Turkish Journal of Engineering
https://dergipark.org.tr/en/pub/tuje
e-ISSN 2587-1366

Predicting the compressive strength of self-compacting concrete using artificial intelligence techniques: A review

Terlumun Sesugh*[1], Michael Onyia[2], Okafor Fidelis [2]

[1] Air Force Institute of Technology, Department of Civil and Environmental Engineering, Nigeria, terlumunsesugh8@gmail.com
[2]University, University of Nigeria Nsukka, Department of Civil Engineering, Nigeria, michaelonyia2023@gmail.com, okaforfo2023@gmail.com

Cite this study: Sesugh, T., Onyia, M., & Fidelis, O. (2024). Predicting the compressive strength of self-compacting concrete using artificial intelligence techniques: A review. Turkish Journal of Engineering, 8 (3), 537-550

https://doi.org/10.31127/tuje.1422225

Keywords
Compression strength prediction
Self-compacting concrete
Artificial intelligence

Review Article

Received:18.01.2024
Revised: 13.03.2024
Accepted: 14.03.2024
Published: 11.07.2024

Abstract

Concrete is one of the most common construction materials used all over the word. In estimating the strength properties of concrete, laboratory works need to be carried out. However, researchers have adopted predictive models in order to minimize the rigorous laboratory works in estimating the compressive strength and other properties of concrete. Self-compacting concrete which is an advanced form of construction is adopted mainly in areas where vibrations may not be possible due to complexity of the form work or reinforcement. This work is targeted at predicting the compressive strength of self-compacting concrete using artificial intelligence techniques**.** A comparative performance analysis of all techniques is presented. The outcomes demonstrated that training in a Deep Neural Network model with several hidden layers could enhance the performance of the suggested model. The artificial neural network (ANN) model, possesses a high degree of steadiness when compared to experimental results of concrete compressive strength. ANN was observed to be a strong predictive tool, as such is recommended for formulation of many civil engineering properties that requires predictions. Much time and resources are saved with artificial intelligence models as it eliminates the need for experimental test which sometimes delay construction works.

1. Introduction

The world's most popular building material has historically been concrete [1]. It has become man's most trusted companion due to its widespread use in construction projects across the world. At least since the Roman Empire's era, concrete has been instrumental for the expansion and improvement of many different cultures. In addition to its fluid-to-solid transformation, the "magic" of concrete lies in its inexpensive price, vast availability, malleability, plasticity, adaptability, high compressive strength, stiffness, and durability [1-4]. Many researchers are still enthusiastic about learning more about this niche of concrete engineering. Even for novel substances like reactive powder concrete, the future seems bright. Concrete's compressive strength ranges from 200 to 800 MPa [2-4].

It is standard practice to take a sizable enough sample of a combination to draw conclusions about the entire concrete mix. According to [5-7], testing concrete on day 7 is used to establish the early-age strength. Since concrete achieves its full strength at 28 days, the examination of the samples would also take that long. The result of the 28-day test can be used as a guide to predict the concrete's compressive strength as it ages. The procedures resemble those described in design regulations. However, these design standards fall short of the mark when concrete ingredients move beyond the traditional cement, aggregate, and water. The calculation of a concrete mix's compressive strength grows more complicated as the sorts of components used to manufacture it vary. Additionally, unaccounted-for components like pozzollans and admixtures are utilized when the type of concrete varies (for example, high strength concrete), rendering it impossible to clearly define the inter-relationship of concrete ingredients at the molecular level [8-12]. In cases when the outcome of the compressive strength might be influenced by factors

like "effect of admixtures," "temperature," etc., the empirical relationship suggested by the codes falls short of capturing the real picture, then, a trustworthy model is necessary to ascertain the strength of concrete samples at different ages in order to complete urgent projects that cannot wait for the results of 28-day testing. The time saved waiting for test results can be converted into monetary gains due to the productive nature of the construction process. This requires looking into alternate approaches. There is much room for debate and exploration regarding the usefulness of these non-traditional approaches.

There has been a lot of interest in utilizing artificial neural networks (ANN) to calculate concrete compressive strength over the previous year's [13, 14]. Several researches have used combinatory models, such as fuzzy logic (FL), genetic programming (GP), artificial neural networks (ANN) etc. [15] forecast how ground granulated blast furnace slag will affect concrete's strength over time. Fast learning with only moderately accurate performance was made possible by the cascade correlation type of ANN [16-18] in order to capture the inherently nonlinear character of patterns in the concrete properties. For a certain category of problems, specifically the measurement of compressive strength, this helps save time and money. The outstanding performance outcomes of combinatorial formations of artificial neural networks and meta-heuristic algorithms are making them increasingly popular for dealing with challenging structural engineering difficulties. Concrete compressive strength determination for deep beams attached to sheared walls, for instance, has been shown to benefit from a hybrid multilayer perceptron (HMP) network according to [18-22]. Artificial neural networks (ANNs) and evolutionary search methods like genetic algorithms (GAs) are combined in evolutionary artificial neural networks (EANNs) as described by [20, 21]. Kovačević [22] lays out the framework for evolutionary artificial neural networks (EANNs). EANNs have been tremendously effective for the purpose of detecting defects in concrete's structure and compressive strength [22-24]. To more precisely and affordably estimate the concrete compressive strength, EANN is being studied as a potential replacement for both costly mathematical models and damaging trials.

2. Method

The authors focused on machine learning models and the compressive strength of self-compacting concrete in order to achieve the goals of this study: A range of websites, including Scholarly Journals, Elsevier, Springers, SCOPUS, the Web of Science, Turkish Journal of Engineering, IEEE, and Science Direct, were consulted in order to acquire scholarly sources for this study. Only published books that the authors felt offered the most intelligent and pertinent answers to their research topic were chosen. Following an examination of these sources, 81 sources in all were determined to be very relevant to the study. Figure 1 presents a flowchart that outlines the research approach used in this work.

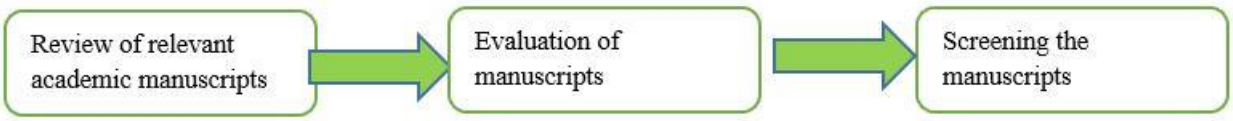

Figure 1. Flow chart of the approach used in obtaining relevant articles for the research.

A total of 80 articles that were researched came from academic journals, which accounted for a greater percentage of the total number of academic sources, also EFNARC guidelines and specifications for self-compacting concrete was also used in this research. Thus, the total number of academic sources was 81. The study applies the idea of modern construction technology to give a methodical assessment of the compressive strength of self-compacting concrete using machine learning models.

2.1. Predicting concrete compressive and flexural strength

There are numerous techniques for predicting compressive strength as adopted by many researchers across the world.

2.1.1. Empirical methods

In an attempt to directly represent the experimental procedure into a given connection, empirical methodologies have been developed. The strength of the connection indicates how important each component was in generating the final result. The created connection must be able to suit the experimental data; therefore, they often rely on extensive experimental analysis. Most empirical relationships for forecasting compressive strength make an effort to connect the strength to the water cement ratio, despite the fact that this ratio typically has a negative effect on the predicted strength. Other times, the models take into account information about compressive strength that is already available and employ experimentally estimated coefficients to establish a connection between the available information and the necessary information [10, 25].

2.1.2. Computational modelling

Complex equations of thermodynamics are the foundation of finite element analysis. Such modelling is often done with the use of computers. They rely on an accurate depiction of the microstructure of the concrete. In order to simulate accurately the hydrations (and other processes) of the different particle sizes in the cement using a computer simulation (pixel based), it is necessary to cast the cement randomly in a unit cell space. It is also possible to encode the processes empirically, after adjusting for experimental data [10, 25, 26].

2.1.3. Mechanical modeling

Parameters in mechanical models are typically represented using a spring-and-dashpot structure. The cement matrix is typically viewed as the spring and the time element (age of testing) as the dashpot when applying this theory to forecast the compression strength of concrete. When the model is compared to a testing dataset, its accuracy declines. Compressive strength is also significantly impacted by the dashpot component, especially at younger ages [10, 25, 26].

