

REVIEW

Open Access



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

Sesugh Terlumun^{1*} , M. E. Onyia² and F. O. Okafor²

Abstract

Concrete is one of the most common construction materials used all over the world. Estimating the strength properties of concrete traditionally demands extensive laboratory experimentation. However, researchers have increasingly turned to predictive models to streamline this process. This review focuses on predicting the compressive strength of self-compacting concrete using artificial intelligence (AI) techniques. Self-compacting concrete represents an advanced construction material particularly suited for scenarios where traditional vibrational methods face limitations due to intricate formwork or reinforcement complexities. This review evaluates various AI techniques through a comparative performance analysis. The findings highlight that employing Deep Neural Network models with multiple hidden layers significantly enhances predictive accuracy. Specifically, artificial neural network (ANN) models exhibit robustness, consistently achieving R^2 values exceeding 0.7 across reviewed studies, thereby demonstrating their efficacy in predicting concrete compressive strength. The integration of ANN models is recommended for formulating various civil engineering properties requiring predictive capabilities. Notably, the adoption of AI models reduces both time and resource expenditures by obviating the need for extensive experimental testing, which can otherwise delay construction activities.

Keywords Prediction, Compression strength, Self-compacting concrete, Artificial intelligence techniques, Review

1 Introduction

Historically, concrete has been the most prevalent building material globally (Rob & Noel, 2003). Its extensive application in construction projects around the world has cemented its status as a reliable and essential component of human infrastructure. Since the era of the Roman Empire, concrete has played a crucial role in the development and enhancement of various cultures. The "magic" of concrete lies not only in its transformative properties from a fluid to a solid state but also in its affordability,

wide availability, malleability, plasticity, adaptability, high compressive strength, stiffness, and durability (Falliano et al., 2018; Rahmat & Mousavi, 2012; Rob & Noel, 2003; Seung-Chang 2003). Researchers continue to explore this specialized field of concrete engineering with enthusiasm, revealing promising prospects for innovative materials such as reactive powder concrete. The compressive strength of concrete typically ranges from 200 to 800 MPa (Falliano et al., 2018; Rahmat & Mousavi, 2012; Seung-Chang 2003), underscoring its versatility and potential for future advancements.

It is standard practice to obtain a sufficiently large sample of a concrete mixture to infer conclusions about the entire batch. According to references (Gagg, 2014; Rana et al., 2022; Shi et al., 2015), concrete testing at day 7 is crucial for assessing early-age strength, although concrete reaches its full strength at 28 days, necessitating a similar duration

*Correspondence:

Sesugh Terlumun
terlumunesugh8@gmail.com

¹ Department of Civil and Environmental Engineering, Air Force Institute of Technology, Kaduna, Nigeria

² Department of Civil Engineering, University of Nigeria Nsukka, Nsukka, Nigeria



© The Author(s) 2024. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

for sample examination. The 28-day test results serve as a basis for predicting the compressive strength of concrete over time. These procedures align with established design standards, yet these design standards may prove inadequate when dealing with concrete compositions that extend beyond conventional cement, aggregate, and water. The complexity of calculating the compressive strength of a concrete mix increases when alternative components such as pozzolans and admixtures are introduced, especially in high-strength concrete formulations (Ashish & Verma, 2019; Danish & Ganesh, 2021; EFNARC, 2002; Esmailkhanian et al., 2014; Surya et al., 2020). Factors like the influence of admixtures and temperature can significantly impact compressive strength outcomes. In such cases, empirical relationships suggested by standards often fail to capture the full complexity, necessitating the development of reliable models to promptly assess concrete strength at various ages, crucial for meeting urgent project deadlines that cannot wait for traditional 28-day test results. The time saved from waiting for test outcomes translates into tangible economic benefits, given the productivity-driven nature of construction processes. Therefore, exploring alternative approaches becomes imperative. There remains considerable scope for debate and exploration regarding the efficacy of these non-traditional methodologies.

There has been considerable interest in the application of artificial neural networks (ANNs) for predicting concrete compressive strength in recent years (Bayer et al., 2019; Kandiri et al., 2020). Various studies have explored hybrid models such as fuzzy logic (FL), genetic programming (GP), and ANN to forecast the influence of ground granulated blast furnace slag on concrete strength over time (Golafshani et al., 2015). The cascade correlation type of ANN has enabled rapid learning albeit with moderate accuracy, effectively capturing the nonlinear patterns inherent in concrete properties (Gesoglu et al., 2010; Öztaş et al., 2006; Siddique et al., 2011). This capability is particularly advantageous for problems like compressive strength measurement, offering savings in time and cost. The combination of artificial neural networks with meta-heuristic algorithms has shown promising results in addressing complex challenges in structural engineering. For instance, hybrid multilayer perceptron (HMP) networks have been beneficial for determining concrete compressive strength in scenarios such as deep beams connected

