gradient boosting method. This method can model the non-linear interactions between various features used in machine learning.

K-nearest neighbour (KNN)

K nearest neighbour assumes that a data point has the same characteristics as that of maximum data points in its vicinity. It is a supervised machine learning technique used for classification and regression. A search space of potential solutions is searched locally within a radius such that ‘k’ nearest neighbours are selected. Figure 3c shows a KNN algorithm for k=6 and k=12, highlighting the significance of the ‘k’ value. The value of ‘k’ is thus an important parameter that needs to be determined first; it should not be too large or too small. Once the value of ‘k’ is decided, the data point is classified based on its proximity to its ‘k’ neighbours. This degree of proximity is quantified by finding the Euclidean distance and then sorting the distance from the most minor to the most significant order. Euclidean distance is given in equation (5) [25].

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{5}$$

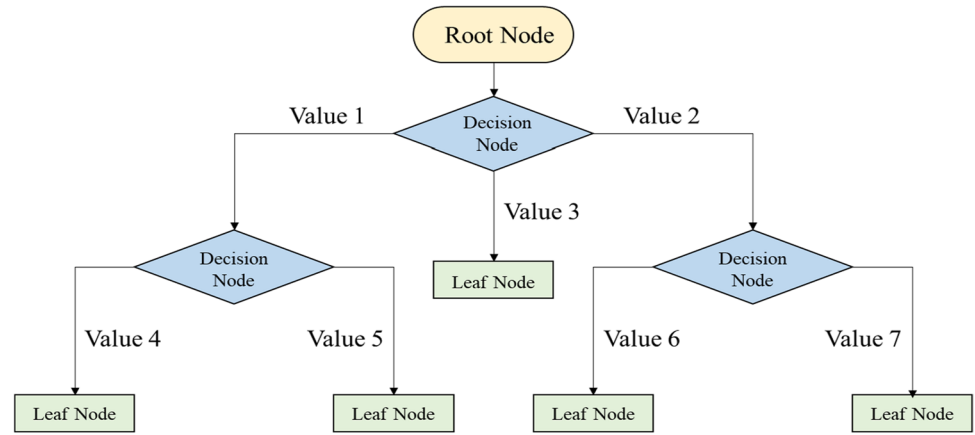
The class label of the first ‘k’ neighbours is considered, and the label that occurs a maximum number of times is assigned as the class of the data point in consideration.

Decision tree algorithm

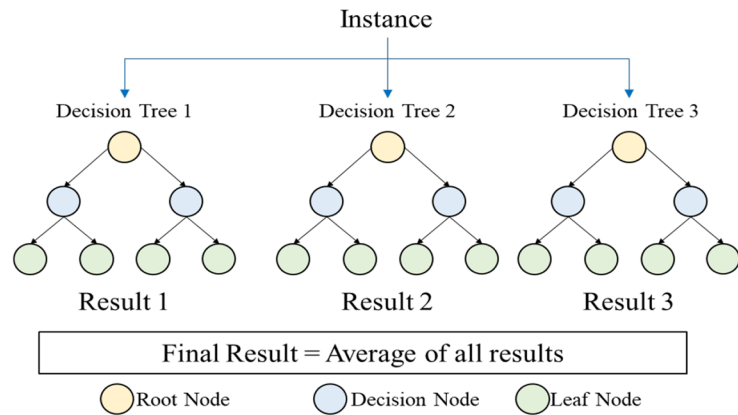
The decision tree is a supervised learning algorithm that shows existing patterns between predictors and a dependent variable using a combination of predictable rules. The classification technique generates decisions based on the information fed [26]. The schematic representation of rules and their outcomes looks like a tree diagram. The decision tree algorithm is well suited for classification as well as regression tasks. It typically has three types of nodes: root, decision, and leaf node. The root node is the base node and typically the starting point of the algorithms.

Figure 4a shows various decision tree components and bifurcation. The decision node is the node that determines the course path for subsequent outcomes. A leaf node is a terminal node having the outcome of a particular rule. Decision trees are sensitive to the data on which they are trained; hence if the training data is changed, the structure of the decision tree alters. An advantage of the decision tree algorithm is that it can also be trained for noisy data. This algorithm is easier than others but prone to overfitting. More complex tree-based algorithms like random forest and boosting trees are often preferred over decision tree algorithms.

Fig. 4 **a** Decision Tree algorithm. **b** Random Forest algorithm



(a)



(b)

Random forest

A random forest is an ensemble of different decision tree algorithms to form a forest and reduce the disadvantage of individual trees. Random forests are an improvement over bagged decision tree.

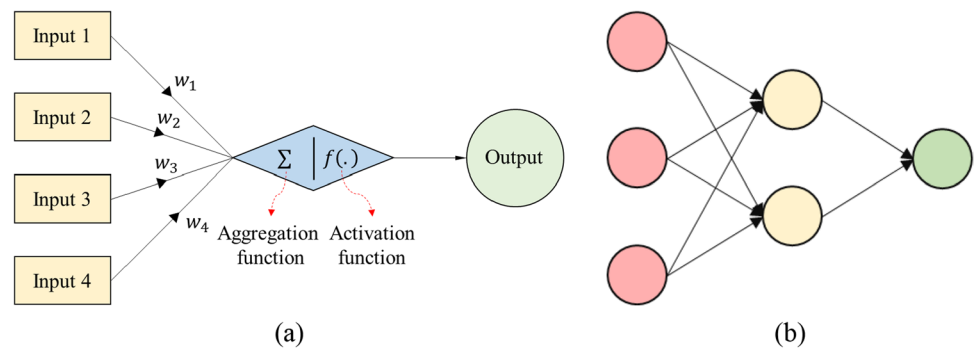
The bagged decision tree algorithm creates multiple subtrees for multiple subsets from the original data. A classification or regression task for a data point is done by taking the average value of all the individual decision trees. Figure 4b shows the classification process of the random forest algorithm.

Bagged decision tree models work better when the correlation between subtrees is less. This is not ensured in the bagged decision tree model; however, random forest ensures this by restricting the learning algorithm to a random feature instead of a learning algorithm searching through all data points in the bagged tree model.

Genetic programming

Genetic programming is a guided random search algorithm based on evolutionary techniques used to search solution space for potential solutions. These algorithms are inspired by nature and are generally used for optimization problems. Genetic programming can solve unconventional problems requiring great computational effort when followed by the traditional approach. A common framework of genetic programming starts by defining the genes which contain chromosomes. Chromosomes are a piece of information contained in a gene. Genetic programming aims to search for a better solution than the previous one. For this purpose, the genes are altered by crossover, mutation, and best gene selection. Genetic programming can also be coupled with other neural network systems to optimize specific parameters and yield better output. Leung et al. [27] tuned the structure of neural networks using an improved genetic algorithm.

Fig. 5 **a** A typical artificial neuron. **b** ANN architecture with three input neurons in the input layer, two hidden neurons in the hidden layer and one output neuron



Shallow neural network

Artificial neural network (ANN)

The artificial neural network is based on mimicking and learning the data with the process of actual neurons in the vertebrates. In actual neurons, the impulses flow from the dendrites to the cell body, which contains a nucleus. Figure 5a shows a typical artificial neuron. The input information from different features in ANN is similar to dendrites, which are processed in a unit containing an evaluation function and an activation function. The input neurons and output neurons are linked through weights and biases. The simplest form of ANN is a perceptron with two input neurons and one output neuron. Figure 5b shows a simple ANN architecture. Multiple layers of perceptron can be used to create a shallow neural network with a hidden node layer between input and output nodes. A typical epoch consists of updating weights, and the iterations are continued till the final value of weights are achieved such that a weighted sum of information at nodes gives the desired result.

Deep machine learning

Convolutional neural network

The convolutional neural network is a deep neural network with high usage in computer vision for image recognition and pattern detection tasks. The convolutional neural network has three layers: convolutional, pooling, and fully connected. The convolutional and pooling layers can be in multitudines, but the end layer is always fully connected.

The convolutional layer consists of input data, a filter, and a feature map. For an image, the input can be the colour saturation value in each pixel, and the entire image is represented as a matrix of pixel information. The filter is a kernel, typically an array of weights that strides along with pixel information. These weights are multiplied with matrix value and added to form an output matrix. This output matrix is called a feature map or a convolved feature.

The pooling layer reduces dimensions by reducing the number of parameters. Pooling is primarily of two types max pooling and average pooling. In max-pooling, a filter is passed to the input through which the maximum feature value is extracted, and an output layer is formed. The average value of features is taken in average pooling, and the output layer is formed.

In the convolutional and pooling layers, a filter restricts the connectivity of all input features to output features; thus, these layers can be described as partially connected layers. The input nodes get translated through an activation function to an end output layer in the fully connected layer.

Deep belief network

A deep belief network suits an unsupervised and supervised machine learning task. A stack of Restricted Boltzmann Machine forms the deep belief network. Individual RBM performs non-linear operations to create an output that is utilized as input by the next RBM in line. RBMs are utilized for unsupervised learning; every layer of RBM is pre-trained to learn the features in an unsupervised manner. Further fine-tuning is done by the backpropagation technique, which utilizes gradient descent; this is used for classification and regression purposes.

Training algorithms

Backpropagation

Backpropagation is a popular supervised training algorithm technique for the neural network. A single iteration in the forward direction consists of taking a weighted sum and passing through an activation function to find a value at a hidden or output neuron. The backpropagation algorithm updates weights while traversing back from the output to the input in the same iteration. The weight updating equation between the output and the hidden neuron is given by equation (6) [19].

$$\Delta V_{jr} = \eta O_j \delta_r \quad \delta_r = O_r (1 - O_r) * (t_r - O_r) \quad (6)$$

where ‘j’ denotes the hidden node, ‘r’ corresponds to the output node, ΔV_{jr} is the weight change, O_j and O_r are activation values at hidden neurons and output neurons, respectively, t_r is the target value at the output neuron, and η is the learning rate. The weight updating equation between the hidden and input neuron is given by equation (7) [19].

$$\Delta W_{ij} = \eta x_i \delta_j \quad \delta_j = O_j * (1 - O_j) \sum V_{jr} * \delta_r \quad (7)$$

where ‘i’ denotes input neurons, x_i is the input value, $\sum V_{jr}$ is the updated weight between hidden neuron ‘j’ and output neuron ‘r’. These iterations are carried on until the target value error is minimized.

Levenberg–Marquardt (LM)

The Levenberg–Marquardt was developed in the early 1960s for solving non-linear least-square problems. The loss function is the least square error in neural network training problems. For the given data points x_i , y_i , a model curve with parameters α_i , β_i is to be found such that the sum of the square of the error between the actual data and the model curve is minimum. The minimization is an iterative process that requires an initialization value. The Levenberg–Marquardt algorithm combines the Gauss–Newton method and the gradient descent method. The gradient descent reduces the sum of squared error by updating the parameter along the steepest descent. In the Gauss–Newton method, the squared error sum is reduced by assuming the error function is quadratic at the local point and minimized. When the parameter of the model curve is near the optimum value, then LM acts more like gradient descent, and when the model curve parameter is away from its optimum value, LM works more like the Gauss–Newton method.

Restricted Boltzmann machine (RBM)

The restricted Boltzmann Machine is a training algorithm generally used in deep neural networks. It is derived from Boltzmann Machine. In RBM, the neurons are bifurcated into two groups: visible neurons, where input is given, and hidden neurons. In RBM, the visible and hidden neuron groups are forbidden to have any connection within the group; they only have an inter-group connection. This is in contrast to Boltzmann Machine, where all neurons are connected, making RBM’s easier to implement and more efficient to train. The hidden neuron group is used to capture the probability of features. Random weights are initialized, and

then Gibbs sampling is performed, simultaneously updating the weights of all neurons in a layer.

Performance and evaluation metrics

Applying an ML algorithm to solve a specific problem depends on how well it can compute the outcome. To quantify this performance, specific evaluation metrics are defined. This section summarises the top ten evaluation metrics used in all the literature studied for this paper. The performance of the literature survey in the subsequent sections is also highlighted on these ten-evaluation metrics; other scarcely used evaluation metrics are excluded.