2.1.4. Statistical methods

In order to provide the most accurate description of the connections between variables, statistical techniques rely on experimental data and apply mathematical equations. In the realm of statistics, multilinear regression is by far the most used approach. Although statistical approaches are intuitive and simple to implement, their data-heavy nature might be a hindrance in some situations. Also, their effectiveness varies with the mathematical function used to fit the data, making them less reliable than alternative approaches [10, 25-27].

2.1.5. Regression analysis

It is widely agreed that regression analysis is a crucial part of any respectable statistical modelling strategy. The convenience of the statistical method in terms of computing coefficients, which may be interpreted in terms of efficiency gains, [28-30], such an approach, however, is typically convoluted and difficult to follow. This trait is ignored for the time being, in order to create a confidence interval for a set of predictions, statistical models must be mathematically rigorous. Correlation analysis also shows how the main constituents affect the final product. R^2, sometimes termed the correlation coefficient, is a metric used to compare the effectiveness of different regression equations. R^2 measures how well a model can make predictions. It's a metric for assessing the model's ability to account for differences in output that result from differences in input values. A value of 0 indicates that no variance in Y is interpreted by the calculated regression model, whereas a value of 1 indicates that all points are on the regression line. Slump test findings and concrete density were presented by [28-30] to complement mix proportions of component materials in estimating high performance concrete's compressive strength.

This was done so that results from tests at varying ages would more accurately reflect the progression from weakness to strength. The final multi-variable power equation benefited from the addition of these independent variables, with a correlation coefficient of 99.99% being achieved. Good connections between his predicted compressive strength and experimental data were also discovered in the work of [31]. The inputs to his model were standardized according to a fixed matrix formula in order to enhance development of concrete mixtures. Regression methods have long been popular because of their simplicity and the little amount of time they require for modelling. Each regression model has great predictive power for low levels of non-linearity between reactants and products (as is typically the case in concrete). This is evident in the work of [32, 33], in which he attempted to evaluate this strategy in comparison to the artificial neural network approach and concluded that the latter provided much superior forecasts.

2.1.6. Artificial intelligence

According to Webster's New World College Dictionary, AI is "the ability of machines or programs to function in ways that simulate human intelligence in tasks such as reasoning and learning." One can then wonder what kinds of issues call for a computer solution that "thinks like a human brain." Knowledge-based inference with partial or unclear data, different types of perception and learning, and control, prediction, classification, and optimization are only a few examples [34, 35]. They can be used to mimic a phenomenon in civil engineering situations where the underlying process is unclear. Neural networks and genetic algorithms are two types of AI techniques; they have structural and functional similarities to microscopic biological models (neural networks that mimic the brain's) and genes (computer programs that 'mutate' to improve their performance), respectively. Adaptive Network based Fuzzy Inference Systems (ANFIS) Fuzzy systems are two further examples of modern artificial intelligence. Artificial Neural Network Fuzzy Inference System (ANFIS) uses a hybrid of the two. Training using linguistic interpretation of variables with fuzzy logic and neural networks are both strong suits of the network created in this way [34, 35]. Methods based on artificial intelligence are potent due to their ability to learn and provide relevant models, even for a wide variety of data. A 'near to reality' conclusion, however, requires more data than just this.

2.2. Fuzzy logic

It was Lotfi Zadeh who originally proposed the use of fuzzy logic [33-37]. Among the various fuzzy inference system approaches he paved the way for include Mamdani, Takagi, and Sugeno. His work on fuzzy logic is credited with revolutionizing the way people represent the world.

According to [33-37], fuzzy logic is a collection of mathematical principles for knowledge representation based on degrees of membership. This stands in stark contrast to the logical values in traditional Boolean logic of either "completely false" (=0) or "completely true" (=1). The Boolean logic is a representation of information having binary boundaries (black or white, for example). The truth usually differs significantly from expectations. Parameters within a set are not always well delineated, and a "grey area" often exists when there is an element that belongs to both sets to varying degrees. An individual may, for instance, have the heights of 10 persons in a set, with the 'tall' group ranging from 1.75 m to 2.20 m and the ‘short' group ranging from 1.50 m to 1.74 m. Consideration of a crisp set may seem insufficient

under these conditions. Some researchers [33-37] summed up this kind of dichotomous thinking by saying that it is "painfully simplified" and "in many circumstances, lacking rapport with reality." Fuzzy sets can be used to simulate the aforementioned crisp set more accurately and robustly. They [33-37] define a fuzzy set as a class of objects having a continuum of degrees of membership. A membership (characteristic) function, in this case, gives each item in the set a score between one and zero based on its membership degree in the set.

Modelling non-linear systems and creating sophisticated controllers are two of the many applications of fuzzy logic control, a strong mathematical tool. In concrete, each component serves a purpose both alone and in concert with the others. This will cause the relationship between the number of ingredients and the resulting compressive strength to be non-linear. FLC can help reduce this nonlinearity since it is implemented when the complexity of a system prevents the use of other modelling approaches [33-37].

2.3. Neural network

The dramatic shift in how people approach problems that neural networks represent is happening now. Since work on artificial intelligence began in the 1950s, neural networks have been at the forefront of efforts to expand robots' limited utility from muscle labor to intellect work. The term "artificial neural network" can refer to a number of different things. In that respect, they are analogous to biological brain networks. In this metaphor, the axon represents the outputs and the synapses represent the weights. In artificial neural networks, the ubiquitous neuron is also referred to as a "processing element." References [36-39] concisely defined artificial neural networks as a class of massively parallel designs that address complex issues by coordinating the efforts of several, relatively uncomplicated processors (or "artificial neurons"). A perceptron is a simple neural network with a one-to-one mapping of inputs to outputs. More complex neural networks may have many layers and use a variety of activation functions. Classification of artificial neural networks helps shed light on their many forms.

Damage detection, identification of structural systems, modelling of material behavior, structural settlement, control and optimization, monitoring ground water, and determining concrete mix proportions are just some of the many areas where ANNs have been put to use [40]. Reference [41] examined the use of artificial neural networks in prediction models to supplement their work in fuzzy logic and attempt to forecast the compressive strength of self-compacting concrete. 500 iterations were performed on the ANN model, which had a hidden layer of 6 neurons. Their successes were comparable, as were the degrees of inaccuracy in their measurements. The resultant model improved upon the fuzzy logic result in that it was able to estimate the strength with an R^2 value of 0.9767. The selected ANN design had a single hidden layer with eleven secret neurons. Adding a second hidden layer improved the accuracy of the model [42]. Using an ANN design with two hidden layers and nine and eight neurons in the first and second hidden layers, respectively, yielded the lowest absolute percentage error (=0.000515). Sarıdemir [43] also considered the advantages of a multi-layer architecture. The material issue was also given proper consideration in his model, as seen by the use of chemical analysis data for fly ash, gradient, and sand chemical compositions. R^2 = 0.9557 was found for compressive strength, while 0.9119 was found for flexural tensile strength, indicating that the prediction model performed well. The influence of water to binder ratio on compressive strength was also considered in the compressive strength forecasting model [43, 44]. Compressive strength was shown to decrease with increasing water-to-binder ratio. Their model has an R^2 of 0.9944 for reproducing experimental outcomes and an R^2 of 0.9767 for predicting testing samples. Concrete strength prediction methodologies were compared with an artificial neural network approach [45-47], the results showed that the ANN approach provided an accurate forecast for low and medium strength concretes. Despite promising findings from a multiple linear regression model, the ANN method performed much better.

2.4. Genetic programming

In order to evaluate a computer's performance on a given job, genetic programmers use a set of "instructions" and a "fitness process" known as "genes expression programming" (GEP). Each node is a piece of code running on a computer, making this a subtype of genetic algorithm (GA). It's a method for improving the performance of a computer programme by moving its components to the most advantageous physical location, as defined by the conditions the it must satisfy. Here are the three genetic manipulations [43-46].

Some of the key adjusting factors are detailed in [43-46] article. The GP building process begins with an initial population size of 49 without cement/FA substitution and 27 with cement/FA substitution of 0.15. The curing time was used to further classify each dataset in the GP model, just as it did in the ANN model. When the desired result is a 28-day compressive strength, four input parameters were selected: water, cement, coarse aggregate, and fine aggregate.