to shear walls (Behnood & Golafshani, 2018; Behnood et al., 2017; Chakravarthy et al., 2023; Kovačević, M., Lozančić, S., Nyarko, E.K., Hadzima-nyarko & M., 2021; Siddique et al., 2011). Artificial ANNs and evolutionary search methods like genetic algorithms (GAs) are integrated in the context of Evolutionary Artificial Neural Networks (EANNs), as elucidated in references (Behnood et al., 2017; Chakravarthy et al., 2023). Reference (Kovačević, M., Lozančić, S., Nyarko, E.K., Hadzima-nyarko & M., 2021) lays out the framework for EANNs. These networks have proven highly efficacious in detecting structural defects and predicting compressive strength in concrete (Aiyer et al., 2014; Golafshani et al., 2020a; Kovačević, M., Lozančić, S., Nyarko, E.K., Hadzima-nyarko & M., 2021). Consequently, EANNs are being investigated as a potentially cost-effective alternative to both expensive mathematical models and destructive testing methods for accurately estimating concrete compressive strength.

2 Methodology

This review focuses on exploring machine learning models and their application to predicting the compressive strength of self-compacting concrete. To achieve this objective, a comprehensive literature review was conducted utilizing various scholarly databases and journals such as Scholarly Journals, Elsevier, Springer, SCOPUS, the Web of Science, Turkish Journal of Engineering, IEEE, and Science Direct. These sources were selected based on their relevance and contribution to the research topic. A total of 81 relevant sources were identified and analyzed. Figure 1 illustrates the research methodology adopted in this study, outlining the systematic approach followed in sourcing and reviewing literature.

A total of 80 articles sourced from academic journals constituted a significant portion of the scholarly references analyzed in this review. In addition to these articles, the EFNARC guidelines and specifications for self-compacting concrete were also utilized, bringing the total number of academic sources to 81. This review employs modern construction technology to systematically evaluate the compressive strength of self-compacting concrete through the application of machine learning models.

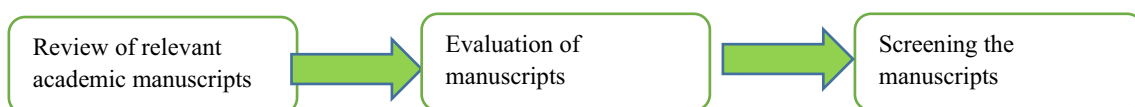


Fig. 1 Flow chart of the approach used in obtaining relevant articles for the research

3 Results and discussions

3.1 Predicting concrete compressive and flexural strength

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

3.1.1 Empirical methods

In an effort to faithfully represent experimental procedures within a specified context, empirical methodologies have been developed. The significance of these methodologies lies in their ability to quantify the contribution of each component to the final outcome. The established relationships must align with experimental data, necessitating comprehensive empirical analysis. In the context of forecasting compressive strength, many empirical models endeavor to correlate strength with the water-cement ratio, despite its typically adverse impact on predicted strength. Alternatively, some models incorporate existing compressive strength data and employ empirically derived coefficients to establish a link between available and required information (Danish & Ganesh, 2021; Sonebi 2004).

3.1.2 Computational modelling

Complex equations in thermodynamics serve as the fundamental basis for finite element analysis, a computational modeling technique widely employed in engineering and materials science. This approach crucially depends on accurate representations of concrete microstructures. To effectively simulate processes such as hydration across various particle sizes within cement using pixel-based computer simulations, it is imperative to randomly distribute cement within a unit cell space. Additionally, empirical encoding of these processes can be pursued following adjustments based on experimental data (Danish & Ganesh, 2021; Jovic et al., 2019; Sonebi 2004).

3.1.3 Mechanical modeling

In mechanical models, parameters are commonly conceptualized using a spring-and-dashpot structure. In this analogy, the cement matrix corresponds to the spring, while the temporal factor (age at the time of testing) represents the dashpot. This theoretical framework is applied to predict the compression strength of concrete. However, when this model is tested against experimental data, its predictive accuracy declines. Specifically, the dashpot component notably influences compressive strength, particularly at early ages (Danish & Ganesh, 2021; Jovic et al., 2019; Sonebi 2004).

3.1.4 Statistical methods

To ensure precise delineation of interrelationships among variables, statistical methodologies leverage empirical

data and employ mathematical formulations. 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 (Danish & Ganesh, 2021; Günal & Mehdi, 2023; Jovic et al., 2019; Sonebi 2004).

3.1.5 Regression analysis

Regression analysis is widely acknowledged as an essential component of robust statistical modeling strategies. Despite its utility in computing coefficients that quantify efficiency gains (Behnood & Golafshani, 2020; Madani et al., 2020; Zhou et al., 2016), the approach is often intricate and challenging to follow (Behnood & Golafshani, 2020; Madani et al., 2020; Zhou et al., 2016). Notwithstanding these complexities, establishing confidence intervals for predictions necessitates rigorous mathematical underpinnings. Correlation analysis further elucidates how key variables influence the final outcome. 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 the regression model fails to interpret any variance in Y , whereas a value of 1 indicates that all points are on the regression line. Moreover, some findings from slump tests and concrete density were presented to complement the mix proportions of constituent materials, aiding in the estimation of compressive strength in high-performance concrete (Behnood & Golafshani, 2020; Madani et al., 2020; Zhou et al., 2016).