A-20 INDEX

The A-20 index is the ratio of the M20 index to the total number of samples in the dataset.

$$A20 = \frac{M20}{M} \quad (8)$$

where M20 is the number of samples whose ratio of experimental value to the predicted value by algorithm does not deviate more than $\pm 20\%$ from 1, thus M20 is the number of samples whose ratio of experimental value to predicted value is between 0.8 and 1.2. ‘M’ is the total number of data samples. For a perfect performance evaluation, the A20 index should be equal to 1.

Mean absolute error (MAE)

The nomenclature for subsections from B to J remains as follows: x_i = Actual value of the i^{th} data. \hat{x}_i = The predicted value of the i^{th} data. x_{mean} = The average value of the actual data of all the samples. n =The total number of data samples.

The mean absolute error (MAE) is the statistical mean of the absolute difference between the predicted and actual values, i.e., the error. The mean absolute error can be described as given in equation (9) [25].

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{x}_i - x_i| \quad (9)$$

For a perfect correlation, the MAE should be 0.

Mean absolute percentage error (MAPE)

It is the statistical mean of the percentage of error with respect to the actual value of the data [25], given as follows.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{\hat{x}_i - x_i}{x_i} \right| * 100 \tag{10}$$

Mean squared error (MSE)

Mean squared error is the mean value of the error squared; this can be considered the second moment of the error about its origin. This helps in understanding the variance and bias of the estimates [25].

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{x}_i - x_i|^2 \tag{11}$$

Regression coefficient (R)

The regression coefficient indicates the correlation between the expected and actual outcomes. The regression coefficient lies between -1 to +1, where -1 regression indicates an asynchronous correlation, whereas +1 represents a perfect correlation.

Root mean squared error (RMSE)

Root mean squared error is a non-negative value describing the predictive performance of the model. RMSE is useful when large errors are undesirable since the errors are squared before taking their average. RMSE value of 0 indicates the predicted and actual values coincide. It is the square root of the mean square error and is given as follows [25].

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n |\hat{x}_i - x_i|^2} \tag{12}$$

RMSE-standard deviation ratio (RSR)

RSR is the ratio of root mean squared error to the standard deviation of the original data [25].

$$RSR = \frac{RMSE}{\sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - x_{mean})^2}} \tag{13}$$

R-squared

The R-squared value determines to what extent the predicted and actual values fit. An R² value of 1 indicates the model can perfectly predict the outcome, whereas an R² value of 0 means the model fails to predict [25].

$$R^2 = 1 - \frac{\sum_{i=1}^n (x_i - \hat{x}_i)^2}{\sum_{i=1}^n (x_i - \hat{x}_{mean})^2} \tag{14}$$

Variance account for (VAF)

Variance Account For is the ratio of the variance of the error to the variance of actual data. It is often expressed as a percentage. The ideal value of VAF should be around 100% [25].

$$VAF\% = \left(1 - \frac{var(x_i - \hat{x}_i)}{var(x_i)} \right) * 100 \tag{15}$$

Weighted mean absolute percentage error (WMAPE)

WMAPE is a variant of MAPE in which the error value of ith data is intensified appropriately according to the weightage of ith data. Thus, an error for a higher magnitude data has more significant weightage in the total error computation. In this, the weighted mean is taken in which the actual value is multiplied by the error and divided by the summation of actual data [25].

$$WMAPE = \frac{\sum_{i=1}^n \left| \frac{x_i - \hat{x}_i}{x_i} \right| * x_i}{\sum_{i=1}^n x_i} * 100 \tag{16}$$

AI for compressive strength prediction

This section bifurcates the algorithms into four subsections and discusses the individual techniques' methodologies, results and performance evaluation. The number of data points affects the model's performance depending upon the type of machine learning technique employed. Figure 6 shows the number of times different prominent features were used as input parameters in the AI-ML studies.

Algorithms based on rule

In algorithms based on the rule, the inputs are defined as suitable membership functions to which rules are applied and relationships are generated. The output of this relationship is de-fuzzified, and a crisp output is generated. This section discusses various papers utilizing fuzzy logic and a combination of fuzzy logic and neural network.

Iqtidar et al. [28] predicted the compressive strength of concrete blended with rice husk using ANFIS architecture, ANN, MNLR, and LR. A Gaussian membership function

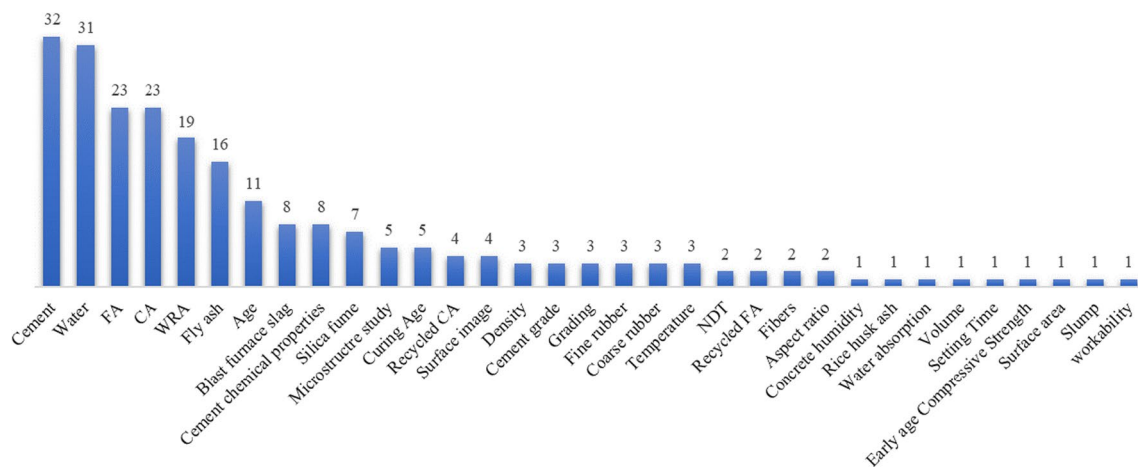


Fig. 6 Bar graph showing prominent features used in all studies. The water-reducing admixtures (superplasticizer, plasticizer, high-range water-reducers, and similar) are grouped as ‘WRA’. The non-destructive tests are grouped as ‘NDT’, chemical properties of cement are

grouped as ‘Cement chemical properties’. Microstructure studies, including microstructural imaging, are grouped as ‘Microstructural study’. Images other than microstructural images are grouped as ‘Surface Images’

was used to fuzzify the total of 192 input data. The input data was collected from Google Scopus, comprised of rice husk ash, age, superplasticizer, water, cement amount, and aggregates; this data was sent to nodes that multiplied the input with certain factors to generate weights for the next layer. The output weights were normalized to get the final weights; these weights were then multiplied to adaptive square nodes, a function of the input parameters. This was then sent to an aggregation layer to produce crisp output. ANFIS performed better than LR and MNL in this experiment, with an R^2 value of 0.89.

Tayfur et al. [29] predicted the compressive strength of high strength concrete using fuzzy logic and ANN. A total of 60 sets of data were produced using different binder contents prepared from the various percentages of silica fumes. Input data of silica fumes content, binder content, and age, and the output of compressive strength was fuzzified using the triangular membership function. Twenty-four fuzzy rules were defined, and the minimum operation was performed for inference to find the fuzzy output set.

The centroid method aggregated and de-fuzzified the outputs to get a crisp output. It was seen that fuzzy logic performed adequately better than ANN with a regression coefficient of 0.95. FL also outperformed ANN in MAE and RMSE evaluation metrics.

Abolpour et al. [30] compared the efficiency of fuzzy logic defined with the triangular membership function and Gaussian membership function in predicting the compressive strength of concrete. A data set of 1030 concrete mixtures from the University of California was used as the input for fuzzification. Input variables include weight percent of cement, water, blast furnace slag, fly

ash, superplasticizer, fine and coarse aggregate, and concrete age in days, the output function being the concrete compressive strength. 897 fuzzy rules and five linguistic values were defined. It was seen that the triangular membership function had a MAE of 11.72% and performed better than the Gaussian membership function, with a MAE of 13.27%.

Sarıdemir et al. [31] predicted the long-term effect of GGBFS on the compressive strength of the concrete using fuzzy logic and ANN. The compressive strength was obtained from the literature data and was verified using fuzzy logic and ANN. The input data include the specimen’s age, water, aggregate, Portland cement, and GGBS. The input data was fuzzified using the triangular membership function. The product method was used as the inference operator, and the weighted average method was used for defuzzification. The FL correlated with the actual compressive strength with an R^2 value of 0.99 and 0.97 for testing and training.

Özcan et al. [32] compared ANN and fuzzy logic for predicting the compressive strength of silica fume concrete. Forty-eight concrete mixtures were produced using four different water-cement ratios, three different cement dosages, and three partial silica fume replacement ratios. A triangular membership function fuzzified the inputs. Nine rules for compressive strength were formed. Mamdani’s inference system was used to get the crisp output. The experimental compressive strength was compared with the predicted compressive strength. The fuzzy logic showed promising outcomes for predicting the compressive strength of silica fume concrete with an R^2 value of 0.93.

Table 1 Summary of Rule-based algorithms w.r.t features and data points

References	Algorithm	Data source	No. of data	Used features
[28]	ANFIS	Google Scopus	192	Water, amount of cement, age, aggregates, rice husk ash, superplasticizer
[29]	Fuzzy logic	Lab Experiment	60	Age, Silica fumes content, binder content
[30]	Fuzzy logic	Literature Data	1030	Weight percent of cement, superplasticizer, blast furnace slag, fine aggregate, coarse aggregate, age of the concrete, fly ash, water
[31]	Fuzzy logic	Literature Data	50	Blaine fineness, %C ₃ S, %SO ₃ , % total alkali
[32]	Fuzzy logic	Lab Experiment	48	Aggregate, amount of cement, water content, plasticizer, amount of silica fume replacement
[33]	Fuzzy logic	Literature Data	180	Sand, crushed stone II, day, Portland cement, high range water reducing agent replacement ratio, water, fly ash replacement ratio, crushed-stone I, CaO
[34]	Fuzzy logic	Cement plant	50	Alkali Strength, SO ₃ , Blaine, C ₃ S

Topcu and Sarıdemir [33] used ANN and fuzzy logic to predict the compressive strength of concrete blended with different fly ashes having high and low lime percentages for 7, 28, and 90 days. The triangular membership function fuzzified the inputs. One hundred eighty experimental data were used, and Sugeno-type fuzzy rules were established to infer input and output. The product method was used as an inference operator. The crisp output showed a very high correlation with the experimental output, with an R² value of 0.99 for training and testing.

Akkurt et al. [34] developed a fuzzy logic model to predict 28th-day cement compressive strength and compared it with the ANN method. The data used for the training algorithm was taken from a cement plant in Izmir, Turkey. A triangular membership function fuzzifies fifty sets of input data. The product operator defined the inference system using the Mamdani fuzzy rules. ANN was constructed with 4-4-1 architecture, suggesting four neurons in the input layer, 4 in the hidden layer, and 1 in the output layer. The average error for the fuzzy logic model was 2.69%; this could have been improved by having more than four input parameters. ANN model performed better with an RMSE value of 1.7. Table 1 summarises the rule-based algorithms reviewed in this paper with respect to the number of samples used for training and the input features used in training. Table 2

summarises the rule-based algorithms with respect to the evaluation metrics and their performance.

Classical ML and clustering

In the classical machine learning and clustering section, classification and regression algorithms like support vector machine (SVM), support vector regression (SVR), various regression techniques, optimization techniques like genetic algorithms, particle swarm optimization, ant colony optimization and clustering techniques like the random forest, tree-based methods and various other similar techniques are included. The description of the methods, the features used, the number of data samples and the data source is summarised in Table 3. The summary of performance evaluation is summarised in Table 4.

Güçlüer et al. [35] compared the performance of ANN, DT, SVM and LR in predicting concrete compressive strength. They utilized results from the non-destructive test of ultrasonic pulse velocity to find the compressive strength and utilized the outputs for training the algorithms. NDT was used to test 522 concrete cubes of standard 150mm size with compressive strength ranging from 35.07 MPa to 53.44 MPa. The M5P decision tree developed by Wang and Witten was used in this study.