Reference [44-46] presented regression equations for predicting in situ concrete compressive strength based on data from ready-mixed concrete mixture proportions and on-site compressive strength testing. They employed 1442 compressive strength test data from 68 distinct mixture types with specified compressive strengths between 18 and 27 MPa, water to cement ratios between 0.39 and 0.62, and maximum sizes of aggregate was between 25 and 100 mm. They [45-47] tested a proposed model for predicting the compressive strength of concrete using in situ data.

2.5. Parameters in machine learning models

The hyper parameters of a machine learning algorithm are external guidelines that inform decision making and control how the algorithm learns. Machine learning engineers must first set these parameters before

training the algorithm, [48, 49, 50]. Hyper parameters include things like the learning rate, the number of clusters in a clustering algorithm, and the number of branches on a regression tree. In reaction to the training data, parameters start to form as the algorithm is trained and guided by the hyper parameters. The weights and biases that the algorithm develops during training are among these parameters. The model parameters are the final values for a machine learning model that, in an ideal world, fit a data set without exceeding or falling short of [48-50].

2.6. Hyper parameters in machine learning models

A hyper parameter in machine learning is a parameter whose value directs the process of learning. On the other hand, training is used to determine the values of other parameters, which are usually node weights. Hyper parameters can be divided into two categories: model hyper parameters, which pertain to the model selection task and cannot be inferred during machine learning, and algorithm hyper parameters, which impact the speed and caliber of learning but theoretically have no effect on the model's performance. The topology and size of a neural network are two instances of model hyper parameters [48-50].

Hyper parameters of an algorithm include learning rate, batch size, and mini-batch size. A mini-batch size would be a smaller sample set, whereas batch size can refer to the entire data sample. While some straightforward methods (like ordinary least squares regression) require none, other model training procedures require different hyper parameters. The training algorithm extracts the parameters from the data and applies them to these hyper parameters. The selection of a model's hyper parameters can affect how long it takes to train and test. Typically, a hyper parameter is of an integer or continuous form, which results in mixed-type optimization issues. Some hyper parameters can only exist if other hyper parameters are valued, for example, in a neural network, the number of layers may affect the size of each hidden layer [48-50].

2.7. Untrainable parameters

Because hyper parameters aggressively boost a model's capacity and have the potential to push the loss function to an undesirable minimum (overfitting to and picking up noise in the data), rather than accurately mapping the richness of the structure in the data, they can sometimes prevent hyper parameters from being learned from the training set. The degree of a polynomial equation that fits a regression model, for instance, would increase until the model perfectly suited the data if we treated it as a trainable parameter. This would result in reduced training error but poor generalization performance.

2.8. Tuning in ML

Only a few hyper parameters account for the majority of performance variation. How much more performance can be obtained by tweaking an algorithm, hyper parameter, or combination of hyper parameters is called its tuning ability. The two most important hyper parameters for machine learning models are learning rate and network size; batching and momentum have no discernible impact on the model's performance. Research has shown that mini-batch sizes between 2 and 32 yield the best results, despite some studies advocating for mini-batch sizes in the thousands.

2.9. Robustness of ML models

Learning has an inherent stochasticity, which suggests that the performance measured by the empirical hyper parameter may not accurately reflect the true performance. Without considerable simplification and robustness, methods that are not resistant to straightforward modifications in hyper parameters, random seeds, or even various implementations of the same algorithm, cannot be incorporated into mission-critical control systems. In particular, algorithms for reinforcement learning must have their performance evaluated across a large number of random seeds and their sensitivity to hyper parameter selection evaluated. Because of the high variance, their evaluation using a small number of random seeds is unable to fully capture performance. Certain reinforcement learning techniques—like Deep Deterministic Policy Gradient (DDPG) are more responsive to selections of hyper parameters than others.

2.10. Optimization

Through the use of hyper parameter optimization, a tuple of hyper parameters is found to produce an optimal model on test data that minimizes a predetermined loss function. The corresponding loss is returned by the objective function after receiving a tuple of hyper parameters.

2.11. Reproduction in ML models

In addition to fine-tuning hyper parameters, machine learning entails reproducibility checks, parameter and result organization, and storage. When there's no strong infrastructure in place for this kind of thing, research code tends to change quickly and jeopardizes important features like reproducibility and bookkeeping. Machine learning online collaboration platforms go one step further by enabling scientists to automatically exchange, arrange, and communicate about experiments, data, and algorithms. Deep learning models can be especially challenging to reproduce [51].

2.12. Creating machine learning models

Using either labeled, unlabeled, or a combination of the two, algorithms are trained to create machine learning models [51,52]. There are four main machine learning algorithms available:

2.12.1. Supervised learning

When an algorithm is trained on "labeled data," or data that has been labeled so that an algorithm can

successfully learn from it, supervised learning takes place. Training labels aid in ensuring that the final machine learning model is capable of classifying data in the way that the researcher has intended.

2.12.2. Unsupervised learning

To train an algorithm, unsupervised algorithms use unlabeled data. The algorithm generates its own data clusters during this process by looking for patterns in the data itself. Researchers who are trying to discover patterns in data that are currently unknown to them can benefit from unsupervised learning and pattern recognition.

2.12.3. Semi-supervised learning

This technique trains an algorithm by combining labeled and unlabeled data. This procedure involves training the algorithm with a small amount of labeled data initially, and then training it with a much larger amount of unlabeled data.

2.12.4. Reinforcement learning

This machine learning approach assigns positive and negative values to actions that are desired and undesirable. In order to maximize rewards through trial and error, the intention is to incentivize programs to steer clear of the unfavorable training examples and toward the positive ones. Unsupervised machine learning can be guided by reinforcement learning.

2.13. Types of machine learning models

Classification and prediction problems are the two main categories of machine learning problems. Models derived from algorithms created for either classification or regression (a technique used for predictive modeling) are used to approach these problems [53]. Depending on how it is trained, an algorithm may occasionally be used to produce regression or classification models. A list of well-known algorithms for building regression and classification models can be seen below. Models of classification Support vector machines, random forests, decision trees, logistic regression, naive Bayes, and K-nearest neighbor (KNN) Models of regression K-nearest neighbor (KNN) regression, decision trees, random forests, neural network regression, and linear regression

2.14. Overfitting in machine learning (ML)

In machine learning (ML), overfitting occurs when a model performs poorly on new, unseen data because it is too closely aligned with the training data it was trained on. Its primary goals are to detect, enable the model to generalize to new data, and guarantee that machine learning becomes less specialized to the training set of data [54,55]. Overfitting is used in machine learning tasks such as image recognition and natural language processing. After overfitting is eliminated, these models may also be able to predict new data more precisely and produce reliable forecasts for practical uses. It happens when there is insufficient data, when there is a lot of unnecessary information in the data set, when training on a specific data set is prolonged, and when the model's complexity needs to be reduced. In machine learning (ML), the term "overfitting" refers to the situation in which a model becomes excessively complex and fits too closely to the training set of data. Helping the model achieve better generalization performance on new data is the main goal of addressing overfitting. It can be found by keeping an eye on all losses, analyzing the learning curve, incorporating regularization terms, running cross-validation on the model, and visually examining the prediction to make sure it closely matches the training set.

The following methods can be used to avoid overfitting in machine learning: dropout, feature selection, early stopping, cross-validation, regularization using L1 and L2, and data augmentation. When a machine learning model becomes overly specialized in the training data and is unable to generalize well to new, unseen data, this is known as overfitting. This occurs in neural networks when the machine learning model gives less significant information in the training data more weight. Because it is unable to distinguish between relevant, crucial data that form the pattern and noisy data, this model requires assistance in order to produce accurate forecasts about new data.

2.15. Causes of overfitting

The model's training dataset is dirty and contains a lot of noise; additionally, the training dataset should be larger, when only a portion of the available data is used to build the model, resulting in an inaccurate representation of the entire dataset. Therefore, by looking at validation metrics such as accuracy and loss in overfitting decision trees, one can identify an overfitting model. Due to the effect of overfitting, these metrics typically have a tendency to rise to a certain point before beginning to decline or plateau. In addition, this model works to achieve an ideal fit, and once it does, the trend of these metrics starts to flatten or decline. To solve this, it is crucial to strike the correct balance between the complexity of the model and the amount of training data that is available.

2.16. Identifying overfitting in ML models

The overfitting can be detected in a few different ways, including the following:

1. All losses need to be kept an eye on, even if the validation loss rises while the training loss falls.
2. It is important to keep a close eye on the machine learning curve because any divergence between the training and validation curves indicates overfitting. Since the regularization term avoids overfitting, it must be included in the loss function.
3. Lastly, visual inspection of the model's prediction is necessary to determine whether the model fits the training data too closely. If so, this indicates overfitting.