The review aimed to enhance the accuracy of test results across different age groups to better reflect the progression from weakness to strength. The final multi-variable power equation was refined by incorporating additional independent variables, achieving a correlation coefficient of 99.99%. Furthermore, significant correlations between predicted compressive strength and experimental data were identified, as documented in the literature (Chiew et al., 2017). Standardization of model inputs using a fixed matrix formula was implemented to optimize the formulation of concrete mixtures. Regression methods have traditionally been favored due to their simplicity and efficiency in modeling, particularly in scenarios where non-linear relationships between reactants and products are minimal, such as in concrete applications. This effectiveness is highlighted in previous studies (Sergio & Mauro, 1997; Seung-Chang, 2003), which

compared regression methods with artificial neural networks and concluded that the former offer superior predictive capabilities.

3.1.6 Artificial intelligence

According to Webster's New World College Dictionary, artificial intelligence (AI) is defined as "the ability of machines or programs to function in ways that simulate human intelligence in tasks such as reasoning and learning." This definition prompts an inquiry into the types of problems that might necessitate a computational approach that emulates human cognitive processes. Examples of such problems include knowledge-based inference with partial or unclear data, various forms of perception and learning, as well as tasks involving control, prediction, classification, and optimization (Erzin, 2007; Syed et al., 2023). These challenges provide contexts in civil engineering where AI techniques can be employed to replicate phenomena whose underlying processes are not fully understood. Neural networks and genetic algorithms represent two distinct yet conceptually similar AI techniques. Neural networks are designed to mimic the brain's cognitive processes, whereas genetic algorithms are inspired by the principles of natural selection and 'mutation' to enhance performance. Additionally, Adaptive Network-based Fuzzy Inference Systems (ANFIS) and fuzzy systems are two further examples of modern artificial intelligence. ANFIS integrates fuzzy logic and neural networks to leverage the strengths of both approaches, utilizing linguistic interpretations of variables and adaptive learning processes to generate effective models (Erzin, 2007; Syed et al., 2023). The effectiveness of AI-based methods lies in their ability to learn from data and produce relevant models for a broad array of applications. However, achieving results that approximate real-world conditions often necessitates a data set that is both extensive and high-quality.

3.2 Fuzzy logic

Lotfi Zadeh is credited with pioneering the field of fuzzy logic (FL), as documented in seminal works (Duan et al., 2013; Sergio & Mauro, 1997; Silva et al., 2021; Syed et al., 2023). His contributions, including the development of fuzzy inference systems such as Mamdani, Takagi, and Sugeno, fundamentally transformed methods for knowledge representation.

Unlike the binary nature of traditional Boolean logic, which strictly adheres to values of "completely false" (0) or "completely true" (1), fuzzy logic introduces a nuanced approach based on degrees of membership (Duan et al., 2013; Erzin, 2007; Sergio & Mauro, 1997; Silva et al., 2021; Syed et al., 2023). In contrast to the rigid boundaries of Boolean logic, where information is represented in

black-and-white terms, fuzzy logic acknowledges shades of gray and accommodates the inherent uncertainties in real-world data. An individual may, for instance, possess data on the heights of 10 persons in a dataset, categorized into groups: 'tall' individuals ranging from 1.75 m to 2.20 m and 'short' individuals ranging from 1.50 m to 1.74 m. In such scenarios, the use of crisp sets may appear inadequate. Scholars (Duan et al., 2013; Erzin, 2007; Sergio & Mauro, 1997; Syed et al., 2023) have criticized this dichotomous approach as "overly simplistic" and often "out of touch with reality." Fuzzy sets offer a more precise and robust method to represent such data. According to Sergio and Mauro (1997); Syed et al., 2023; Erzin, 2007; Duan et al., 2013), a fuzzy set is defined as a class of objects with degrees of membership that form a continuum. Here, a membership function assigns each item in the set a score between one and zero, reflecting its degree of membership in the set.

Modelling non-linear systems and designing sophisticated controllers are two pivotal applications of fuzzy logic control (FLC), a robust mathematical framework. In concrete, each constituent component serves a distinct purpose, individually and collectively. This interplay leads to nonlinearity in the relationship between constituent quantities and resulting compressive strength. FLC mitigates this nonlinearity by its application in scenarios where the system's complexity precludes the use of traditional modeling approaches (Duan et al., 2013; Erzin, 2007; Sergio & Mauro, 1997; Syed et al., 2023).