Table 2 Summary of Rule-based algorithms with respect to the evaluation metrics

Reference	Algorithm	Evaluation and performance (training, testing)
[28]	ANFIS	R ² =0.89
[29]	Fuzzy logic	RMSE (Train, Test)=(2.99, 4.31) MAE (Train, Test)=(2.17, 3.68) R=0.95
[30]	Fuzzy logic	MAE (Gaussian)=13.27% MAE (Triangular)=11.72%
[31]	Fuzzy logic	RMSE (Train, Test)=(1.52,3.38) MAPE (Train, Test)=(3.26,6.08) R ² (Train, Test)=(0.99,0.97)
[32]	Fuzzy logic	R ² =0.93
[33]	Fuzzy logic	RMSE (Train, Test)=(1.72,2.02) MAE (Train, Test)=(3.6, 3.38) R ² (Train, Test)=(0.9989, 0.9986)
[34]	Fuzzy logic	RMSE=1.83 MAE=2.69

Table 3 Summary of Classical ML and Clustering algorithms w.r.t features and data points

References	Algorithm	Data source	No. of data	Used features
[10]	GEP	Cement Plant	104	% sieve residue on 45 & 90 micron, %Fe ₂ O ₃ , %K ₂ O, % SiO ₂ , %MgO, specific surface, %Free lime, %SO ₃ , Modulus of Alumina, Litre weight, %Ignition loss, setting time, 1–2–7 days strength, %Al ₂ O ₃ , % composite, %CaO
[21]	Gradient boost, XGBoost, RF, Adaboost, DT, KNN, LR, Ridge reg., SVR, Lasso reg	Literature data	220	w/c ratio, silica fume, fibre volume fraction, coarse aggregate size, sand to aggregate ratio, aspect ratio, superplasticizer
[28]	MNLR, LR	Google Scopus	192	Water, aggregates, amount of cement, age, rice husk ash, superplasticizer
[35]	DT, SVM, LR	University of California dataset repository, Irvine	1030	CA to FA ratio, Fly ash—binder ratio, Blast furnace slag, superplasticizer to binder ratio, Age, Compressive strength
[36]	LSSVM coupled with SA, Genetic Programming	University of California dataset repository, Irvine	1030	Fly ash, superplasticizer, cement, water, blast furnace slag, coarse aggregate, curing age, fine aggregate
[37]	Bagging, AdaBoost, GEP, DT	Literature Data	1030	Water, cement, fly ash, superplasticizer, fine aggregate, blast furnace slag, coarse aggregate, age
[38]	DT, Bagging, Gradient Boost	Literature Data	207	Temperature, Super Plasticizer, Water, Coarse Aggregate, Fine Aggregate, Fly Ash, Silica Fume, Cement, Nano Silica
[39]	Bagging, RF, Boosting trees, SVR, GPR	Literature Data	166	CA, fine rubber, FA, coarse rubber, superplasticizer, slag, water, cement, silica fume, fly ash
[40]	MPMR, RVM, GP	Literature Data	112	Fine aggregate, coarse aggregate, water, water-binder ratio, cement, fly ash
[41]	SVM, MLR	Lab Tests	124	Recycled CA, Natural sand, cement, normal CA, superplasticizer, water, recycled sand
[42]	MLR, XGB	Lab Tests	270	Superplasticizer, cement, Natural CA (different %), Recycled CA (different %), age, w/c ratio
[43]	SVM, RF, DT, AdaBoost, KNN	Literature Data	424	Water binder ratio, particle size distribution, sand, Cement grade, plasticizer, age
[44]	AdaBoost, SVM	Literature Data	1030	Fine aggregate, curing time, Blast-furnace slag, Water, Coarse aggregate, Super-plasticizer, Fly ash, cement
[45]	GPR, SVM	Literature Data	83	Fly ash (FA), Silica fume (SF), cement (C), superplasticizers (SP), sand (S), coarse aggregate (CA), water (W), Fibre (F), aspect ratio (AR)
[46]	KNN, RT, RF	Literature Data	457	Cement, w/c ratio, NA-C (% of coarse aggregate from total aggregate), RB-F (% of fine rubber from total aggregate), NA-F (% of fine aggregate from total aggregate), and RB-C (% coarse rubber from total aggregate)
[47]	GPR, MARS, MPMR	Literature Data	1030	Fine aggregate and age in days, fly ash, cement, water, blast furnace slag, superplasticizer, coarse aggregate
[48]	DT, RF	Literature Data	49	Workability, water-cement ratio, cement content, water content, curing ages
[49]	GEP	Literature Data	70	T (°C—curing temperature), Al ₂ O ₃ %, TiO ₂ %, K ₂ O%, Cr ₂ O ₃ %, CA (day(s)—curing age), SA (m ² /kg Surface area of cementitious material)
[50]	MARS, M5 Tree model, SVR	Literature Data	46	Oven-dry density, foamed volume, water-to-binder ratio, cement content
[51]	LR, MLR, NLR	Lab Tests	28	Surface images

Table 3 (continued)

References	Algorithm	Data source	No. of data	Used features
[52]	DMEP	Literature Data	50	Blaine fineness, %SO ₃ , % total alkali, %C ₃ S
[53]	SVM	Lab Tests	74	Recycled fine aggregate replacement ratio, water-cement ratio, fly ash replacement ratio, recycled coarse aggregate replacement ratio
[54]	MLR, GPR	Lab Tests	180 images	Microtomographic images, eigenvalues

DT algorithm provided the best results with a regression coefficient of 0.86, MAE of 2.59 MPa and RMSE of 3.77 for 28 days of compressive strength.

Salami et al. [36] predicted the compressive strength of concrete having ternary-blend using the least square support vector machine (LSSVM) coupled with simulated annealing. A comparison was made with the genetic programming (GP) method used as a benchmark for performance. The dataset from the University of California dataset repository, Irvine, was used, and 1030 data values were used to train and test algorithms. The ternary blend consists of cement, blast furnace slag and fly ash. LSSVM coupled with simulated annealing outperformed GP with R² of 0.982 and 0.954 for training and testing, respectively, while GP had R² of 0.86 and 0.89 for training and testing, respectively.

Kang et al. [21] predicted steel-reinforced fibre concrete's compressive strength and flexural strength using 11 machine learning techniques and compared their performance. These techniques were the Gradient boosting regressor, XGBoost, RF regressor, AdaBoost, DT, KNN, LR, Ridge, SVR, Lasso, and MLP. The data for training and testing the algorithms were taken from different works of literature. It was seen that the water-cement ratio and the silica fumes used in concrete were the most prominent factors affecting the prediction of compressive strength. In contrast, the volume of steel fibre and the content of silica fume in concrete was more responsible for the prediction of flexural strength. It was seen that the extreme Gradient boosting (XGBoost), RF regressor and DT regressor performed better than the other soft computing techniques.

Ahmad et al. [37] used supervised machine learning techniques of bagging, GEP, AdaBoost and DT and compared their performance to predict the compressive strength of concrete blended with cementitious material. Statistical analysis was performed, and evaluation metrics RMSE, R², MAE and MSE were determined. Along with this, the validity of algorithms was validated using K-fold cross-validation. The data of 1030 sets of values were distributed randomly and split into ten groups. Of the ten groups, nine were used for training and one for validation. This was repeated ten times, and the average of evaluation metrics was taken to get the performance of the algorithms. Bagging performed

the best in both performance evaluation methods, followed by AdaBoost, GEP and DT.

Ahmad et al. [38] compared the performance of supervised machine learning algorithms to predict the compressive strength of concrete at high temperatures. Concrete subjected to high temperatures affects its internal structure and compressive strength significantly. The ML techniques adopted in this paper are bagging, DT, ANN and Gradient boosting. This study used nine input parameters: fine aggregates, coarse aggregates, temperature, silica fumes, cement, fly ash, nano-silica, water, and superplasticizer. DT and ANN were used as individual algorithms for forecasting the compressive strength at high temperatures, while Bagging and Gradient boosting were used as an ensemble. The statistical evaluation and K-fold cross-validation determined that bagging performed better than other algorithms with an R² value of 0.9, followed by Gradient boosting, DT and ANN, with an R² value of 0.89, 0.84 and 0.82, respectively.

Kova et al. [39] developed a model for predicting the compressive strength of self-compacting rubberized concrete using MLP-ANN, MLP-ANN ensemble, bagging, RF, Boosting trees, SVR and GPR. The total number of inputs was 10, whereas the output was single (compressive strength). For the bagging method, max trees were limited to 500; the minimum leaf size was kept between 2 and 15. The learning curve for bagging was seen to be saturated at 269 trees. For the RF method, the maximum trees were limited to 500, and the value of subsets for splitting was kept between 2 and 9. In boosting trees, the maximum number of trees was limited to 100 due to overtraining. The SVR model was analysed using linear, RBF and sigmoid kernel; the RBF kernel performed the best out of the three. In the GPR method, the standardization was done using a Z-score in which the mean value was made 0, the variance was made 1, and constant base function models were analysed.

The MNLR and LR technique by Iqtidar et al. [25] was used to predict rice husk ash concrete compressive strength was below average compared with other models used; LR performed the worst with an R² value of 0.63 and MNLR having an R² value of 0.7.

Khursheed et al. [40] predicted the compressive strength of fly ash concrete using MPMR, RVM, GP, ENN and ELM. For MPMR, the Gaussian noise was kept at 0.002, and the other tuning parameter was kept at 1.2; these were

Table 4 Summary of Classical ML and Clustering-based algorithms with respect to the evaluation metrics

References	Algorithm	Evaluation and Performance (training, testing)
[10]	GEP	$R^2 = 0.775$
[21]	Gradient Boosting XGBoost RF AdaBoost DT KNN LR Ridge reg. SVR Lasso reg. MNLR LR DT SVM	RMSE = 1.51 1.63 1.84 2.07 2.2 2.43 2.53 2.65 2.82 3.35 $R^2 = 0.7$ $R^2 = 0.63$ RMSE = 3.77 4.03 4.02 RMSE (train, test) = 2.23, 3.335 5.918, 4.662 RMSE = 4.54 6.88 7.12 7.57 RMSE = 10.79 7.81
[28]		
[35]		MAE = 2.59 2.93 2.9
[36]	LSSVM – Coupled with SA Genetic Programming	R^2 (train, test) = 0.982, 0.954 0.859, 0.894
[37]	Bagging AdaBoost GEP DT DT Bagging Gradient Boost	$R^2 = 0.92$ MSE = 20.57 47.38 50.69 57.3 MSE = 112.23 61.08 85.47 MAPE = 18.81
[38]		MAE = 3.26 5.13 5.24 5.88 MAE = 7.54 5.65 6.93
[39]	Bagging RF Boosting Tree SVR GPR MPMR	MAE = 6.05 5.7174 5.4248 3.5352 3.1022 MAPE = 9.499, 16.509
[40]		$R^2 = -0.9841, 0.9006$ 0.9781, 0.8836 0.9643, 0.2905
[41]	RVM GP SVM MLR	0.032, 0.084 0.042, 0.287 RMSE = 0.041, 0.042 0.065, 0.062
[42]	MLR XGB	RMSE = 4.521, 3.3883 4.5701, 3.3414 $R^2 = 0.898, 0.912$ 0.892, 0.917
		MAE = 1.18 1.25 1.3 1.7 1.79 1.66 1.8 1.95 2.08 2.69
		MAPE = 0.042, 0.105 WMAPE = 0.042, 0.105 VAF = 98.58, 89.608 RSR = 0.119, 0.351 0.054, 0.154 0.07, 0.235 $R = 0.98, 0.96$ 0.69, 0.64 MAPE = 5.1137, 5.0604 5.2355, 5.0008 VAF = 0.7685, 0.8055 0.7799, 0.8092

Table 4 (continued)