2.17. Preventing ML from overfitting

The following methods are among the many that aid in preventing overfitting in machine learning:

1. The model increases the loss function by adding a penalty term through the joint Regularization L1 & L2 techniques. It thus deters the model from fitting the training set too closely, which avoids overfitting.
2. Cross-validation aids in its avoidance. The training automatically ends just before the point of overfitting when the early stopping technique in the model is applied.
3. Using the data augmentation technique improves the creation of new data from previously used training data. This keeps the model from overfitting by increasing its exposure to a wider range of data samples.
4. The model can be trained to prevent overfitting to noisy data by reducing the number of features through feature selection.
5. The dropout technique forces the model to learn more robust and generalizable representations, preventing data overfitting. Consequently, by using these techniques, the overfitting issue may be lessened, leading to the creation of a machine-learning model that is more precise and useful.

2.18. Fuzzy logic approach

The four main components of each fuzzy system are, Fuzzy rule base, Fuzzification, and Defuzzification and Inference Engine [56]. Figure 2 present a general fuzzy logic model architecture.

The Fuzzification is the first step that estimates the fuzzy input and converts it into one or more fuzzy sets. Gaussian and trapezoidal fuzzifications are the two types. In that kind of fuzzy logic, any item can belong to any number of subsets of the universal set. The foundation of fuzzy rules is linguistic IF-THEN-constructions with the overall procedure "IF A THEN B," where A and B are schemes incorporating linguistic variables. The premise is designated as A, while the rule's importance is designated as B. In order to account for ambiguity and inaccuracy, linguistic variables and fuzzy IF-THEN rules are used. Depending on how unique the challenges are, there are two different types of rule techniques. In a fuzzy inference engine, fuzzy rule base is integrated and used in the analysis of fuzzy outputs [56,57]. The Defuzzification process transforms outputs from a fuzzy inference engine into precise numerical values and crisp output. The flow chart for the GP is shown Figure 3.

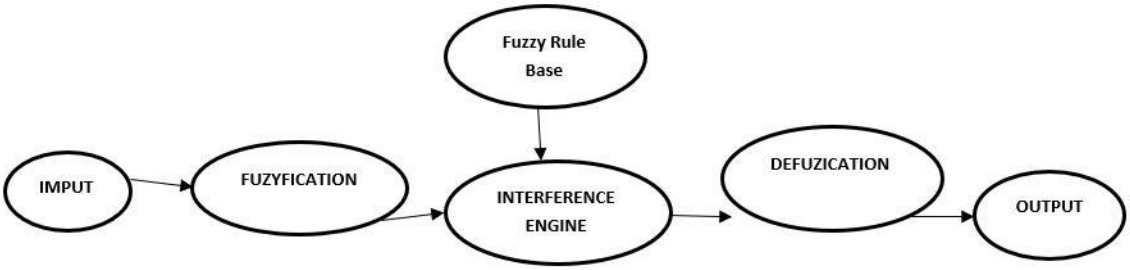

Figure 2. Fuzzy Logic architecture.

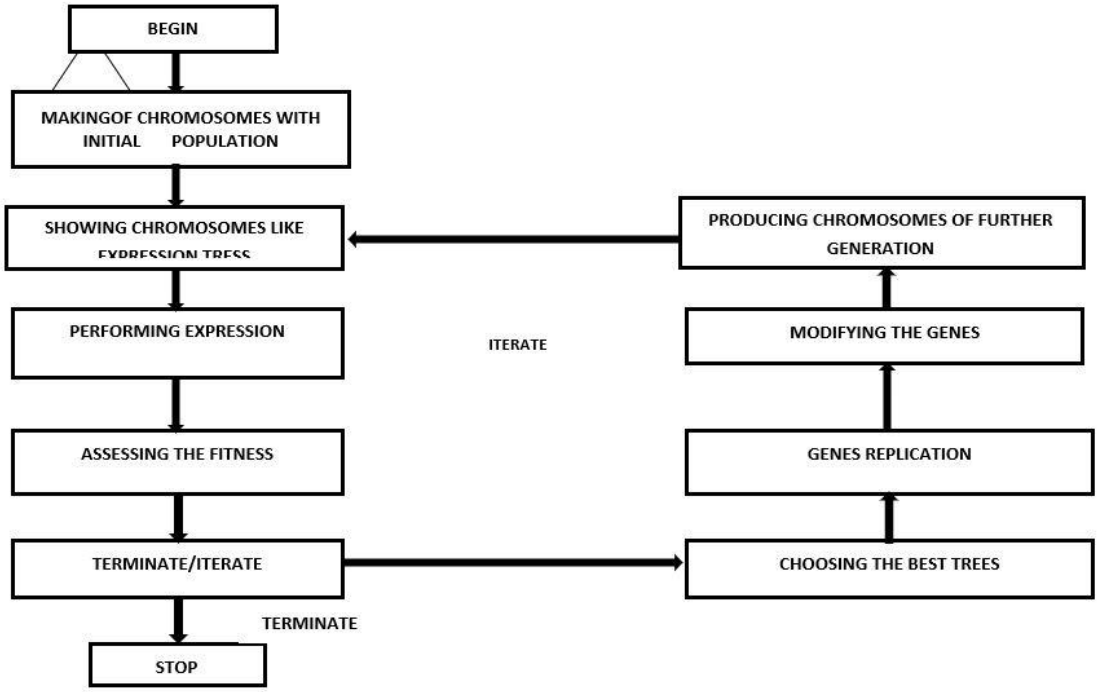

Figure 3. Genetic algorithm flow chart diagram.

2.19. Evaluating the performance of the GP model

For a model to be deemed adequate, it is recommended that the dataset ratio to the total number of input features equal 3, and a ratio of 5 is desired. The GEP model can be validated using various statistical computations from the training, testing, and validation sets. In order to assess if the model was effectively trained and whether there is a significant association between the model and experimental data with little error, the values of the RMSE, MAE, and RSE parameters will be computed for the testing phase [56-58].

With the aid of several statistical techniques, the GEP model will also be examined for external validation. It is suggested that one of the regression lines (k or k′) passing through the origin's slope should be close to 1. The same dataset will also be used in this study's linear regression model, which will be used to calculate the SCC's fc. It is important to remember that the robustness and generalizability of the resulting model depend on the fitting parameters. The fitting parameters for the GEP algorithm will be determined using test runs or experimental results. How long the program will run depends on the population size (the number of chromosomes). Depending on the quantity and complexity of the prediction model, the levels will be taken into account as the population size. The head size and the number of genes were two variables that the algorithm that created the models in this study used to define the architecture of the models [59-61].

The head size, or "head size," of the model, which refers to the size of the model's "head," contributes to determining how complex each term is inside the model. On the other hand, the number of sub-ETs (basic data structures) that make up the model depends on the number of genes. Five alternative head sizes will be considered in this study: 8, 9, 12, 10, and 14. There will be either three or four genes [61, 62]. The GEP algorithm will be used to develop the precise parameters for each model.

2.20. Model evaluation criteria

One common performance indicator is the correlation coefficient (R). R is insensitive to the division and multiplication of output values by constants, hence it cannot be used as a primary indicator of how well the model predicts. As a result, the study will also evaluate the mean absolute error (MAE), relative squared error (RSE), and relative root mean square error (RRMSE) [63-67]. To evaluate the model's performance in regard to both the RRMSE and R, Gandomi and Roke propose a performance index [56]. Equations 1 to 7 provide the mathematical formulas for these error functions.

$$Fc = f(\alpha + \beta_1 Y_1 + \beta_2 Y_2 + \beta_3 Y_3 + \beta_4 Y_4 \ldots\ldots + \beta_n Y_n) + £_1 \quad (1)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(ei - mi)^2}{n}} \quad (2)$$

$$MAE = \frac{\sum_{i=1}^{n}|ei - mi|}{n} \quad (3)$$

$$RSE = \frac{\sum_{i=1}^{n}(mi - ei)^2}{\sum_{i=1}^{n}(\bar{e} - ei)^2} \quad (4)$$

$$RRMSE = \frac{1}{|\bar{e}|}\sqrt{\frac{\sum_{i=1}^{n}(ei - mi)^2}{n}} \quad (5)$$

$$R = \frac{\sum_{i=1}^{n}(ei - \bar{e}i)(mi - \overline{mi})}{\sqrt{\sum_{i=1}^{n}(ei - \bar{e}i)^2 \sum_{i=1}^{n}(mi - \overline{mi})^2}} \quad (6)$$

$$\rho = \frac{RRMSE}{1 + R} \quad (7)$$

where Fc is the compressive strength, β1 to βn is the regression coefficients, α is the regression constant, $£_{1\ is}$ the error term

2.21. Artificial neural network

The compressive and flexural strength of the mixed design specimens will be measured. Following casting for 7, 14, 21, and 28 days, all mix design samples will be examined. Compressive strength will be determined using cube samples, and flexural strength will be determined using beam specimens. After 28 days of casting, cube samples will be weighed to determine their density [68-71]. The flow chart for the ANN is shown in Figure 4.