4 Neural Network

The transformative evolution of neural networks in problem-solving approaches is currently underway. Since the inception of artificial intelligence in the 1950s, neural networks have spearheaded endeavors to augment robots' functionalities beyond physical labor to intellectual tasks. The term "artificial neural network" encompasses various conceptualizations, analogous to biological brain networks. Here, axons represent outputs and synapses signify weights. In artificial neural networks, the ubiquitous neuron is also referred to as a "processing element." According to Silva et al. (2021), Duan et al. (2013), Alade et al. (2018), and Iqbal et al. (2020), artificial neural networks are succinctly 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 exemplifies a simple neural network with a direct input-to-output mapping. In contrast, more intricate neural networks feature multiple layers and employ diverse activation functions. Classification of artificial neural networks sheds light on diverse structures and functionalities.

Damage detection, structural systems identification, material behavior modelling, structural settlement analysis, control and optimization strategies, groundwater monitoring, and determination of concrete mix proportions are just some of the many areas where ANNs have been put to use (Saridemir, 2010). In a study referenced by Gandomi and Roke (2015), ANNs were explored within prediction models alongside fuzzy logic to forecast the compressive strength of self-compacting concrete. Their ANN model, featuring a hidden layer comprising 6 neurons, underwent 500 iterations. Comparable success rates were achieved, although discrepancies in measurement accuracy persisted. Notably, the ANN model exhibited superior performance with an R^2 value of 0.9767 compared to the fuzzy logic approach. Subsequent refinement involving an ANN configuration with eleven neurons in a single hidden layer, and later with a two-hidden-layer setup (nine neurons in the first layer and eight in the second), resulted in further accuracy improvements, achieving the lowest absolute percentage error ($=0.000515$) (Khan et al., 2021). Reference (Faradonbeh et al., 2018) also considered the advantages of a multi-layer architecture. Material considerations are integral to this model, evident in the utilization of chemical analysis data for fly ash, gradient, and sand compositions. The model achieved R^2 value of 0.9557 for compressive strength and 0.9119 for flexural tensile strength, indicating that the prediction model performed well. The influence of water-to-binder ratio on compressive strength was also analyzed, revealing a decrease as the ratio increased, thereby influencing the compressive strength forecasting model (Faradonbeh et al., 2018; Khandelwal et al., 2017). Their model demonstrated high accuracy with an R^2 of 0.9944 in reproducing experimental outcomes and 0.9767 for predicting testing samples. Comparative analyses of concrete strength prediction methodologies, including artificial neural network (ANN) approaches (Ferreira, 2002, 2006; Gholampour et al., 2017), highlight the ANN's superior performance over multiple linear regression models, especially for low and medium strength concretes.

4.1 Genetic programming

In order to evaluate a computer's performance executing a specific task, genetic programmers employ a framework called Gene Expression Programming (GEP), which utilizes a set of instructions and a fitness evaluation process. GEP operates akin to a subtype of genetic algorithms (GA), where each node represents a segment of code executed on a computer. This methodology aims to enhance program efficiency by strategically relocating its components to optimal physical positions based on predefined conditions that must be met. Previous studies have

identified three key genetic manipulations (Faradonbeh et al., 2018; Ferreira, 2002, 2006; Khandelwal et al., 2017).

Some critical adjustment factors are elucidated in articles (Faradonbeh et al., 2018; Ferreira, 2002, 2006; Khandelwal et al., 2017). The genetic programming (GP) building process begins with an initial population size of 49 without cement/FA substitution and 27 with a cement/FA substitution rate set at 0.15. The curing period serves to further categorize each dataset within the GP model, mirroring its role in the ANN model. Specifically targeting a 28-day compressive strength outcome, the GP model incorporates four primary input parameters: water, cement, coarse aggregate, and fine aggregate.

Reference (Ferreira, 2002, 2006; Khandelwal et al., 2017) presented regression equations for predicting the in situ concrete compressive strength. Their models were derived from comprehensive datasets encompassing ready-mixed concrete mixture compositions and corresponding on-site compressive strength assessments. The study utilized 1,442 compressive strength tests across 68 distinct concrete mix designs, characterized by specified compressive strengths ranging from 18 to 27 MPa, water to cement ratios varying between 0.39 and 0.62, and aggregate maximum sizes spanning 25 to 100 mm. Additionally, references (Ferreira, 2002, 2006; Gholampour et al., 2017) tested a proposed model for predicting the compressive strength of concrete using in situ data.

4.2 Parameters in machine learning models

Hyperparameters in machine learning are external settings that guide decision-making and influence the learning process of algorithms. Machine learning engineers must configure these hyperparameters before training the algorithm (Asteris et al., 2021; Mehmannaavaz et al., 2014; Neira et al., 2020). Hyperparameters encompass various elements such as the learning rate, the number of clusters in a clustering algorithm, and the number of branches in a regression tree. During the training process, the algorithm adjusts its internal parameters—such as weights and biases—based on the hyperparameters and the training data. These internal parameters are refined through training to achieve optimal model performance. Ideally, the final model parameters should fit the data set precisely without underfitting or overfitting (Asteris et al., 2021; Mehmannaavaz et al., 2014; Neira et al., 2020).