References	Algorithm	Evaluation and Performance (training, testing)
[43]	SVM	RMSE = 17.736, 17.667
	RF	3.359, 5.24
	DT	9.256, 8.439
	AdaBoost	1.305, 5.291
	KNN	7.527, 7.789
[44]	AdaBoost	RMSE = 2.2
	SVM	6.28
[45]	GPR (Pearson kernel)	RMSE = 1.702, 6.228
	GPR (RBF kernel)	1.4179, 6.3556
	SVM (Pearson kernel)	2.3705, 6.1318
	SVM (RBF kernel)	2.1123, 6.3898
[46]	KNN	RMSE = 8.725
	RT	7.549
	RF	8.058
[47]	GPR	RMSE = 0.01199, 0.06375
	MARS	0.05477, 0.05846
	MPMR	0.01199, 0.07358
[48]	DT	RMSE = 2.3094, 2.6627
	RF	1.2232, 1.6354
[49]	GEP	R = 0.95
[50]	MARS	R = 0.96
	M5 Tree Model	0.9
	SVR	0.9214
[51]	LR	RMSE = 2.616
	MLR	2.424
	NLR	2.418
[52]	DMEP	RMSE = 1.47
[53]	SVM	Relative Absolute Error = 4.35
[4]	MLR	MAE = 4.952
	GPR	4.459
		RMSE = 17.736, 17.667
		3.359, 5.24
		9.256, 8.439
		1.305, 5.291
		7.527, 7.789
		RMSE = 2.2
		6.28
		RMSE = 1.702, 6.228
		1.4179, 6.3556
		2.3705, 6.1318
		2.1123, 6.3898
		RMSE = 8.725
		7.549
		8.058
		RMSE = 0.01199, 0.06375
		0.05477, 0.05846
		0.01199, 0.07358
		RMSE = 2.3094, 2.6627
		1.2232, 1.6354
		R = 0.95
		R = 0.96
		0.9
		0.9214
		RMSE = 2.616
		2.424
		2.418
		RMSE = 1.47
		Relative Absolute Error = 4.35
		MAE = 4.952
		4.459
		R ² = 0.3786, 0.4203
		0.9768, 0.9439
		0.8186, 0.8526
		0.9965, 0.9473
		0.8807, 0.874
		MAE = 1.64
		4.44
		R = 0.992, 0.885
		0.9946, 0.8819
		0.9848, 0.8897
		0.988, 0.8816
		MAE = 5.847
		5.216
		6.087
		MAE = 0.00101, 0.04292
		0.03978, 0.04399
		0.00099, 0.05659
		R ² = 0.8185, 0.6889
		0.9625, 0.838
		R ² = 0.9391
		0.9482
		0.9463
		MAE = 2.18
		MAPE = 0.458, 0.434
		0.074, 0.109
		0.218, 0.189
		0.017, 0.097
		0.169, 0.143
		R ² = 0.982
		0.855
		14.95
		MSE = 2.896, 38.786
		2.0103, 40.3935
		5.6192, 37.5991
		4.462, 40.8293
		R = 0.8803
		0.914
		0.9
		R = 0.9984, 0.9485
		0.9652, 0.957
		0.9984, 0.9352

determined by the trial and error method. In the RVM model, the width of the radial basis kernel was determined using posterior modelling scrutiny, and the free parameter in the RBF kernel was determined as 0.04. For GP, the size of the population was kept at 800, the number of generations was kept at 400, frequency of generations was 50. The anxiety parameter in the ENN model was kept at 0.34, and the confidence parameter was kept at 0.66. The ELM model worked optimistically at six hidden neurons in a single layer, and the radial basis function was used as an activation function. The MPMR performed the best, followed by RVM, GP, ENN and ELM in decreasing order.

Salimbahrami and Shakeri [41] compared the performance to predict the compressive strength of recycled aggregate concrete using SVM and ANN. These were further compared with MLR using K-fold cross-validation. In this study, natural aggregate concrete, the control specimen and recycled aggregate concrete (with and without fibre) are produced in the lab and tested for 7 and 28-day strength. Workability was improved using a poly-carboxylic ether-based superplasticizer. In natural aggregate, the fine aggregate is procured from river sand, and coarse aggregates are crushed gravels with a maximum of 19mm in size. The influence parameter of the RBF kernel for SVM was kept at 0.008, the number of support vectors was 48, and the regularisation parameter C was kept at 1000. A multiple linear regression method was used to model the relation between output compressive strength and six input parameters, as described in Table 3. The input layer of ANN consists of 6 neurons, the optimum neurons in the hidden layer was determined to be 7, and the log-sigmoid function was used as an activation function and backpropagation algorithm for training. SVM and ANN performed the best, with SVM having a marginal advantage over ANN, and MLR performed moderately fine.

Rizvon and Jayakumar [42] predicted the compressive strength of recycled coarse aggregate concrete and correlated it with the hardened features of the concrete. Five replacement ratios of coarse aggregate (0 to 100%) were determined, and varying water to cement ratios (0.3, 0.4 and 0.48) along with superplasticizer were used to prepare 15 mixes. Multi-linear regression was used to predict the compressive strength with six independent variables. XGB model was also employed to predict the compressive strength of cylinder and cube strength. The XGB model was found to be 0.5% better than MLR in predicting the compressive strength of the cylinder.

Asteris et al. [43] predicted the compressive strength of cement-based mortar using KNN, SVM, RF, DT and AdaBoost. The optimum value of 'k' in KNN was determined as 3, and the Minkowski function was used in training. This study used the v-SVM, and a sigmoid kernel was used to build SVM. The number of trees for RF was kept at 10, and the split subset limit was greater than 5. The split subset

limit for DT was greater than 6, and the minimum number of instances in DT was kept at 2, with a maximum tree depth of five. The number of estimators for AdaBoost was kept at 30. AdaBoost performed the best with an R^2 value of 0.95 for testing, followed by RF, KNN, DT and SVM with testing R^2 values of 0.94, 0.87, 0.85 and 0.42, respectively.

Feng et al. [44] compared the performance of AdaBoost with ANN and SVM in the prediction of the compressive strength of concrete. For implementing AdaBoost, the 1030 input and output parameters data was collected, a strong learner was generated, the learner was validated, and finally, the application was made to get the output of compressive strength. Classification and regression tree were used to generate the weak learner, and it was integrated by the median of weak learners. 90% of the data was used as a training set, and 10% was used as a testing set. The data was also trained using ANN and SVM. It was seen that AdaBoost performed better than ANN and SVM, with an R^2 value of 0.982, followed by ANN with 0.903 and SVM with 0.855.

Sihag et al. [45] estimated the 28-day compressive strength of high strength concrete using GPR, SVM and ANN. The GPR and SVM had two models each, based on the kernel functions of Pearson kernel function and the Radial basis function. In GPR, the Gaussian noise was kept at 0.1; in SVM, the regularisation parameter C was kept at 10. A sensitivity analysis was performed by eliminating one input in every trial, revealing that cement and silica fumes are the most critical parameters in determining the output. The statistical analysis showed that ANN has better performance than GPR and SVM. Also, it was found that the Pearson kernel performed better than the RBF kernel.

Nyarko et al. [46] predicted the compressive strength of rubberized concrete using KNN, regression trees (RT), RF and ANN. A database of 457 data from the literature was used for training and testing purposes. For KNN, five-fold cross-validation for various values of 'K' ranging from 1 to 20 was performed, and the optimum value of 'K' was determined as 3. For RT, the minimum parent size was kept between 1 and 40, and five-fold cross-validation determined the optimum minimum parent size of 3. The optimum number of trees to be included in the RF model is 21. For ANN, three hidden layers were used with 9, 3 and 2 hidden neurons, respectively. A tan sigmoidal function was used as an activation function, and the Levenberg Marquardt training algorithm was used to update the weights. The ANN outperformed other methods in the testing phase with an R-value of 0.978, followed by RT, RF and KNN with an R-value of 0.914, 0.9 and 0.88, respectively.

Dutta et al. [47] compared the performance of GPR, MARS, and MPMR in predicting the compressive strength of concrete. A total of 1030 data sets were used in this study; 80% were used for training and 20% for testing. For GPR, the Gaussian noise was kept at 0.1, and the sigma parameter

was kept at 0.001. The MARS model was built using 70 basis functions. For MPMR, a radial basis function is used. It is seen that the MARS model performs better with a testing regression value of 0.957, followed by GPR and MPMR with regression values of 0.9485 and 0.9352, respectively.

Chopra et al. [48] compared the performance of DT and ANN in predicting the compressive strength of the concrete at 28, 56 and 91 days. A decision tree model with 33 nodes was built. The number of observations used to build the RF model was 33, and for ANN, an architecture of 6 inputs, ten hidden nodes in the first hidden layer and one output is made. Performance evaluation of the model suggested that ANN performed better than the other two, with an R2 value of 0.95 in the testing phase, followed by 0.84 and 0.69 for RF and DT, respectively.

Mirzahosseini et al. [49] predicted the compressive strength of glass cullet-modified concrete using GEP. For this purpose, 70 numbers of 50mm mortar cubes with glass powder and constant water-cement and sand to cementitious ratios and varying glass powder replacement ratios were cast in the laboratory. Compressive strength was determined at 1, 7, 28 and 91 days. The study was conducted using two GEP models; the first model contained two curing temperatures of 23 and 50 degrees Celcius, for which all test results were included. In the second model, three curing temperatures of 10, 23 and 50 were considered valid on 45 out of 70 tests. Statistical performance evaluation suggested GEP performed exceptionally well with a regression value of 0.95.

Yaseen et al. [50] predicted the compressive strength of lightweight foamed concrete using ELM and compared its performance with MARS, M5 Tree model and SVR. Forty-six sets of input and output data were gathered from the literature study. The MARS model was built using the cubic function for better smoothness; a recursive partitioning regression was used to optimize function approximation. The M5 tree model utilized a split value of 2, a value of smoothening of 15 and 0.05 as the splitting threshold. The SVR model was built with an RBF kernel. The ELM model comprised three layers in the network architecture. Sigmoid was used as an activation function. An optimization study was done to find the optimal number of hidden neurons (between 50 and 150); thus, the model was run 500 times and mean square error was used to evaluate the performance. Performance evaluation of all the models suggested ELM performed the best out of the other models.

Basyigit et al. [51] predicted the compressive strength of concrete by image processing trained using LR, MLR, and NLR. For this, 28 cubes of 100mm were cast with seven different water to cement ratios (0.37 to 0.79), four concrete cubes for each ratio, three cubes were tested for compressive strength, fourth was used to take the image. The images were cropped into 2350x2350 pixels, and a histogram of the pixel value was plotted. The threshold values for

aggregate, cement and air gap were defined. The theoretical and obtained values were trained using different methods to obtain the best regression. Multi-linear regression and Non-linear regression performed regression almost with equal efficacy.

The compressive strength prediction of cement by double-layer MEP was proposed by Akkurt et al. [52]; in double-layer MEP, the chromosomes, instead of a single layer, are placed in a double layer. Thus, the chromosomes are capable of holding more information and expressions. Further, crossover, mutation and fitness evaluation are performed. The performance evaluation of all the methods suggested that double-layer MEP performed better than others, with an RMSE value of 1.47.

Baykasoglu et al. [10] predicted the cement compressive strength using GEP and NN and compared it with regression analysis. A four-month (104 days) data was collected from a cement plant in Adiyaman, Turkey. Nineteen input features were used in this study, as described in the table below; the output was considered as 28 days strength. Gene expression programming was executed with an initial population size of 50, and between 3000 and 20000 generations were generated. Subsets of 19 input features are defined with relation, and their performance is checked; the best regression value for GEP is 0.775.

Shallow neural network

This section discusses the shallow neural network models used by researchers to predict the compressive strength of concrete. Shallow neural networks are characterized by a single layer of hidden node layer, while deep neural networks have more than a single hidden node layer [55]. This section discusses neural network models like ANN, MLP, MFNN and similar algorithms. The description of the methods, the features used, the number of data samples and the data source is summarised in Table 5. The summary of performance evaluation is summarised in Table 6.

A three-layered feed-forward artificial neural network was used by Güçlüer et al. [35] to predict the compressive strength of concrete. The hidden layer consisted of 3 nodes, and the sigmoid function was used as an activation function. They used a learning rate of 0.3. ANN performed second best after DT with a regression value of 0.856.

Kang et al. [21] used MLP to predict the compressive strength of steel fibre-reinforced concrete. MLP was modelled with eight inputs and two hidden layers. Keras was used to build the MLP model and backpropagation as the training function. ReLu was used as the activation function for the internal hidden layer, and the output layer utilized the Adam function as the activation function. MLP was performed with an RMSE value of 3.41 MPa and a MAE of 2.79 MPa.