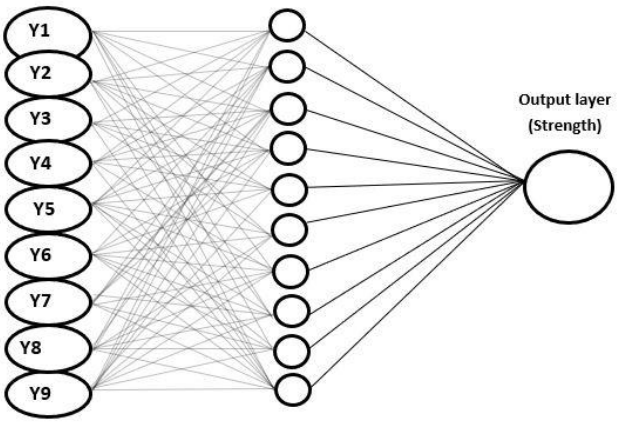

Figure 4. ANN architecture.

Reference [72-75], A evaluated the feasibility of MARS and GEP models in predicting the compressive strength of SCC at 28 days. In their research, values of compressive strength were estimated using multivariate adaptive regression spline approach [76-81]. The research indicated high ability of both the GEP and MARS models in predicting SCC strength. The parameters selected for

the GEP models were defined as shown in Table 1. From the work, predictive models were developed and the model tree shown in Figure 5 while Figure 6 represent the scatter plot for training of the GEP models and Figure 7 shows the scatter plot for testing of the GEP models.

Table 1.: Parameters of optimized GEP models [80].

Parameters	Description of parameters	Setting of parameters
P1	Function set	+,-, x,l,exp, power
P2	Mutation rate	0.138
P3	Inversion rate	0.546
P4	One point and two point recombination rate	0.277
P5	Gene recombination rate	0.277
P6	Gene tramsportatiom rate	0.277
P7	Maximum tree depth	6
P8	Number of genes	3
P9	Number of chromosones	30

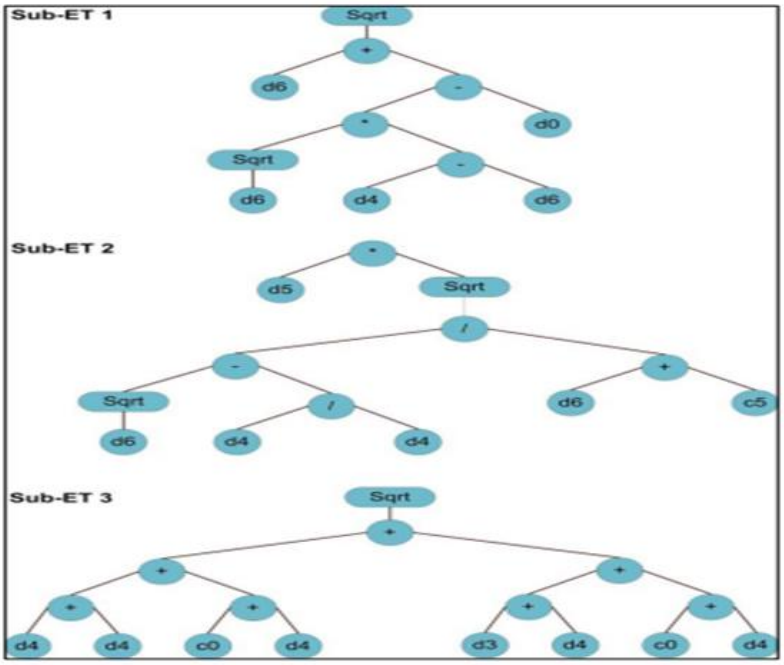

Figure 5. GEP models in concrete strength prediction [80].

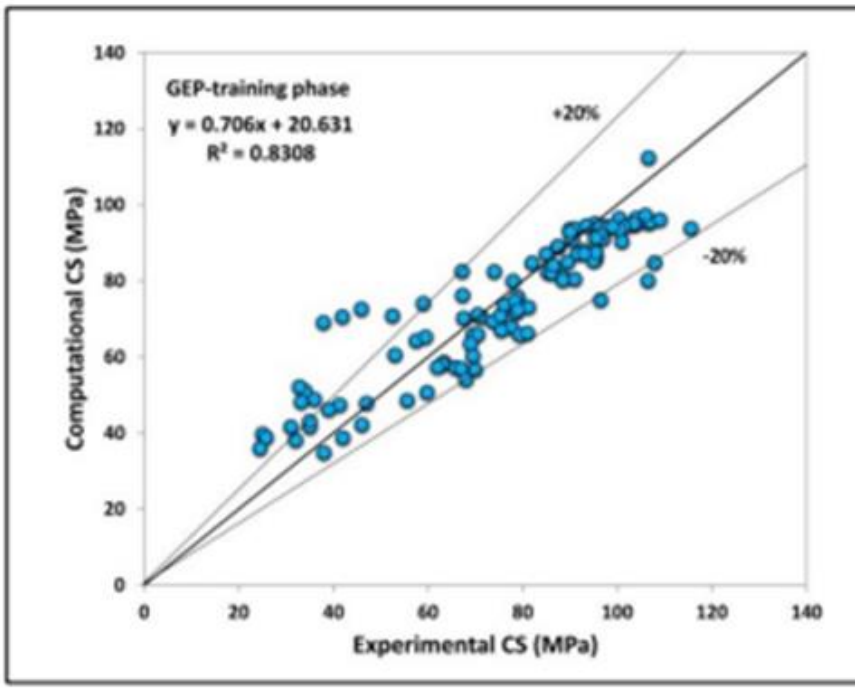

Figure 6. Scatter plot of observed and predicted compressive strength for training of GEP model [80].

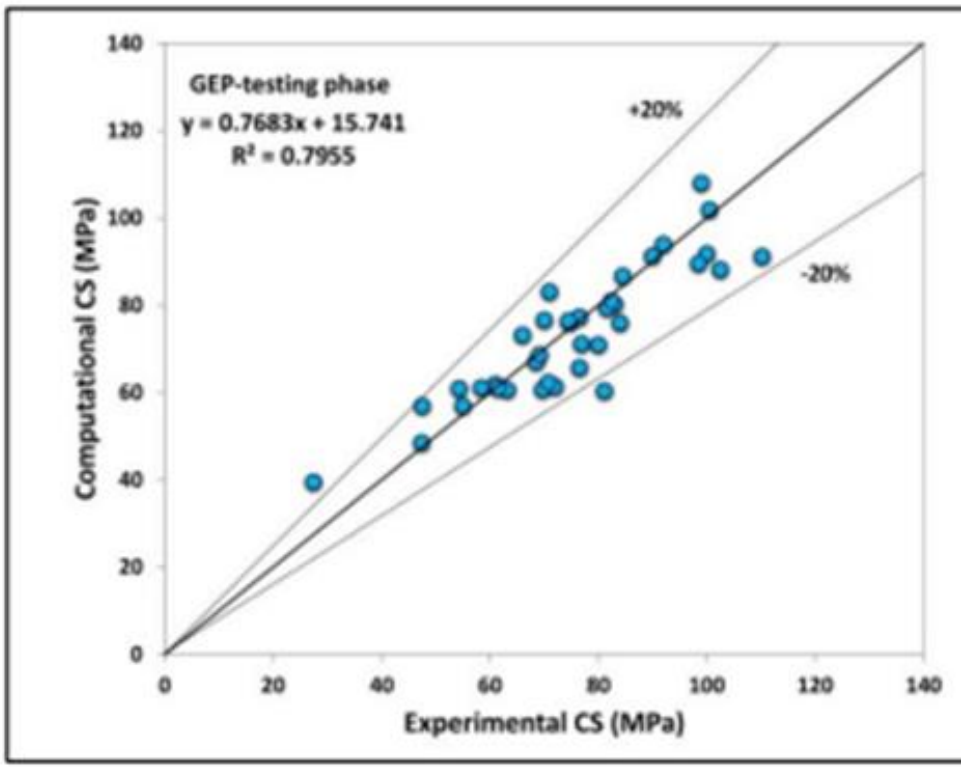

Figure 7. Scatter plot of observed and predicted compressive strength for testing of GEP model [80].