A hyperparameter in the context of machine learning refers to a parameter that governs the learning process itself, distinct from the parameters whose values are determined through training, typically node weights. These hyperparameters fall into two main 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 hyperparameters (Asteris et al., 2021; Mehmannaavaz et al., 2014; Neira et al., 2020).

Hyperparameters of an algorithm include learning rate, batch size, and mini-batch size. A mini-batch size denotes 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 hyperparameters. The training algorithm extracts parameters from the data and applies them to these hyperparameters. The selection of a model's hyperparameters can affect how long it takes to train and test. Typically, a hyperparameter is of an integer or continuous form, leading to challenges in mixed-type optimization. Some hyperparameters may only exist contingent upon the values of others, for example, in a neural network, the number of layers can affect the size of each hidden layer (Asteris et al., 2021; Mehmannaavaz et al., 2014; Neira et al., 2020).

4.3 Untrainable parameters

Hyperparameters play a crucial role in enhancing a model's capacity, yet their aggressive tuning may lead to suboptimal outcomes by driving the loss function towards undesired minima. This phenomenon, known as overfitting, occurs when the model begins to excessively capture noise in the data rather than faithfully representing its underlying structure. Consequently, the learning process of these hyperparameters from the training set may be hindered. For example, consider the degree of a polynomial equation used in a regression model. If this degree were treated as a trainable parameter, the model could potentially adjust it to perfectly fit the training data, thereby minimizing training error. However, such an approach often sacrifices generalization performance, as the model becomes overly specialized to the training dataset and fails to generalize well to unseen data.

4.4 Tuning in ML

Only a few hyperparameters account for the performance variation of machine learning models. The extent to which adjusting an algorithm, hyperparameter, or combination thereof enhances performance is referred to as its tuning capability. Among the various hyperparameters, the most critical for machine learning models are the learning rate and network. In contrast, empirical evidence suggests that batch size and momentum exert negligible influence on model performance (Asteris et al., 2021; Mehmannaavaz et al., 2014; Neira et al.,

2020). Research has shown that mini-batch sizes ranging from 2 to 32 generally yield optimal results, despite occasional arguments in favor of larger mini-batch sizes in the thousands.

4.5 Robustness of ML models

Learning processes inherently exhibit stochasticity, implying that performance evaluations based on empirical hyperparameters may not faithfully represent 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, evaluating these algorithms with a limited number of random seeds inadequately captures their performance dynamics (Asteris et al., 2021; Mehmannaavaz et al., 2014; Neira et al., 2020). Certain reinforcement learning techniques like Deep Deterministic Policy Gradient (DDPG) are more responsive to selections of hyper parameters than others.

4.6 Optimization

Through the use of hyperparameter optimization, a tuple of hyperparameters is found to produce an optimal model on test data that minimizes a predetermined loss function. The objective function computes and returns the associated loss upon receiving the set of hyperparameters.

4.7 Reproduction in ML models

In addition to fine-tuning hyperparameters, machine learning entails reproducibility checks, parameter and result organization, and storage. Without robust infrastructure, research code often undergoes frequent modifications, risking essential aspects such as reproducibility and record-keeping. Online collaboration platforms for machine learning facilitate seamless exchange, organization, and communication of experiments, data, and algorithms among scientists. Reproducing deep learning models, in particular, presents notable challenges (Khan, 2012).

4.8 Creating machine learning models

Machine learning models are developed using various approaches that leverage labeled, unlabeled, or a combination of both types of data (Khan, 2012; Onyelowe et al., 2021). There are four main machine learning algorithms available:

1. **Supervised learning:** This method involves training algorithms on labeled data, where each data point is associated with a known label. The labels serve as a guide for the algorithm to learn to classify new data accurately. Supervised learning ensures that the resulting model aligns with the intended classification objectives set by researchers.
2. **Unsupervised learning:** Algorithms under unsupervised learning operate on unlabeled data. Without predefined labels, these algorithms autonomously identify patterns and structures within the data. This approach is particularly useful for researchers seeking to uncover hidden patterns and structures in datasets.
3. **Semi-supervised learning:** This technique trains an algorithm by combining labeled and unlabeled data. Initially, the algorithm is trained on a small set of labeled data to establish a foundational understanding. Subsequently, it utilizes a larger pool of unlabeled data to further refine its learning and improve model performance.
4. **Reinforcement learning:** In reinforcement learning, algorithms learn through interaction with an environment where they receive feedback in the form of rewards or penalties. The goal is to maximize cumulative rewards over time by selecting actions that lead to positive outcomes while avoiding negative ones. Reinforcement learning can guide unsupervised machine learning by providing incentives for discovering beneficial patterns.