Table 5 Summary of Shallow Neural Network Algorithms w.r.t features and data points

References	Algorithm	Data Source	No. of Data	Used features
[9]	ANN	Literature Data	209	Ultrasonic pulse velocity, rebound hammer
[10]	NN	Cement Plant	104	% sieve residue on 45 & 90 micron, %Fe ₂ O ₃ , %K ₂ O, % SiO ₂ , %MgO, specific surface, %Free lime, %SO ₃ , Modulus of Alumina, Litre weight, %Ignition loss, setting time, 1–2-7 days strength, %Al ₂ O ₃ , % composite, %CaO
[20]	MFNN	Cement Plant & Lab Tests	100	Sand-aggregate ratio, CA max size, cement dosage, admixture dosage, Cement grade, fineness modulus of sand, slump, water-cement ratio, water dosage, admixture effect, aggregate-cement ratio
[21]	MLP	Literature data	220	w/c ratio, silica fume, fibre volume fraction, coarse aggregate size, sand to aggregate ratio, aspect ratio, super-plasticizer
[28]	ANN	Google Scopus	192	Water, aggregates, amount of cement, age, rice husk ash, superplasticizer
[29]	ANN	Lab Tests	60	Silica fumes content, age, binder content
[31]	ANN	Literature Data	50	Blaine fineness, %SO ₃ , %C ₃ S, % total alkali
[32]	ANN	Lab Experiment	48	Amount of cement, aggregate, plasticizer, water content, amount of silica fume replacement
[33]	ANN	Literature Data	180	Sand, crushed stone II, day, Portland cement, high range water reducing agent replacement ratio, water, fly ash replacement ratio, crushed-stone I, CaO
[35]	ANN	University of California dataset repository, Irvine	1030	CA to FA ratio, Fly ash—binder ratio, Blast furnace slag, superplasticizer to binder ratio, Age, Compressive strength
[38]	ANN	Literature Data	207	Temperature, Super Plasticizer, Water, Coarse Aggregate, Fine Aggregate, Fly Ash, Silica Fume, Cement, Nano Silica
[39]	MLP-ANN, MLP-ANN ensemble	Literature Data	166	CA, fine rubber, FA, coarse rubber, superplasticizer, slag, water, cement, silica fume, fly ash
[40]	ENN, ELM	Literature Data	112	fine aggregate, coarse aggregate, water, water-binder ratio, cement, fly ash
[41]	ANN	Lab Tests	124	Recycled CA, natural sand, cement, normal CA, superplasticizer, water, recycled sand
[44]	ANN	University of California dataset repository, Irvine	1030	Fine aggregate, curing time, Blast-furnace slag, Water, Coarse aggregate, Super-plasticizer, Fly ash, Cement
[45]	ANN	Literature Data	83	Fly ash (FA), Silica fume (SF), cement (C), superplasticizers (SP), sand (S), coarse aggregate (CA), water (W), Fibre (F), aspect ratio (AR)

Table 5 (continued)

References	Algorithm	Data Source	No. of Data	Used features
[46]	ANN	Literature Data	457	Cement, w/c ratio, NA-C (% of coarse aggregate from total aggregate), RB-F (% of fine rubber from total aggregate), NA-F (% of fine aggregate from total aggregate), and RB-C (% coarse rubber from total aggregate)
[48]	ANN	Literature Data	49	Workability, water-cement ratio, cement content, water content, curing ages
[50]	ELM	Literature Data	46	Oven-dry density, foamed volume, water-to-binder ratio, cement content
[53]	BPNN	Lab Tests	74	Recycled fine aggregate replacement ratio, water-cement ratio, fly ash replacement ratio, recycled coarse aggregate replacement ratio
[54]	ANN	Lab Tests	180 images	Microtomographic images, eigenvalues
[56]	Hybrid ensemble model	University of California dataset repository, Irvine	1030	Water to binder ratio, Age, Blast furnace slag, Fly ash—binder ratio, CA to FA ratio, superplasticizer to binder ratio
[57]	ANN	Lab Tests	27 images	Mean, median, standard deviation
[58]	BPNN	Literature Data	139	Recycled coarse aggregate, Water-cement ratio, Water-total material ratio, Natural coarse aggregate, Water absorption, Fine aggregate
[59]	ANN	Lab Tests	144 images	Statistical parameters of the greyscale pixel matrix
[60]	ANN	Lab Tests	60 samples	Surface images
[61]	Feed-forward backpropagation NN	Lab Tests	130	Total pore length, hydrated part area, pore area ratio, total dendrite length and average roundness, un-hydrated part area
[62]	GA-ANN	Cement Plant	150	Surface area, the chemical composition of cement, particle size distribution, C ₃ S and silicate moduli
[63]	ANN	Literature Data	263	Curing time, NaOH/Na ₂ SiO ₃ , AL/Fly ash, fly ash/Aggregate, Curing temperature, NaOH concentration
[64]	ANN, Second-order ANN	Literature Data	177	Sand-cement ratio, density, water-cement ratio

Asteris et al. [56] developed a hybrid ensemble model to predict the compressive strength of concrete. The hybrid ensemble model (HENSM) was constructed by training four conventional machine learning algorithms using ANN. The conventional ML models include MARS-linear, MARS non-linear, GPR and MPMR. The University of California dataset repository was used to get 1030 sets of 6 inputs and one output of compressive strength. Around 721 data sets (approx. 70%) were utilized for training and the remaining model for validation. For ANN, only one hidden layer was considered with 20 neurons, achieving the best result. The maximum value of the basis function for MARS – linear and non-linear methods was taken as 29. The non-linear method had a cubic function. For GPR, the width of the radial basis function was determined by trial as 0.4, and the

Gaussian noise was taken as 0.07. For MPMR, the radial basis width was determined as 0.7, and the noise was determined as 0.0047. The HENSM model outperformed all other individual ML algorithms to achieve an R² value of 0.99 and 0.89 in training and testing, respectively. A score analysis was performed to rank the performance of all models, and HENSM scored highest, followed by GPR, MARS – L, ANN, MARS – C and MPMR in decreasing order.

The MLP-ANN and MLP-ANN ensemble were utilized by Kova et al. [39] to predict the compressive strength of self-compacting rubberized concrete. The MLP-ANN algorithm was trained using the LM training algorithm; the input layer consists of 10 nodes, while the output layer has one node of compressive strength. The optimum number of nodes in the hidden layer was found by trial and error

Table 6 Summary of Shallow Neural Network algorithms with respect to the evaluation metrics

Reference	Algorithm	Evaluation and performance (training, testing)	$R^2 = 0.978$	a-20 index = 100
[9]	ANN	RMSE = 1.4678		
[10]	NN	$R^2 = 0.695$		
[20]	MFNN	MAE = 1.6		
[21]	MLP	RMSE = 3.41	MAE = 2.79	
[28]	ANN	$R^2 = 0.98$		
[29]	ANN	RMSE = 4.74, 3.71	MAE = 3.91, 3.21	R = 0.83
[31]	ANN	RMSE = 1.036, 2.511	R2 = 0.996, 0.981	MAPE = 2.433, 4.688
[30]	ANN	RMSE = 1.71, 2.81	MAE = 3.325, 5.067	R2 = 0.999, 0.997
[33]	ANN	RMSE = 4.48	MAE = 3.42	R = 0.85
[38]	ANN	RMSE = 11.03	MAE = 9.15	MSE = 121.66
[39]	MLP-ANN	RMSE = 7.44	MAE = 5.54	MAPE = 17.68
	MLP-ANN ensemble	3.6888	2.8099	8.54
[40]	ENN	RMSE = 0.109, 0.25	MAPE = 47.313, 37.064	VAF = 76.518, -23.848
	ELM	0.123, 0.126	0.6972, 0.7276	69.786, 68.421
[41]	ANN	RMSE = 0.042, 0.043	68.187, 38.05	0.55, 0.563
[44]	ANN	RMSE = 5.14	MAE = 3.41	MAPE = 9.77
[45]	ANN	RMSE = 4.327, 6.128	R = 0.971, 0.909	MSE = 18.728, 37.557
[46]	ANN	RMSE = 3.74	MAE = 2.835	R = 0.9779
[48]	ANN	RMSE = 0.7483, 0.8771	$R^2 = 0.9713, 0.9507$	
[50]	ELM	R = 0.9851		
[53]	BPNN	Relative Absolute Error = 6.63		
[54]	ANN	MAE = 4.179		
[56]	Hybrid ensemble model	RMSE MAE 0.022, 0.0104, 0.049 0.0701	R^2 0.989, 0.889	MAPE 3.818, 16.276
			a-20 97.642, 78.641	VAF 98.877, 88.834
[57]	ANN	$R^2 = 0.9865$		RSR 0.106, 0.335
[58]	BPNN	$R^2 = 0.903, 0.829$	WMAPE 0.025, 0.119	
[59]	ANN	$R^2 = 0.98$		
[60]	ANN	$R^2 = 0.99$		
[61]	Feed-forward backpropagation NN	$R^2 = 0.9971$	MSE = 0.004447	
[62]	GA-ANN	MAPE = 2.24%		
[63]	ANN	RMSE = 3.6	$R^2 = 0.893$	MAPE = 10.036

Table 6 (continued)

Reference	Algorithm	Evaluation and performance (training, testing)
[64]	ANN Second-order ANN	RMSE = 7.17 5.82 MAE = 5.59 4.5 R = 0.9 0.93

as 8, and it utilizes the sigmoid activation function. In contrast, the output layer utilizes the linear activation function. Ensemble models were also created for generalization purposes, with 100 base models with varying nodes up to 16 numbers in the hidden layer.

The ANN model used to predict the compressive strength of Rice husk ash concrete has one input layer with ten neurons, two hidden layers with a maximum of up to ten neurons in each layer, and one output layer with one neuron. ANN performed the best of all the methods used in this study, with an R^2 value of 0.98 [28].

Waris et al. [57] predicted the compressive strength of concrete with different cement replacement material (CRM) ratios using image processing and ANN. 18 concrete cylinders were cast for this purpose with different CRM ratios ranging from 0 to 25%. 9 cylinders were tested for their 14-day compressive strength under CTM. The remaining nine cylinders were cut into three slices; a DSLR took images of the surface of the slice. The captured images were converted to grayscale and resized to 256x256 resolution. Statistical features of mean, median and standard deviation were extracted from the pixels of the images; these features were taken as input, and the compressive strength, tensile strength and slump were taken as output. The ANN architecture has ten hidden neurons and two hidden layers. ANN performed well with an R^2 value of 0.9865.

Asteris and Mokos [9] predicted the compressive strength of concrete using a backpropagation neural network. The study aims at correlating the experimental readings from the ultrasonic pulse velocity (UPV) test and Schmidt’s rebound hammer test to the compressive strength of concrete. Although these tests can directly indicate concrete’s durability or compressive strength, they often exhibit large dispersion in their results and deviation from the actual value of compressive strength. Hence, the BPNN was applied with input as UPV and Schmidt’s rebound hammer value to overcome these limitations. Three BPNN architectures were used. The first architecture (2-25-1) had two neurons in the input layer, 25 in the hidden layer, and one in the output layer. The second architecture (1-26-1) had one neuron in the input layer, 26 neurons in the hidden and one in the output layer. The third architecture (1-28-1) had one neuron in the input layer, 28 in the hidden layer, and 1 in the output layer. A hyperbolic sigmoid transfer function was used as an activation function at the hidden and output layer. The optimum architecture was found to be 2-25-1 when both inputs were used, when only UPV was used as an input (1-28-1) performed better and when only a rebound hammer was used as an input (1-26-1).

Naderpour et al. [58] used a backpropagation neural network to predict the compressive strength of environmentally friendly concrete. The dataset of recycled aggregate concrete

from various literature was used in this study. A neural network is built with six inputs, one hidden layer, one output, a backpropagation training algorithm and a log-sigmoid activation function. The optimum number of hidden nodes in the hidden layer was 18. The model's performance was satisfactory and had a regression of 0.829 in testing.