3. Conclusion

From the findings of this work, it was observed that use of artificial intelligence techniques in predicting the compressive strength of self-compacting concrete give approximate strength values close to experimental investigations. The research has shown that if a model gives R^2 > 0.8, then there is a correlation between predicted and experimental values for the available data in the dataset. The review observed that all the techniques studied, R^2 was always greater than 0.8 for 28 days' compressive strength which proves both of the models can be used for accurate prediction. The ANN model was identified to possesses a high degree of steadiness when compared to experimental results of concrete compressive strength, hence, ANN is a strong predictive tool, for both in situ and experimental prediction and as such is recommended for formulation of many civil engineering properties that requires predictions. Much time and resources are saved with artificial intelligence models as it eliminates the need for experimental test which sometimes delay construction works. Reinforcement learning techniques like Deep Deterministic Policy Gradient (DDPG) are more responsive to selections of hyper parameters than others. Through the use of hyper parameter optimization, a tuple of hyper parameters can produce an optimal model on test data that minimizes a predetermined loss function. Overfitting occurs in machine learning when a model performs poorly on new or unseen data because it is too closely aligned with the training data that was used for the model but over fitting can be avoided through, cross validation, augmentations, drop out and feature selection techniques.

Acknowledgement

The authors appreciate the staff of the Department of Civil Engineering, University of Nigeria Nsukka for considering this work for a post graduate program, we remain grateful for all your recommendations to improve this work.

Author contributions

Sesugh Terlumun: Conceptualization, Methodology, Writing-original draft preparation of the manuscript
Onyia Michael: Visualization, Writing-Reviewing and Editing
Okafor Fidelis: Validation, Writing-Reviewing and Editing

Conflicts of interest

The authors declare no conflicts of interest.

References

1. Gaimster, R., & Dixon, N. (2003). Self-compacting concrete. Advanced Concrete Technology, 3, 1-23. https://doi.org/10.1016/B978-075065686-3/50295-0
2. Falliano, D., De Domenico, D., Ricciardi, G., & Gugliandolo, E. (2018). Experimental investigation on the compressive strength of foamed concrete: Effect of curing conditions, cement type, foaming agent and dry density. Construction and Building Materials, 165, 735-749. https://doi.org/10.1016/j.conbuildmat.2017.12.241

3. Lee, S. C. (2003). Prediction of concrete strength using artificial neural networks. Engineering Structures, 25(7), 849-857. https://doi.org/10.1016/S0141-0296(03)00004-X
4. Madandoust, R., & Mousavi, S. Y. (2012). Fresh and hardened properties of self-compacting concrete containing metakaolin. Construction and Building Materials, 35, 752-760. https://doi.org/10.1016/j.conbuildmat.2012.04.109
5. Tufail, R. F., Naeem, M. H., Ahmad, J., Waheed, H., Majdi, A., Farooq, D., ... & Butt, F. (2022). Evaluation of the fresh and mechanical properties of nano-engineered self compacting concrete containing graphite nano/micro platelets. Case Studies in Construction Materials, 17, e01165. https://doi.org/10.1016/j.cscm.2022.e01165
6. Shi, C., Wu, Z., Lv, K., & Wu, L. (2015). A review on mixture design methods for self-compacting concrete. Construction and Building Materials, 84, 387-398. https://doi.org/10.1016/j.conbuildmat.2015.03.079
7. Hamada, H., Alattar, A., Tayeh, B., Yahaya, F., & Thomas, B. (2022). Effect of recycled waste glass on the properties of high-performance concrete: A critical review. Case Studies in Construction Materials, 17, e01149. https://doi.org/10.1016/j.cscm.2022.e01149
8. Efnarc, S. (2002). Guidelines for Self-Compacting Concrete, Rep. from EFNARC. 44, 32.
9. Tejaswini, G. L. S., & Rao, A. V. (2020). A detailed report on various behavioral aspects of self-compacting concrete. Materials Today: Proceedings, 33, 839-844. https://doi.org/10.1016/j.matpr.2020.06.273
10. Danish, P., & Ganesh, G. M. (2021). Self-compacting concrete—optimization of mix design procedure by the modifications of rational method. In 3rd International Conference on Innovative Technologies for Clean and Sustainable Development: ITCSD 2020 3, 369-396. https://doi.org/10.1007/978-3-030-51485-3_25
11. Esmaeilkhanian, B., Khayat, K. H., Yahia, A., & Feys, D. (2014). Effects of mix design parameters and rheological properties on dynamic stability of self-consolidating concrete. Cement and Concrete Composites, 54, 21-28. https://doi.org/10.1016/j.cemconcomp.2014.03.001
12. Ashish, D. K., & Verma, S. K. (2019). An overview on mixture design of self-compacting concrete. Structural Concrete, 20(1), 371-395. https://doi.org/10.1002/suco.201700279
13. Bayer, İ. R., Turanlı, L., & Mehta, P. K. (2019). Mass concrete construction using self-compacting mortar. Turkish Journal of Engineering, 3(3), 110-119. https://doi.org/10.31127/tuje.462548
14. Kandiri, A., Golafshani, E. M., & Behnood, A. (2020). Estimation of the compressive strength of concretes containing ground granulated blast furnace slag using hybridized multi-objective ANN and salp swarm algorithm. Construction and Building Materials, 248, 118676. https://doi.org/10.1016/j.conbuildmat.2020.118676
15. Golafshani, E. M., Rahai, A., & Sebt, M. H. (2015). Artificial neural network and genetic programming for predicting the bond strength of GFRP bars in concrete. Materials and Structures, 48, 1581-1602. https://doi.org/10.1617/s11527-014-0256-0
16. Gesoğlu, M., Güneyisi, E., Özturan, T., & Özbay, E. (2010). Modeling the mechanical properties of rubberized concretes by neural network and genetic programming. Materials and Structures, 43, 31-45. https://doi.org/10.1617/s11527-009-9468-0
17. Belalia Douma, O., Boukhatem, B., Ghrici, M., & Tagnit-Hamou, A. (2017). Prediction of properties of self-compacting concrete containing fly ash using artificial neural network. Neural Computing and Applications, 28, 707-718. https://doi.org/10.1007/s00521-016-2368-7
18. Siddique, R., Aggarwal, P., & Aggarwal, Y. (2011). Prediction of compressive strength of self-compacting concrete containing bottom ash using artificial neural networks. Advances in Engineering software, 42(10), 780-786. https://doi.org/10.1016/j.advengsoft.2011.05.016
19. Behnood, A., & Golafshani, E. M. (2018). Predicting the compressive strength of silica fume concrete using hybrid artificial neural network with multi-objective grey wolves. Journal of Cleaner Production, 202, 54-64. https://doi.org/10.1016/j.jclepro.2018.08.065
20. Chakravarthy HG, N., Seenappa, K. M., Naganna, S. R., & Pruthviraja, D. (2023). Machine Learning Models for the Prediction of the Compressive Strength of Self-Compacting Concrete Incorporating Incinerated Bio-Medical Waste Ash. Sustainability, 15(18), 13621. https://doi.org/10.3390/su151813621
21. Behnood, A., Behnood, V., Gharehveran, M. M., & Alyamac, K. E. (2017). Prediction of the compressive strength of normal and high-performance concretes using M5P model tree algorithm. Construction and Building Materials, 142, 199-207. https://doi.org/10.1016/j.conbuildmat.2017.03.061
22. Kovačević, M., Lozančić, S., Nyarko, E. K., & Hadzima-Nyarko, M. (2021). Modeling of compressive strength of self-compacting rubberized concrete using machine learning. Materials, 14(15), 4346. https://doi.org/10.3390/ma14154346
23. Aiyer, B. G., Kim, D., Karingattikkal, N., Samui, P., & Rao, P. R. (2014). Prediction of compressive strength of self-compacting concrete using least square support vector machine and relevance vector machine. KSCE Journal of Civil Engineering, 18, 1753-1758. https://doi.org/10.1007/s12205-014-0524-0
24. Golafshani, E. M., Behnood, A., & Arashpour, M. (2020). Predicting the compressive strength of normal and High-Performance Concretes using ANN and ANFIS hybridized with Grey Wolf Optimizer. Construction and Building Materials, 232, 117266. https://doi.org/10.1016/j.conbuildmat.2019.117266
25. Ikeagwuani, C. C. (2021). Estimation of modified expansive soil CBR with multivariate adaptive regression splines, random forest and gradient