4.9 Types of machine learning models

In the realm of machine learning, problems can generally be categorized into two main types: classification and prediction. These categories represent fundamental challenges addressed by machine learning algorithms. Depending on the nature of the problem, algorithms are tailored to develop models that are either classifiers or predictors (Prasad et al., 2019). Notably, some algorithms can be trained to produce models suitable for both regression (used for predictive modeling) and classification tasks. A variety of well-established algorithms are employed for constructing regression and classification models, as listed below:

Classification Models:

1. Support Vector Machines.
2. Random Forests.
3. Decision Trees.
4. Logistic Regression.
5. Naive Bayes.
6. K-Nearest Neighbor (KNN).

Regression Models:

1. K-Nearest Neighbor (KNN) Regression.
2. Decision Trees.
3. Random Forests.
4. Neural Network Regression.
5. Linear Regression.

4.10 Overfitting in machine learning (ML)

In machine learning (ML), overfitting denotes a condition where a model exhibits diminished performance on new, unseen data due to excessive alignment with the training data from which it was derived. The primary objective is to enhance the model's ability to generalize to novel data instances, thereby ensuring its applicability beyond the specific dataset used for training (Dias & Pooliyadda, 2001; Nguyen et al., 2019). Overfitting manifests commonly in tasks such as image recognition and natural language processing. Upon mitigating overfitting, these models can achieve improved precision in predicting new data and provide dependable forecasts for practical applications. Overfitting arises due to several factors: insufficient data, inclusion of extraneous information in the dataset, prolonged training on a specific dataset, and excessive model complexity. In the context of ML, overfitting describes a scenario where the model becomes overly intricate and closely adheres to idiosyncrasies within the training data. Mitigating overfitting aims primarily at enhancing the model's generalization capability to handle unseen data effectively. Techniques for identifying and addressing overfitting include monitoring all loss metrics, scrutinizing the learning curve, integrating regularization techniques, conducting cross-validation, and visually inspecting predictions to verify alignment with the training data.

Various strategies are available to prevent overfitting in machine learning, including dropout, feature selection, early stopping, cross-validation, regularization (such as L1 and L2), and data augmentation. When a machine learning model becomes excessively attuned to the nuances of the training data, impeding its ability to generalize effectively to new data instances, it is said to suffer from overfitting. This phenomenon is particularly noticeable in neural networks, where the model assigns undue significance to incidental details in the training dataset. Addressing overfitting is crucial for ensuring the model's accuracy in making predictions about new data, distinguishing between essential patterns and irrelevant noise.

4.11 Causes of overfitting

The training dataset utilized by the model exhibits considerable noise and imperfections. Moreover, the

dataset's size is insufficient; utilizing only a fraction of the available data leads to an incomplete representation of the entire dataset. Consequently, when evaluating decision trees for overfitting through validation metrics such as accuracy and loss, one can discern instances of overfitting. Overfitting causes these metrics to typically ascend until reaching a peak, after which they commence a decline or reach a plateau. Furthermore, as the model strives to achieve an optimal fit, these metrics stabilize or decrease. Addressing this issue necessitates striking a delicate balance between model complexity and the volume of available training data.

4.12 Identifying overfitting in ML models

Overfitting can be identified through various indicators, as outlined below:

1. Monitoring all losses is crucial; notable is when the validation loss increases while the training loss decreases.
2. Close attention should be paid to the machine learning curve, as any noticeable divergence between the training and validation curves suggests overfitting. Incorporating a regularization term into the loss function is essential to mitigate overfitting.
3. Visual inspection of the model's predictions is indispensable for assessing whether the model excessively conforms to the training data, which serves as a direct indicator of overfitting.

4.13 Preventing ML from overfitting

Several methods are available to mitigate overfitting in machine learning models, as outlined below:

1. Regularization techniques such as L1 and L2 are employed by augmenting the loss function with penalty terms. These techniques discourage the model from excessively tailoring itself to the training data, thereby mitigating overfitting.
2. Cross-validation serves as a pivotal method to prevent overfitting. By employing techniques such as

early stopping, training halts just before the onset of overfitting, thereby ensuring that the model generalizes well to unseen data.

3. Another effective approach is data augmentation, which enriches the training dataset by generating synthetic data from existing samples. By exposing the model to a diverse array of data instances, data augmentation enhances its robustness against overfitting.
4. Feature selection reduces model complexity by excluding irrelevant or noisy features, thereby preventing overfitting to noisy data.
5. The dropout technique forces the model to learn more robust and generalizable representations, thus preventing data overfitting.

Implementing these techniques collectively diminishes the overfitting problem, resulting in the development of more precise and reliable machine learning models.

4.14 Fuzzy logic approach

The four main components of each fuzzy system are Fuzzy rule base, Fuzzification, Defuzzification and Inference Engine (Golafshani et al., 2020b). Figure 2 presents a general fuzzy logic model architecture.