Dogan et al. [59] predicted the compressive strength of concrete using image processing trained by ANN. One hundred forty-four concrete cylinders of 150mm diameter and 300mm length were prepared in the laboratory. A 50mm portion was cut from the top, and then the surface of 144 concrete samples was photographed using a digital camera in a controlled environment; after this, a compression test was performed on the samples to get the compressive strength as output data. The image was converted to greyscale with a pixel size of 3888 x 2592; thus, a matrix of grayscale values of size 3888 x 2592 was created. This was further simplified by taking mean, median, and standard deviation along the matrix rows and forming a final matrix of 3 x 2592. This matrix was used as an input layer in ANN. The hidden layer consisted of 16 nodes, and the ANN architecture was trained using Levenberg-Marquardt's training algorithm. The model performed exceptionally well, with an R^2 value of 0.98.

Li et al. [54] used image processing with neural networks to predict the compressive strength of the concrete. For this purpose, microstructure images of concrete were taken using micro-computed tomography. The grey-level histogram showed the quantities of different substances. The grey values of pixels are formed as a matrix, namely the grey level matrix. Along with this, a co-occurrence matrix is also formed, showing the surface's texture. A neural network is trained with input as eigenvalues of grey level and co-occurrence matrix, a hidden layer with ten hidden neurons and the output layer with a single neuron of compressive strength. Backpropagation is used as a training algorithm. This method was also compared with traditional ML techniques like MLR and GPR. The performance evaluation showed that ANN had the least MAE of 4.179, followed by 4.46 and 4.95 for GPR and MLR, respectively.

Dogan et al. [60] estimated the compressive strength of concrete using statistical features of concrete images. For this, 60 cubes of 150mm were cast in a laboratory-controlled environment, and the surface images were taken on 7 and 28 days along with NDT tests like ultrasonic pulse velocity. The data of the image was converted into the matrix, with each value of the matrix corresponding to the pixel value of the image. Further, statistical data like mean, standard deviation, the median was extracted from the matrix, which was later used as input in ANN; a Levenberg-Marquardt backpropagation algorithm trained the ANN. Four models of ANN were used for statistical data of mean, standard deviation, and median and when all used together, and hidden

layer contained 12, 8, 26 and 14 neurons, respectively. ANN performed exceptionally well, with an R^2 value of 0.99.

Onal and Ozturk [61] predicted the compressive strength of cement mortar by establishing a relation between the microstructure and compressive strength using feed-forward backpropagation neural network (ANN). For this purpose, 130 samples of 50mm cement mortar cubes were prepared from four different chemical admixture doses and were tested on 1, 2, 7, 28 and 90 days for compressive strength. Before conducting compressive strength tests, microstructural studies were conducted on the polished surface of cubes. Features like the dendrite length, pore length, pore area ratio, average roundness and area of hydrated and un-hydrated parts were used as input in ANN. ANN was performed with an R^2 value of 0.9971.

An ANN model was developed by Özcan [32] to predict the compressive strength of silica fume concrete. ANN architecture was built by six input features in the input layer, an optimum number of hidden neurons in the hidden layer was found by trial and error as 11, and the output layer consisted single output of compressive strength. The ANN was trained using the Levenberg-Marquardt training algorithm, and logarithmic sigmoid and pure linear transfer functions were used as activation functions. The ANN performed better with an R^2 of 0.977 than its counterpart fuzzy logic method in this study.

The ANN model studied by Sardemir et al. [31] for predicting the compressive strength of concrete consisted of 5 input features specified in Table 5, 6 hidden neurons in the first hidden layer, five hidden neurons in the second hidden layer and compressive strength as output neuron. The backpropagation training algorithm and sigmoid activation function were used. Compared to Fuzzy logic, which was also undertaken in this study, ANN performed better with an R^2 value of 0.981 in testing.

Topcu and Sardemir [33] compared ANN with fuzzy logic to predict the compressive strength of concrete containing fly ash. The ANN architecture had nine input features in the input layer, 11 hidden neurons in the hidden layer, and compressive strength as output in the output layer. Backpropagation and sigmoid function were used as training algorithms and activation functions, respectively. ANN performed marginally less than Fuzzy logic in this study, with an R^2 value of 0.997 vs. 0.998 for Fuzzy logic.

Baykasoglu et al. [10] defined a neural network (NN) model to predict the compressive strength of cement. Up to 65 NN models were trained with different numbers of hidden layers (1 to 3) and different numbers of hidden neurons (7, 10, 13); the models were trained using the Delta-Bar-Delta algorithm. The best NN model secured an R^2 value of 0.695.

Akkurt et al. [62] predicted the compressive strength of cement mortar using an ensemble of genetic algorithms and artificial neural networks called GA-ANN. A total of 150

Table 7 Summary of Deep Machine Learning algorithms w.r.t features and data points

References	Algorithm	Data source	No. of data	Used features
[22]	DCNN (ConcNet_A, ConcNet_G, ConNet_R)	Surface images and images from the video	4000 images + 61,996 video images	Surface images and images from the video
[53]	CNN	Lab Tests	74	Recycled fine aggregate replacement ratio, water-cement ratio, fly ash replacement ratio, recycled coarse aggregate replacement ratio
[63]	DNN, Deep residual network	Literature Data	263	Curing time, NaOH/Na ₂ SiO ₃ , AL/Fly ash, fly ash/Aggregate, Curing temperature, NaOH concentration
[64]	Higher-order neuron	Literature Data	177	Sand-cement ratio, density, water-cement ratio
[65]	DNN	Literature Data	233	Fine Aggregate, Cement, Coarse Aggregate, Binder, Water/Cement, compressive strength specimen type, Water, Crumb rubber, Cement replacement, Superplasticizer, Water/Binder, Chipped rubber
[66]	DBN	3D Microstructure Images	27,000 images	3D Microstructure Images
[67]	BPNN (w/o SFS & NID); BPNN (with SFS & NID)	Literature Data	110	Admixture, quartz powder, cement, sand, steel fibre, silica fume, fly ash, water
[68]	DCNN (AlexNet, GoogleNet, ResNet)	Surface Images	5145 images	Surface microscopic images
[72]	CNN	Microscopic images	1080 images	Microscopic images
[73]	Simple averaging ensemble, weighted averaging ensemble, super learner, Integrated stacking, separate stacking	University of California, Irvine	270	Fly ash, cement, superplasticizer, coarse aggregate, water-cement ratio, days, water, fine aggregate

data sets were collected from the cement plant for this study. This data set was divided into training and testing sets using GA. The training and testing data sets have almost the same range of data, and the average of both is similar to the average of the entire data set. This proper partitioning of the data ensures optimal learning from ANN. A sensitivity analysis was also performed using different input parameters, and it was found that an increase in tri-calcium silicate, sulfur trioxide and surface area increased the compressive strength of cement mortar.

Guang and Zong [20] predicted the compressive strength of concrete using multi-layer feed-forward neural networks (MFNN). In this study, 11 input features are used, as described in the table below. The neural network consists of 11 input neurons, seven hidden neurons in the hidden layer and one output neuron. Backpropagation was used as a training algorithm. In the first phase of NN training, 65 concrete cubes of 150mm were cast in the laboratory, out of which 50 were used for learning and 15 for testing. In the second phase of NN training, 100 data sets were collected from a cement plant, of which 85 were used for learning and

15 for testing. This method proved to be reasonably accurate in estimating the strength of concrete.

Deep machine learning

This section discusses deep machine learning models like a deep neural network, convolutional neural network, deep residual network, deep belief network, and similar algorithms. The description of the methods, the features used, the number of data samples and the data source is summarised in Table 7. The summary of performance evaluation is summarised in Table 8.

Ly et al. [65] predicted the compressive strength of rubber concrete using a deep neural network (DNN). 233 datasets from different studies were used for training the deep neural network; this was named dataset 1 and contained 12 inputs, as mentioned in Table 7. Along with this, two supplementary datasets were also gathered from the literature, namely dataset 2 (187 samples and four input parameters which included compressive strength of controlling concrete as a new variable) and dataset 3 (183 samples and seven input

Table 8 Summary of Deep Machine Learning algorithms with respect to the evaluation metrics

Reference	Algorithm	Evaluation and Performance (training, testing)		
[22]	ConcNet_A	RMSE = 3.82		
	ConcNet_G	3.64		
	ConcNet_R	3.56		
[53]	CNN	Relative Absolute Error = 3.65		
[63]	DNN	RMSE = 3.391	$R^2 = 0.912$	MAPE = 9.487
	Deep Residual Network	2.75	0.934	8.552
[64]	Higher-order neuron	RMSE = 4.05	MAE = 2.85	R = 0.97
[65]	DNN	RMSE = 3.4, 3.01	MAE = 2.30, 2.32	$R^2 = 0.97, 0.98$
[66]	DBN	MAE = 2.81		
[67]	BPNN (w/o SFS & NID)	$R^2 = 0.215$		
	BPNN (with SFS & NID)	$R^2 = 0.801$		
[68]	DCNN (AlexNet)	RMSE = 4.641	$R^2 = 0.745$	MAPE = 17.675
	DCNN (GoogleNet)	4.612	0.748	18.403
	DCNN (ResNet)	4.463	0.764	17.765
[72]	CNN	MAE = 3.606		
[73]	Simple averaging ensemble	$R^2 = 0.966, 0.965$	MSE = 0.005, 0.006	A20 = 0.593
	Weighted averaging ensemble	0.976, 0.973	0.004, 0.005	0.64
	Super learner	0.995, 0.88	0.028, 0.131	0.1
	Integrated stacking	0.967, 0.964	0.005, 0.006	0.648
	Separate stacking (SVM regressor)	0.973, 0.96	0.0043, 0.0068	0.593
	Separate stacking (AdaBoost reg.)	0.987, 0.963	0.0021, 0.006	0.537
	Separate stacking (RF regressor)	0.995, 0.976	0.0009, 0.0041	0.7
	Separate stacking (Bagging reg.)	0.994, 0.971	0.001, 0.0049	0.611
	Separate stacking (Gradient boosting)	0.997, 0.972	0.0005, 0.0045	0.593

parameters which included density of concrete as new variable). The DNN architecture had three hidden layers with neurons ranging from 1 to 20. The DNN was trained using Broyden–Fletcher–Goldfarb–Shanno Quasi-Newton backpropagation algorithm. DNN structure having 12 input nodes, 16 nodes in 1st hidden layer, 14 nodes in 2nd hidden layer, three nodes in 3rd hidden layer and one output node performed the best and showed a regression coefficient of $R = 0.98$.

Guo et al. [66] developed 3-dimensional microstructure images of concrete to estimate its compressive strength using a deep belief network (DBN). The 3-D microstructure of cement was obtained by micro-computed tomography. From the 3-D images, the grey level histogram, which suggests the hydrating components of cement and the grey level co-occurrence matrix, represents the spatial relationship in the hydrated cement. Statistical features like mean, energy, entropy, kurtosis, variance, and skewness are extracted from the grey-level histogram matrix. Statistical features like IDM (Inverse Different Moment), energy, entropy and correlation are extracted from the grey-level co-occurrence matrix. The extracted features form the input of the deep belief network. The training process of the deep belief network was parallelized on CUDA (compute unified device architecture). DBN performed well with a MAE of 2.81 MPa.

Abuodeh et al. [67] used deep machine learning techniques to predict the compressive strength of ultra-high-performance concrete (UHPC). A backpropagation neural network was implemented with sequential feature selection (SFS), and a neural interpretation diagram (NID) was used to verify selected parameters by SFS. Eight constituents were analysed from 110 UHPC compression data by executing SFS and NID, and the most influential constituents which improved backpropagation neural networks were selected. Abram's compressive strength model was later modified using the influential features selected. The eight features used for analysis were cement, silica fume, fly ash, sand, steel fibre, quartz powder, water and admixture. Four influential features were cement, fly ash, silica fume and water. BPNN with SFS and NID performed significantly well with an R^2 value of 0.801 compared to R^2 of 0.215 before SFS and NID.