boosting machine. Innovative Infrastructure Solutions, 6(4), 199. https://doi.org/10.1007/s41062-021-00568-z
26. Onyia, M. E., Ambrose, E. E., Okafor, F. O., & Udo, J. J. (2023). Mathematical modelling of compressive strength of recycled ceramic tile aggregate concrete using modified regression theory. Journal of Applied Sciences and Environmental Management, 27(1), 33-42. https://doi.org/10.4314/jasem.v27i1.6
27. Aicha, M. B., Al Asri, Y., Zaher, M., Alaoui, A. H., & Burtschell, Y. (2022). Prediction of rheological behavior of self-compacting concrete by multi-variable regression and artificial neural networks. Powder Technology, 401, 117345. https://doi.org/10.1016/j.powtec.2022.117345
28. Patel, R., Hossain, K. M. A., Shehata, M., Bouzoubaa, N., & Lachemi, M. (2004). Development of statistical models for mixture design of high-volume fly ash self-consolidating concrete. Materials Journal, 101(4), 294-302. https://doi.org/10.14359/13363
29. Razavi Tosee, S. V., & Nikoo, M. (2019). Neuro-fuzzy systems in determining light weight concrete strength. Journal of Central South University, 26(10), 2906-2914. https://doi.org/10.1007/s11771-019-4223-3
30. Günal, A. Y., & Mehdi, R. (2024). Application of a new fuzzy logic model known as" SMRGT" for estimating flow coefficient rate. Turkish Journal of Engineering, 8(1), 46-55. https://doi.org/10.31127/tuje.1225795
31. Zhou, Q., Wang, F., & Zhu, F. (2016). Estimation of compressive strength of hollow concrete masonry prisms using artificial neural networks and adaptive neuro-fuzzy inference systems. Construction and Building Materials, 125, 417-426. https://doi.org/10.1016/j.conbuildmat.2016.08.064
32. Behnood, A., & Golafshani, E. M. (2020). Machine learning study of the mechanical properties of concretes containing waste foundry sand. Construction and Building Materials, 243, 118152. https://doi.org/10.1016/j.conbuildmat.2020.118152
33. Madani, H., Kooshafar, M., & Emadi, M. (2020). Compressive strength prediction of nanosilica-incorporated cement mixtures using adaptive neuro-fuzzy inference system and artificial neural network models. Practice Periodical on Structural Design and Construction, 25(3), 04020021. https://doi.org/10.1061/(ASCE)SC.1943-5576.0000499
34. Chiew, F. H., Ng, C. K., Chai, K. C., & Tay, K. M. (2017). A fuzzy adaptive resonance theory-based model for mix proportion estimation of high-performance concrete. Computer-Aided Civil and Infrastructure Engineering, 32(9), 772-786. https://doi.org/10.1111/mice.12288
35. Lee, S. C. (2003). Prediction of concrete strength using artificial neural networks. Engineering structures, 25(7), 849-857. https://doi.org/10.1016/S0141-0296(03)00004-X
36. Lai, S., & Serra, M. (1997). Concrete strength prediction by means of neural network. Construction and Building Materials, 11(2), 93-98. https://doi.org/10.1016/S0950-0618(97)00007-X
37. Ibrahim, S. M., Ansari, S. S., & Hasan, S. D. (2023). Towards white box modeling of compressive strength of sustainable ternary cement concrete using explainable artificial intelligence (XAI). Applied Soft Computing, 149, 110997. https://doi.org/10.1016/j.asoc.2023.110997
38. Erzin, Y. (2007). Artificial neural networks approach for swell pressure versus soil suction behaviour. Canadian Geotechnical Journal, 44(10), 1215-1223. https://doi.org/10.1139/T07-052
39. Silva, F. A., Delgado, J. M., Cavalcanti, R. S., Azevedo, A. C., Guimarães, A. S., & Lima, A. G. (2021). Use of nondestructive testing of ultrasound and artificial neural networks to estimate compressive strength of concrete. Buildings, 11(2), 44. https://doi.org/10.3390/buildings11020044
40. Duan, Z. H., Kou, S. C., & Poon, C. S. (2013). Prediction of compressive strength of recycled aggregate concrete using artificial neural networks. Construction and Building Materials, 40, 1200-1206. https://doi.org/10.1016/j.conbuildmat.2012.04.063
41. Alade, I. O., Bagudu, A., Oyehan, T. A., Abd Rahman, M. A., Saleh, T. A., & Olatunji, S. O. (2018). Estimating the refractive index of oxygenated and deoxygenated hemoglobin using genetic algorithm–support vector regression model. Computer Methods and Programs in Biomedicine, 163, 135-142. https://doi.org/10.1016/j.cmpb.2018.05.029
42. Iqbal, M. F., Liu, Q. F., Azim, I., Zhu, X., Yang, J., Javed, M. F., & Rauf, M. (2020). Prediction of mechanical properties of green concrete incorporating waste foundry sand based on gene expression programming. Journal of Hazardous Materials, 384, 121322. https://doi.org/10.1016/j.jhazmat.2019.121322
43. Sarıdemir, M. (2010). Genetic programming approach for prediction of compressive strength of concretes containing rice husk ash. Construction and Building Materials, 24(10), 1911-1919. https://doi.org/10.1016/j.conbuildmat.2010.04.011
44. Gandomi, A. H., & Roke, D. A. (2015). Assessment of artificial neural network and genetic programming as predictive tools. Advances in Engineering Software, 88, 63-72. https://doi.org/10.1016/j.advengsoft.2015.05.007
45. Khan, M. A., Zafar, A., Akbar, A., Javed, M. F., & Mosavi, A. (2021). Application of Gene Expression Programming (GEP) for the prediction of compressive strength of geopolymer concrete. Materials, 14(5), 1106. https://doi.org/10.3390/ma14051106
46. Faradonbeh, R. S., Hasanipanah, M., Amnieh, H. B., Armaghani, D. J., & Monjezi, M. (2018). Development of GP and GEP models to estimate an environmental issue induced by blasting operation. Environmental Monitoring and Assessment, 190, 1-15. https://doi.org/10.1007/s10661-018-6719-y
47. Khandelwal, M., Shirani Faradonbeh, R., Monjezi, M., Armaghani, D. J., Majid, M. Z. B. A., & Yagiz, S. (2017). Function development for appraising brittleness of intact rocks using genetic programming and non-