Fuzzification is the first step in which fuzzy inputs are assessed and transformed into one or more fuzzy sets. Two primary types of fuzzification commonly employed are Gaussian and trapezoidal. Within the framework of fuzzy logic, any element may belong to multiple subsets of the universal set. Fundamentally, fuzzy rules are structured as linguistic IF–THEN statements, employing the general format "IF A THEN B," where A and B are schemes incorporating linguistic variables. Here, A denotes the condition (premise), while B signifies the consequence (rule significance). In order to account for ambiguity and inaccuracy, linguistic variables and fuzzy IF–THEN rules are used. Depending on how unique the challenges are, two different types of rule techniques are used. In a fuzzy inference engine, the fuzzy rule base is incorporated to analyze fuzzy outputs (Ben, Al-Asri, Zaher, Hafidi, Burtschell, 2022; Golafshani et al., 2020b). Subsequently, the Defuzzification process converts these

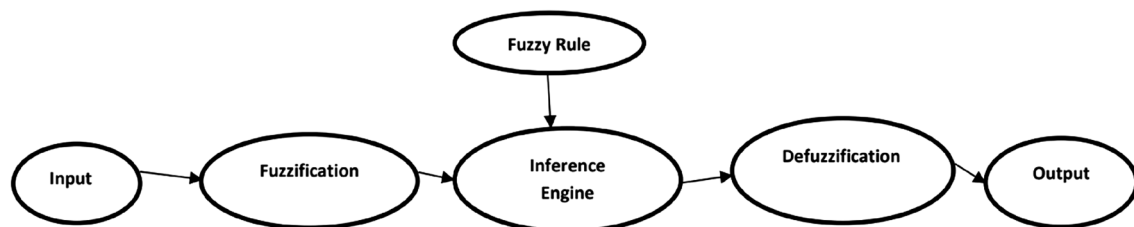


Fig. 2 Fuzzy Logic Architecture

outputs from the fuzzy inference engine into precise numerical values and crisp outputs.

4.15 Evaluating the performance of the GP model:

In assessing the adequacy of a model, it is recommended that the ratio of the dataset size to the total number of input features be at least 3, with a preference for a ratio of 5. Validation of the GEP model involves employing various statistical computations across training, testing, and validation datasets. Evaluation focuses on determining the effectiveness of model training and establishing significant associations between the model and experimental data, while minimizing errors. Key parameters such as RMSE, MAE, and RSE are computed during the testing phase (Ben, Al-Asri, Zaher, Hafidi, Burtschell, 2022; Golafshani et al., 2020b). Additionally, employing statistical techniques allows for external validation of the GEP model. A crucial criterion is that one of the regression lines (k or k') with a slope passing through the origin should closely approximate 1. The same dataset is utilized in the linear regression model of this review, which calculates the SCC represented as f_c . 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 are determined

using test runs or experimental results. contingent upon the population size (number of chromosomes), which dictates the duration of program execution. Population size levels are chosen based on the quantity and complexity of the prediction model. In this study, the algorithm used two variables—head size and number of genes—to define model architecture (Babatunde et al., 2022; Mogaraju, 2023; Shariati et al., 2021). The head size, representing the size of the model's "head," determines the complexity of each term within the model. Meanwhile, the number of sub-ETs (elementary trees), basic data structures comprising the model, is determined by the number of genes. Five alternative head sizes—8, 9, 10, 12, and 14—are considered in this review, each associated with either three or four genes (Mogaraju, 2023; Nguyen et al., 2019). The GEP algorithm is used to derive the precise parameters for each model. The flow chart for the GP is shown Fig. 3.

4.16 Model evaluation criteria

One commonly used performance indicator is the correlation coefficient (R). However, 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. Consequently, this study