Huynh et al. [63] applied deep and shallow machine learning to predict the compressive strength of geopolymer concrete blended with fly ash. This research was carried out on the premise that conventional compressive strength prediction of geopolymer concrete requires a large volume of raw material, expensive equipment and extensive time. Deep neural networks (DNN), deep residual networks and

ANN were employed for this purpose. DNN architecture comprises seven neurons in the input layer, 128 neurons in the first hidden layer, and 256 in the second. The deep residual network also consisted of 7, 128 and 256 neurons in the input layer, the first hidden layer and the second hidden layer, respectively; however, the deep residual network also had a third hidden layer with 256 neurons for the addition operation. The ANN architecture consists of 7 input neurons and 384 hidden neurons in the first hidden layer. The Adam gradient descent method was used as a training algorithm to update the weights, and ReLU was used as an activation function. It was seen that the deep residual network performed best with an R^2 value of 0.934, followed by DNN and ANN with R^2 values of 0.912 and 0.893, respectively.

Jang et al. [68] estimated the compressive strength of the concrete using image processing based on deep convolutional neural networks (DCNN). For this purpose, three DCNN algorithms (AlexNet, GoogleNet and ResNet) were proposed. Three concrete cylinders of 100mm diameter were made with OPC for every three water-cement ratios (0.33, 0.5, 0.68) for 3, 7 and 28 days, thus making 27 samples. A portable digital microscope was used to get microscopic images of the top and bottom surfaces of the cylinder. Up to 200 images of a single surface at different sections and illumination levels were taken. Data augmentation, like rotation and flipping, was also applied. The final output layer of AlexNet, GoogleNet and ResNet was modified to Euclidean loss function. The weights were updated using backpropagation, and the activation function used was ReLU. ANN also analysed the image processing, and the performance was compared. It was inferred that DCNN with ResNet with an R^2 of 0.764 performed better than the other two DCNN models, with GoogleNet having an R^2 value of 0.748 and AlexNet with 0.745, ANN performed poorly with a 0.2 value.

Shin et al. [22] estimated the compressive strength of concrete using the digital vision-based method. For this purpose, a Deep Convolutional Neural Network using three modified algorithms, namely ConcNet_A, ConcNet_G and ConcNet_R, was used. ConcNet_A model is an improved version of AlexNet in which the learning rate is improved using ReLU. ConcNet_G is an improvised version of GoogleNet and uses 22 layers to overcome overfitting problems. ConcNet_R is an improvised version of ResNet, and in this, instead of 152 layers as in ResNet, 50 layers are proposed since the image size for training is considerably smaller than the actual size. For this study, an image size of 84x84 was chosen, randomly cropped from a 112x112 resolution image. Data augmentation, like flip, was also applied. The ConcNet_R performed better prediction with an RMSE of 3.56.

A digital image correlation (DIC) technique was used by Afrazi et al. [70] to identify the surface displacement

field and crack initiation along with its propagation in quasi-brittle materials, and a numerical finite element program was also developed to predict the same. The results were consistent with physical measurements. Majedi et al. [71] from their study developed a micromechanical model to overcome the contact problem in finite element modelling. This model improved the results and was comparable with the experimental results.

Deng et al. [53] used CNN to predict the compressive strength of recycled concrete. The concrete samples were prepared with recycled fine as well as coarse aggregates. Mix designs with 16 different water-cement ratios, 16 different fly ash replacement ratios, 21 different coarse aggregate replacement ratios and 21 different fine aggregate replacement ratios were prepared, thus totalling 74 samples. The performance of CNN was then compared with BPNN and SVM. Four inputs were considered, and the architecture of BPNN considered nine hidden nodes in the hidden layer. CNN was trained till 33 epochs. The prediction capability of CNN was comparatively better than BPNN and SVM.

Nguyen et al. [64] used a deep neural network to predict the compressive strength of foamed concrete. The performance was compared with conventional ANN and second-order ANN used by Fan et al. [66]. A higher-order DNN was developed based on the second-order neuron architecture; the higher-order neuron comprises three activation functions, and a second-order neuron could be considered a particular case. Instead of the sigmoid function as used in second-order neurons, here ReLU activation function was used. The evaluation of the performance of higher-order DNN was the best, with a regression coefficient of 0.97 followed by 0.93 and 0.9 for second-order neurons and conventional ANN.

Li et al. [72] estimated the compressive strength of cement using CNN and image processing. For this purpose, microstructure images of concrete were taken using micro-computed tomography. The different grey levels in the image correlate to different substances after hydration. Microstructure images are taken as input layers for CNN, and the convolution layers perform the feature extraction and mapping. The convolution layers contain convolution kernels that abstract features from the concrete images. The CNN architecture in this study consisted of 1 input layer, two convolution layers, two sub-sampling layers, one full connection layer and one output layer. This method performed exceptionally well with a MAE of 3.606 MPa.

Barkhordari et al. [73] predicted the compressive strength of fly ash concrete using several deep neural network model ensembles like a simple averaging ensemble, weighted averaging ensemble, super learner, integrated stacking ensemble, and separate stacking ensemble using various regressors. In this study, 6 DNNs are used for

basic learning. The simple averaging ensemble is the average of six DNNs; the weighted averaging ensemble is the weighted average of the six DNNs; stacking ensemble merges the sub-models into a single model; this was performed using various regressors like SVM regressor, AdaBoost regressor, RF, Bagging and Gradient boosting regressor. The integrated stacking ensemble used a neural network in which 6 DNN sub-models were used in the input layer. By trial and error, the hidden neurons were decided in the single hidden layer.

Conclusion and future research

Challenges and concluding remarks

Compressive strength is one of concrete's most important mechanical properties, and determining the same requires expensive equipment and is time-consuming. AI-based techniques showed promising results in predicting the compressive strength of different types of concrete with various cementitious materials and cement mortars. These methods can significantly reduce the cost, labour, time and material involved in the determination. For this purpose, this study reviewed literature from various databases and summarised their findings. The techniques used, dataset, number of data points, evaluation parameters, and performances were also summarised. Following conclusions and challenges were drawn from the studies.

Generalisation

Several types of concretes, like concrete blended with cementitious and waste materials (fly ash, GGBS, silica fumes, rubber, rice husk ash), were modelled using soft computing techniques; however, a unified approach to generalize all concrete types is not perused.

Sensitivity

Soft computing techniques are sensitive to the parameters chosen to build the model and the parameters used to build the dataset. Generating synthetic data or collecting extensive data can help in these scenarios.

Precision

A good prediction for the compressive strength of concrete or cement should have the least deviation from the actual value for different hydration chronologies on various days.

Soft computing techniques have been shown to have some errors of different magnitudes at various hydration levels; this needs to be minimized.

Large scale application

Although a lot of research is being carried out in this domain, its practicality and applicability to actual structures on a large scale are minimally carried out. Soft computing techniques like digital vision and image processing can help capture more extensive data and produce quicker results. The chemical formulation of cement and other materials can also be utilized using soft computing techniques to predict compressive strength.

Future research

After analysing the research to date, as per the authors' best knowledge, it was established that the research in the prediction of compressive strength using soft computing techniques is majorly focused on developing models and predicting the compressive strength for hydrated or partially hydrated concrete. Efforts to predict the compressive strength of hardened concrete while it is in its fresh or un-hydrated state were still lacking. Thus, the authors of this paper propose a novel research direction of predicting the compressive strength of hardened concrete while the concrete is still in its fresh state or workable.

Funding No external funding has been received for this review work.

Declarations

Conflict of 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.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent For this type of study a formal consent is not required.

References

1. Bureau of Indian Standards (2005) IS 456: 2000-Plain and reinforced concrete-code of Practice
2. European Committee for Standardization (2002) EN 1990: Eurocode-basis of structural design
3. British Standards Institution (1997) Structural use of concrete. BSI

4. ACI Committee 318 and American Concrete Institute (2011) Building code requirements for structural concrete (ACI 318–11) and commentary. American Concrete Institute
5. Bureau of Indian Standards, IS 1199 (Part 1) (2018) Fresh Concrete - Methods of Sampling, Testing and Analysis, vol 1
6. Bureau of Indian Standards, IS 516 (2021) - Hardened concrete - Methods of test. 2021. [Online]. Available: www.standardsbis.in
7. European Committee for Standardization, BS EN 12390–3 (2019)
8. Ghahri Saremi S, Goulias D (2020) Concrete strength gain monitoring with non-destructive methods for potential adoption in quality assurance. *Constr Build Mater* 260:120464. <https://doi.org/10.1016/J.CONBUILDMAT.2020.120464>
9. Asteris PG, Mocos VG (2020) Concrete compressive strength using artificial neural networks. *Neural Comput Appl* 32(15):11807–11826. <https://doi.org/10.1007/s00521-019-04663-2>
10. Baykasoğlu A, Dereli T, Tanış S (2004) Prediction of cement strength using soft computing techniques. *Cem Concr Res* 34(11):2083–2090. <https://doi.org/10.1016/j.cemconres.2004.03.028>
11. Mahdinia S, Eskandari-Naddaf H, Shadnia R (2019) Effect of cement strength class on the prediction of compressive strength of cement mortar using GEP method. *Constr Build Mater* 198:27–41. <https://doi.org/10.1016/J.CONBUILDMAT.2018.11.265>
12. Sharma R (2021) Effect of wastes and admixtures on compressive strength of concrete. *J Eng Des Technol* 19(1):219–244. <https://doi.org/10.1108/JEDT-01-2020-0031/FULL/XML>
13. del Viso JR, Carmona JR, Ruiz G (2008) Shape and size effects on the compressive strength of high-strength concrete. *Cem Concr Res* 38(3):386–395. <https://doi.org/10.1016/J.CEMCONRES.2007.09.020>
14. Ortiz J, Aguado A, Agulló L, García T (2005) Influence of environmental temperatures on the concrete compressive strength: Simulation of hot and cold weather conditions. *Cem Concr Res* 35(10):1970–1979. <https://doi.org/10.1016/J.CEMCONRES.2005.01.004>
15. Park DC, El-Sharkawi MA, Marks RJ, Atlas LE, Damborg MJ (1991) Electric load forecasting using an artificial neural network. *IEEE Trans Power Syst* 6(2):442–449. <https://doi.org/10.1109/59.76685>
16. Himmelblau DM (2000) Applications of artificial neural networks in chemical engineering. *Korean J Chem Eng* 17(4):373–392. <https://doi.org/10.1007/BF02706848>
17. Rafiee J, Arvani F, Harifi A, Sadeghi MH (2007) Intelligent condition monitoring of a gearbox using artificial neural network. *Mech Syst Signal Process* 21(4):1746–1754. <https://doi.org/10.1016/J.YMSP.2006.08.005>
18. Mandal S, Subba Rao N, Harish L (2012) Damage level prediction of non-reshaped berm breakwater using ANN, SVM and ANFIS models. *Int J Nav Archit Ocean Eng* 4(2):112–122. <https://doi.org/10.2478/IJNAOE-2013-0082>
19. Yilmaz I, Kaynar O (2011) Multiple regression, ANN (RBF, MLP) and ANFIS models for prediction of swell potential of clayey soils. *Expert Syst Appl* 38(5):5958–5966. <https://doi.org/10.1016/J.ESWA.2010.11.027>
20. Ni HG, Wang JZ (2000) Prediction of compressive strength of concrete by neural networks. *Cem Concr Res* 30(8):1245–1250
21. Kang MC, Yoo DY, Gupta R (2021) Machine learning-based prediction for compressive and flexural strengths of steel fiber-reinforced concrete. *Constr Build Mater*. <https://doi.org/10.1016/j.conbuildmat.2020.121117>
22. Shin HK, Ahn YH, Lee SH, Kim HY (2019) Digital vision based concrete compressive strength evaluating model using deep convolutional neural network. *Comput Mater Continua* 61(2):911–928. <https://doi.org/10.32604/cmc.2019.08269>
23. Khormani M, Kalat Jaari VR, Aghayan I, Ghaderi SH, Ahmadyfard A (2020) Compressive strength determination of concrete specimens using X-ray computed tomography and finite element method. *Constr Build Mater*. <https://doi.org/10.1016/j.conbuildmat.2020.119427>
24. Boole G (1854) An investigation of the laws of thought on which are founded the mathematical theories of logic and probabilities. Walton and Maberly, London. <https://doi.org/10.5962/bhl.title.29413>
25. Rencher AC (2002) Methods of multivariate analysis. Wiley, Hoboken
26. Salzberg SL (1994) C4.5: programs for machine learning by J. Ross Quinlan. Morgan Kaufmann publishers, Inc., 1993. *Mach Learn* 16(3):235–240. <https://doi.org/10.1007/BF00993309>
27. Leung FHF, Lam HK, Ling SH, Tam PKS (2003) Tuning of the structure and parameters of a neural network using an improved genetic algorithm. *IEEE Trans Neural Networks* 14(1):79–88. <https://doi.org/10.1109/TNN.2002.804317>
28. Iqtidar A et al (2021) Prediction of compressive strength of rice husk ash concrete through different machine learning processes. *Crystals*. <https://doi.org/10.3390/cryst11040352>
29. Tayfur G, Erdem TK, Kirca Ö (2014) Strength prediction of high-strength concrete by fuzzy logic and artificial neural networks. *J Mater Civ Eng* 26(11):04014079. [https://doi.org/10.1061/\(asce\)mt.1943-5533.0000985](https://doi.org/10.1061/(asce)mt.1943-5533.0000985)
30. Abolpour B, Abolpour R, Bakhshi H (2013) Estimation of concrete compressive strength by a fuzzy logic model. *Res Chem Intermed* 39(2):707–719. <https://doi.org/10.1007/s11164-012-0591-0>
31. Saridemir M, Topçu IB, Özcan F, Severcan MH (2009) Prediction of long-term effects of GGBFS on compressive strength of concrete by artificial neural networks and fuzzy logic. *Constr Build Mater* 23(3):1279–1286. <https://doi.org/10.1016/j.conbuildmat.2008.07.021>
32. Özcan F, Atiş CD, Karahan O, Uncuoğlu E, Tanyildizi H (2009) Comparison of artificial neural network and fuzzy logic models for prediction of long-term compressive strength of silica fume concrete. *Adv Eng Software* 40(9):856–863. <https://doi.org/10.1016/j.advengsoft.2009.01.005>
33. Topçu IB, Saridemir M (2008) Prediction of compressive strength of concrete containing fly ash using artificial neural networks and fuzzy logic. *Comput Mater Sci* 41(3):305–311. <https://doi.org/10.1016/j.commatsci.2007.04.009>
34. Akkurt S, Tayfur G, Can S (2004) Fuzzy logic model for the prediction of cement compressive strength. *Cem Concr Res* 34(8):1429–1433. <https://doi.org/10.1016/j.cemconres.2004.01.020>
35. Güçlüter K, Özbeyaz A, Göymen S, Günaydn O (2021) A comparative investigation using machine learning methods for concrete compressive strength estimation. *Mater Today Commun*. <https://doi.org/10.1016/j.mtcomm.2021.102278>
36. Salami BA, Olayiwola T, Oyehan TA, Raji IA (2021) Data-driven model for ternary-blend concrete compressive strength prediction using machine learning approach. *Constr Build Mater*. <https://doi.org/10.1016/j.conbuildmat.2021.124152>
37. Ahmad W, Ahmad A, Ostrowski KA, Aslam F, Joyklad P, Zajdel P (2021) Application of advanced machine learning approaches to predict the compressive strength of concrete containing supplementary cementitious materials. *Materials*. <https://doi.org/10.3390/ma14195762>
38. Ahmad A, Ostrowski KA, Maşlak M, Farooq F, Mehmood I, Nafees A (2021) Comparative study of supervised machine learning algorithms for predicting the compressive strength of concrete at high temperature. *Materials*. <https://doi.org/10.3390/ma14154222>