linear multiple regression models. Engineering with Computers, 33, 13-21. https://doi.org/10.1007/s00366-016-0452-3
48. Ferreira, C. (2002). Gene expression programming in problem solving. Soft Computing and Industry: Recent Applications, 635-653. https://doi.org/10.1007/978-1-4471-0123-9_54
49. Ferreira, C. (2006). Gene expression programming: mathematical modeling by an artificial intelligence, 21. Springer.
50. Gholampour, A., Gandomi, A. H., & Ozbakkaloglu, T. (2017). New formulations for mechanical properties of recycled aggregate concrete using gene expression programming. Construction and Building Materials, 130, 122-145. https://doi.org/10.1016/j.conbuildmat.2016.10.114
51. Asteris, P. G., Skentou, A. D., Bardhan, A., Samui, P., & Pilakoutas, K. (2021). Predicting concrete compressive strength using hybrid ensembling of surrogate machine learning models. Cement and Concrete Research, 145, 106449. https://doi.org/10.1016/j.cemconres.2021.106449
52. Neira, P., Bennun, L., Pradena, M., & Gomez, J. (2020). Predviđanje tlačne čvrstoće betona pomoću umjetnih neuronskih mreža. Građevinar, 72(07), 585-592. https://doi.org/10.14256/JCE.2438.2018
53. Mehmannavaz, T., Khalilikhorram, V., Sajjadi, S. M., & Samadi, M. (2014). Presenting an Appropriate Neural Network for Optimal Mix Design of Roller Compacted Concrete Dams. Research Journal of Applied Sciences, Engineering and Technology, 7(9), 1872-1877. https://doi.org/10.19026/rjaset.7.475
54. Khan, M. I. (2012). Predicting properties of high performance concrete containing composite cementitious materials using artificial neural networks. Automation in Construction, 22, 516-524. https://doi.org/10.1016/j.autcon.2011.11.011
55. Onyelowe, K. C., Iqbal, M., Jalal, F. E., Onyia, M. E., & Onuoha, I. C. (2021). Application of 3-algorithm ANN programming to predict the strength performance of hydrated-lime activated rice husk ash treated soil. Multiscale and Multidisciplinary Modeling, Experiments and Design, 4, 259-274. https://doi.org/10.1007/s41939-021-00093-7
56. Prasad, B. R., Eskandari, H., & Reddy, B. V. (2009). Prediction of compressive strength of SCC and HPC with high volume fly ash using ANN. Construction and Building Materials, 23(1), 117-128. https://doi.org/10.1016/j.conbuildmat.2008.01.014
57. Nguyen, T., Kashani, A., Ngo, T., & Bordas, S. (2019). Deep neural network with high-order neuron for the prediction of foamed concrete strength. Computer-Aided Civil and Infrastructure Engineering, 34(4), 316-332. https://doi.org/10.1111/mice.12422
58. Dias, W. P. S., & Pooliyadda, S. P. (2001). Neural networks for predicting properties of concretes with admixtures. Construction and Building Materials, 15(7), 371-379. https://doi.org/10.1016/S0950-0618(01)00006-X
59. Abunassar, N., Alas, M., & Ali, S. I. A. (2023). Prediction of compressive strength in self-compacting concrete containing fly ash and silica fume using ANN and SVM. Arabian Journal for Science and Engineering, 48(4), 5171-5184. https://doi.org/10.1007/s13369-022-07359-3
60. Yaman, M. A., Abd Elaty, M., & Taman, M. (2017). Predicting the ingredients of self compacting concrete using artificial neural network. Alexandria Engineering Journal, 56(4), 523-532. https://doi.org/10.1016/j.aej.2017.04.007
61. Belalia Douma, O., Boukhatem, B., Ghrici, M., & Tagnit-Hamou, A. (2017). Prediction of properties of self-compacting concrete containing fly ash using artificial neural network. Neural Computing and Applications, 28, 707-718. https://doi.org/10.1007/s00521-016-2368-7
62. Golafshani, E. M., Behnood, A., & Arashpour, M. (2020). Predicting the compressive strength of normal and High-Performance Concretes using ANN and ANFIS hybridized with Grey Wolf Optimizer. Construction and Building Materials, 232, 117266. https://doi.org/10.1016/j.conbuildmat.2019.117266
63. Demir, V., & Doğu, R. (2024). Prediction of elevation points using three different heuristic regression techniques. Turkish Journal of Engineering, 8(1), 56-64. https://doi.org/10.31127/tuje.1257847
64. Salami, B. A., Iqbal, M., Abdulraheem, A., Jalal, F. E., Alimi, W., Jamal, A., ... & Bardhan, A. (2022). Estimating compressive strength of lightweight foamed concrete using neural, genetic and ensemble machine learning approaches. Cement and Concrete Composites, 133, 104721. https://doi.org/10.1016/j.cemconcomp.2022.104721
65. Shariati, M., Armaghani, D. J., Khandelwal, M., Zhou, J., & Khorami, M. (2021). Assessment of longstanding effects of fly ash and silica fume on the compressive strength of concrete using extreme learning machine and artificial neural network. Journal of Advanced Engineering and Computation, 5(1), 50-74. http://dx.doi.org/10.25073/jaec.202151.308
66. Mogaraju, J. K. (2024). Machine learning empowered prediction of geolocation using groundwater quality variables over YSR district of India. Turkish Journal of Engineering, 8(1), 31-45. https://doi.org/10.31127/tuje.1223779
67. Acı, M., Acı, Ç. İ., & Avcı, M. (2018). Performance comparison of ANFIS, ANN, SVR, CART AND MLR techniques for geometry optimization of carbon nanotubes using CASTEP. Turkish Journal of Engineering, 2(3), 119-124. https://doi.org/10.31127/tuje.408976
68. Kiani, B., Gandomi, A. H., Sajedi, S., & Liang, R. Y. (2016). New formulation of compressive strength of preformed-foam cellular concrete: an evolutionary approach. Journal of Materials in Civil Engineering, 28(10), 04016092. https://doi.org/10.1061/(ASCE)MT.1943-5533.0001602

69. Nehdi, M., Djebbar, Y., & Khan, A. J. M. J. (2001). Neural network model for preformed-foam cellular concrete. Materials Journal, 98(5), 402-409.
70. Othman, M. M. (2023). Modeling of daily groundwater level using deep learning neural networks. Turkish Journal of Engineering, 7(4), 331-337. https://doi.org/10.31127/tuje.1169908
71. Tiza, M. T., Ogunleye, E., Jiya, V. H., Onuzulike, C., Akande, E. O., & Terlumun, S. (2023). Integrating Sustainability into Civil Engineering and the Construction Industry. Journal of Cement Based Composites, 4(1), 1-11. https://doi.org/10.36937/cebacom.2023.5756
72. Algaifi, H. A., Bakar, S. A., Alyousef, R., Sam, A. R. M., Alqarni, A. S., Ibrahim, M., ... & Salami, B. A. (2021). Machine learning and RSM models for prediction of compressive strength of smart bio-concrete. Smart Structural System, 28, 535-551. https://doi.org/10.12989/sss.2021.28.4.535
73. Chou, J. S., & Pham, A. D. (2013). Enhanced artificial intelligence for ensemble approach to predicting high performance concrete compressive strength. Construction and Building Materials, 49, 554-563. https://doi.org/10.1016/j.conbuildmat.2013.08.078
74. Imam, A., Salami, B. A., & Oyehan, T. A. (2021). Predicting the compressive strength of a quaternary blend concrete using Bayesian regularized neural network. Journal of Structural Integrity and Maintenance, 6(4), 237-246. https://doi.org/10.1080/24705314.2021.1892572
75. Ly, H. B., Nguyen, M. H., & Pham, B. T. (2021). Metaheuristic optimization of Levenberg–Marquardt-based artificial neural network using particle swarm optimization for prediction of foamed concrete compressive strength. Neural Computing and Applications, 33(24), 17331-17351. https://doi.org/10.1007/s00521-021-06321-y
76. Shariati, M., Mafipour, M. S., Mehrabi, P., Bahadori, A., Zandi, Y., Salih, M. N., ... & Poi-Ngian, S. (2019). Application of a hybrid artificial neural network-particle swarm optimization (ANN-PSO) model in behavior prediction of channel shear connectors embedded in normal and high-strength concrete. Applied Sciences, 9(24), 5534. https://doi.org/10.3390/app9245534
77. Shariati, M., Mafipour, M. S., Mehrabi, P., Shariati, A., Toghroli, A., Trung, N. T., & Salih, M. N. (2021). A novel approach to predict shear strength of tilted angle connectors using artificial intelligence techniques. Engineering with Computers, 37, 2089-2109. https://doi.org/10.1007/s00366-019-00930-x
78. Shariati, M., Mafipour, M. S., Mehrabi, P., Ahmadi, M., Wakil, K., Trung, N. T., & Toghroli, A. (2020). Prediction of concrete strength in presence of furnace slag and fly ash using Hybrid ANN-GA (Artificial Neural Network-Genetic Algorithm). Smart Structures and Systems, An International Journal, 25(2), 183-195. https://doi.org/10.12989/sss.2020.25.2.183
79. Pham, A. D., Ngo, N. T., Nguyen, Q. T., & Truong, N. S. (2020). Hybrid machine learning for predicting strength of sustainable concrete. Soft Computing, 24(19), 14965-14980. https://doi.org/10.1007/s00500-020-04848-1
80. Babajanzadeh, M., & Azizifar, V. (2018). Compressive strength prediction of self-compacting concrete incorporating silica fume using artificial intelligence methods. Civil Engineering Journal, 4(7), 1542-1552. http://dx.doi.org/10.28991/cej-0309193
81. Özcan, F., Atiş, C. D., 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. Advances in Engineering Software, 40(9), 856-863. https://doi.org/10.1016/j.advengsoft.2009.01.005