39. Kovačević M, Lozančić S, Nyarko EK, Hadzima-nyarko M (2021) Modeling of compressive strength of self-compacting rubberized concrete using machine learning. *Materials*. <https://doi.org/10.3390/ma14154346>
40. Khursheed S, Jagan J, Samui P, Kumar S (2021) Compressive strength prediction of fly ash concrete by using machine learning techniques. *Innov Infrastruct Solut*. <https://doi.org/10.1007/s41062-021-00506-z>
41. Salimbahrami SR, Shakeri R (2021) Experimental investigation and comparative machine-learning prediction of compressive strength of recycled aggregate concrete. *Soft Comput* 25(2):919–932. <https://doi.org/10.1007/s00500-021-05571-1>
42. Rizvon SS, Jayakumar K (2021) Machine learning techniques for recycled aggregate concrete strength prediction and its characteristics between the hardened features of concrete. *Arabian J Geosci*. <https://doi.org/10.1007/s12517-021-08674-z>
43. Asteris PG, Koopialipoor M, Armaghani DJ, Kotsonis EA, Lourenço PB (2021) Prediction of cement-based mortars compressive strength using machine learning techniques. *Neural Comput Appl* 33(19):13089–13121. <https://doi.org/10.1007/s00521-021-06004-8>
44. Feng DC et al (2020) Machine learning-based compressive strength prediction for concrete: an adaptive boosting approach. *Constr Build Mater*. <https://doi.org/10.1016/j.conbuildmat.2019.117000>
45. Kumar M, Sihag P, Singh V (2019) Enhanced soft computing for ensemble approach to estimate the compressive strength of high strength concrete. *J Mater Eng Struct* 6(1):93–103
46. Hadzima-Nyarko M, Nyarko EK, Lu H, Zhu S (2020) Machine learning approaches for estimation of compressive strength of concrete. *Eur Phys J Plus*. <https://doi.org/10.1140/epjp/s13360-020-00703-2>
47. Dutta S, Samui P, Kim D (2018) Comparison of machine learning techniques to predict compressive strength of concrete. *Comput Concr* 21(4):463–470. <https://doi.org/10.12989/cac.2018.21.4.463>
48. Chopra P, Sharma RK, Kumar M, Chopra T (2018) Comparison of machine learning techniques for the prediction of compressive strength of concrete. *Adv Civ Eng*. <https://doi.org/10.1155/2018/54817055>
49. Mirzahosseini M, Jiao P, Barri K, Riding KA, Alavi AH (2019) New machine learning prediction models for compressive strength of concrete modified with glass cullet. *Eng Comput* 36(3):876–898. <https://doi.org/10.1108/EC-08-2018-0348>
50. Yaseen ZM et al (2018) Predicting compressive strength of lightweight foamed concrete using extreme learning machine model. *Adv Eng Software* 115:112–125. <https://doi.org/10.1016/j.advengsoft.2017.09.004>
51. Başığit C, Çomak B, Kiliçarslan Ş, Serkan Üncü I (2012) Assessment of concrete compressive strength by image processing technique. *Constr Build Mater* 37:526–532. <https://doi.org/10.1016/j.conbuildmat.2012.07.055>
52. Zhang Q, Yang B, Wang L, and Zhu F (2012) Predicting cement compressive strength using double-layer multi-expression programming. In: *Proceedings - 4th international conference on computational and information sciences*. DOI: <https://doi.org/10.1109/ICCIS.2012.207>.
53. Deng F, He Y, Zhou S, Yu Y, Cheng H, Wu X (2018) Compressive strength prediction of recycled concrete based on deep learning. *Constr Build Mater* 175:562–569. <https://doi.org/10.1016/j.conbuildmat.2018.04.169>
54. Li M, Yang B, Wang L, Liu Y, Zhao X, Zhou J and Zhang L (2016) The prediction of cement compressive strength based on gray level images and neural network. In: *2016 3rd international conference on informative and cybernetics for computational social systems* 103–108.
55. Ustebay S, Turgut Z, Aydin MA (2019) Cyber attack detection by using neural network approaches: shallow neural network, deep neural network and autoencoder. *Commun Comput Inf Sci* 1039:144–155. https://doi.org/10.1007/978-3-030-21952-9_11
56. Asteris PG, Skentou AD, Bardhan A, Samui P, Pilakoutas K (2021) Predicting concrete compressive strength using hybrid ensembling of surrogate machine learning models. *Cem Concr Res*. <https://doi.org/10.1016/j.cemconres.2021.106449>
57. Waris MI, Mir J, Plevris V, and Ahmad A (2020) Predicting compressive strength of CRM samples using Image processing and ANN. In: *IOP Conference series: materials science and engineering*. DOI: <https://doi.org/10.1088/1757-899X/899/1/012014>.
58. Naderpour H, Rafiean AH, Fakharian P (2018) Compressive strength prediction of environmentally friendly concrete using artificial neural networks. *J Building Eng* 16:213–219. <https://doi.org/10.1016/j.jobbe.2018.01.007>
59. Dogan G, Arslan MH, Ceylan M (2017) Concrete compressive strength detection using image processing based new test method. *Meas J Int Meas Confed* 109:137–148. <https://doi.org/10.1016/j.measurement.2017.05.051>
60. Doğan G, Arslan MH, Ceylan M (2015) Statistical feature extraction based on an ann approach for estimating the compressive strength of concrete. *Neural Network World* 25(3):301–318. <https://doi.org/10.14311/NNW.2015.25.016>
61. Onal O, Ozturk AU (2010) Artificial neural network application on microstructure-compressive strength relationship of cement mortar. *Adv Eng Softw* 41(2):165–169. <https://doi.org/10.1016/j.advengsoft.2009.09.004>
62. Akkurt S, Ozdemir S, Tayfur G, Akyol B (2003) The use of GA-ANNs in the modelling of compressive strength of cement mortar. *Cem Concr Res* 33(7):973–979. [https://doi.org/10.1016/S0008-8846\(03\)00006-1](https://doi.org/10.1016/S0008-8846(03)00006-1)
63. Huynh AT et al (2020) A machine learning-assisted numerical predictor for compressive strength of geopolymer concrete based on experimental data and sensitivity analysis. *Appl Sci* 10(21):1–16. <https://doi.org/10.3390/app10217726>
64. Nguyen T, Kashani A, Ngo T, Bordas S (2019) Deep neural network with high-order neuron for the prediction of foamed concrete strength. *Comput Aided Civ Infrastruct Eng* 34(4):316–332. <https://doi.org/10.1111/mice.12422>
65. Ly HB, Nguyen TA, Tran VQ (2021) Development of deep neural network model to predict the compressive strength of rubber concrete. *Constr Build Mater* 301:124081. <https://doi.org/10.1016/j.conbuildmat.2021.124081>
66. Guo J et al (2020) Estimating cement compressive strength using three-dimensional microstructure images and deep belief network. *Eng Appl Artif Intell*. <https://doi.org/10.1016/j.engappai.2019.103378>
67. Abuodeh OR, Abdalla JA, Hawileh RA (2020) Assessment of compressive strength of Ultra-high Performance Concrete using deep machine learning techniques. *Appl Soft Comput J*. <https://doi.org/10.1016/j.asoc.2020.106552>
68. Jang Y, Ahn Y, Kim HY (2019) Estimating Compressive Strength of Concrete Using Deep Convolutional Neural Networks with Digital Microscope Images. *J Comput Civ Eng* 33(3):04019018. [https://doi.org/10.1061/\(asce\)cp.1943-5487.0000837](https://doi.org/10.1061/(asce)cp.1943-5487.0000837)
69. Fan F, Cong W, Wang G (2018) A new type of neurons for machine learning. *Int J Numer Methods Biomed Eng* 34(2):e2920. <https://doi.org/10.1002/CNM.2920>
70. Afrazi M, Lin Q, Fakhimi A (2022) Physical and numerical evaluation of mode II Fracture of Quasi-Brittle materials. *Int J Civ Eng*. <https://doi.org/10.1007/s40999-022-00718-z>

71. Majedi MR, Afrazi M, Fakhimi A (2020) A Micromechanical Model for Simulation of Rock Failure Under High Strain Rate Loading. *International Journal of Civil Engineering*. <https://doi.org/10.1007/s40999-020-00551-2>
72. Li M, Wang L, Yang B, Zhang L, and Liu Y (2017) Estimating cement compressive strength from microstructure images using convolutional neural network. In: *IEEE Symposium series on computational intelligence*, pp 1–7.
73. Barkhordari MS, Armaghani DJ, Mohammed AS, Ulrikh DV (2022) Data-driven compressive strength prediction of fly ash concrete using ensemble learner algorithms. *Buildings* 12(2):132. <https://doi.org/10.3390/buildings12020132>

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.