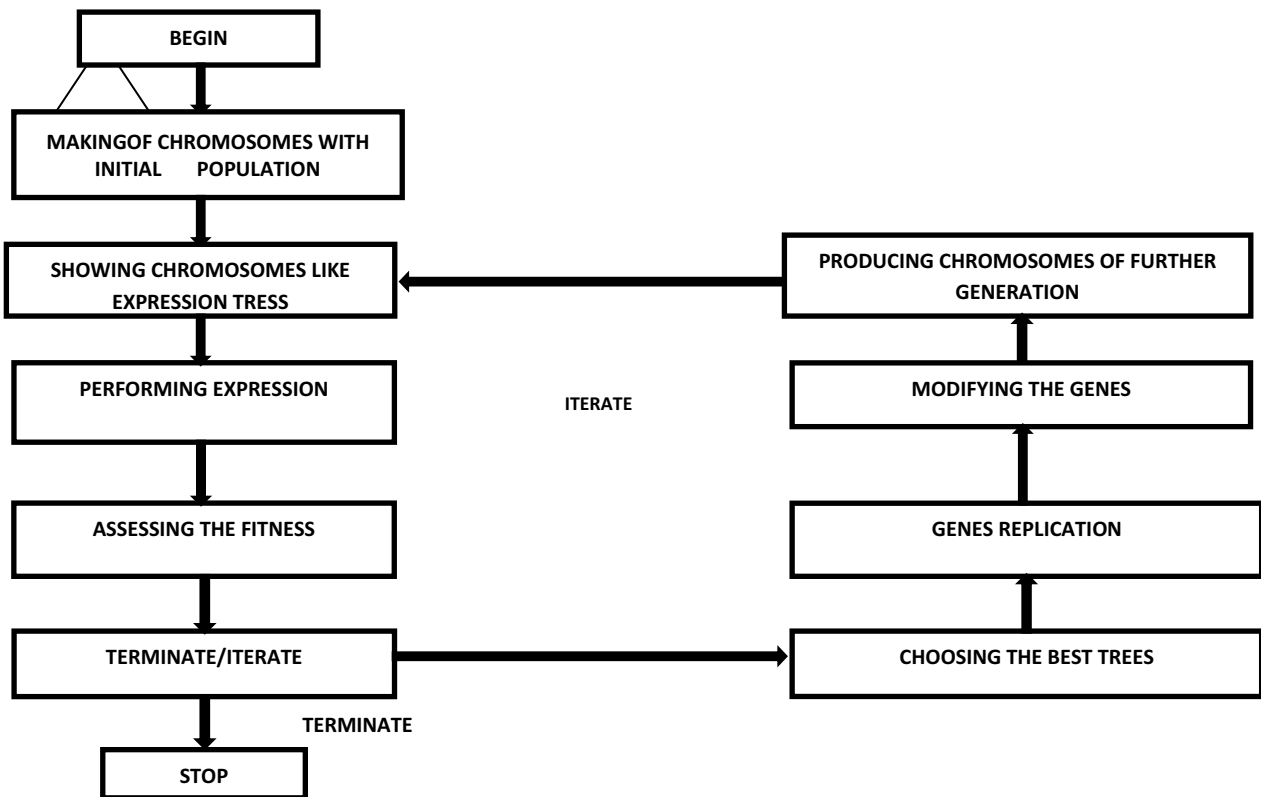


Fig. 3 Genetic Algorithm Flow Chart Diagram

also assesses additional metrics including mean absolute error (MAE), relative squared error (RSE), and relative root mean square error (RRMSE) (Aci et al., 2018; Kiani et al., 2016; Nehdi et al., 2001; Othman, 2023; Tiza et al., 2023). To evaluate the model's performance in regard to both the RRMSE and R, Gandomi and Roke proposed a performance index (Golafshani et al., 2020b). Equations 1–7 provide the mathematical formulas for these error functions.

$$F_c = f(\alpha + \beta_1 Y_1 + \beta_2 Y_2 + \beta_3 Y_3 + \beta_4 Y_4 \dots + \beta_n Y_n) + \epsilon_1 \tag{1}$$

where F_c is the compressive strength, β_1 to β_n denote the regression coefficients, α is the regression constant, ϵ_1 is the error term.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (e_i - m_i)^2}{n}} \tag{2}$$

$$MAE = \sqrt{\frac{\sum_{i=1}^n |e_i - m_i|^2}{n}} \tag{3}$$

$$RSE = \sqrt{\frac{\sum_{i=1}^n (m_i - e_i)^2}{\sum_{i=1}^n (\bar{e} - e_i)^2}} \tag{4}$$

$$RRMSE = \frac{1}{|\bar{e}|} \sqrt{\frac{\sum_{i=1}^n (e_i - m_i)^2}{n}} \tag{5}$$

$$R = \frac{\sum_{i=1}^n (e_i - \bar{e}_i)(m_i - \bar{m}_i)}{\sqrt{\sum_{i=1}^n (e_i - \bar{e}_i)^2 \sum_{i=1}^n (m_i - \bar{m}_i)^2}} \tag{6}$$

$$\rho = \frac{RRMSE}{1 + R} \tag{7}$$

4.17 Experimental setup of artificial neural network

The experimental setup focuses on evaluating the compressive and flexural strength of mixed design specimens.

Samples undergo casting and subsequent testing at intervals of 7, 14, 21, and 28 days. Compressive strength is determined using cube samples, and flexural strength is determined using beam specimens. After 28 days of casting, cube samples are weighed to determine their density (Algaifi et al., 2021; Chou 2013; Imam et al., 2021; Ly et al., 2021). The flow chart for the ANN is shown in Fig. 4.

References [72, 73, 74, 75] evaluate the feasibility of MARS and GEP models in predicting the 28-day compressive strength of SCC. In their research, values of compressive strength were estimated using multivariate adaptive regression spline approach [76, 77, 78, 79, 80, 81]. The findings indicated high ability of both the GEP and MARS models in predicting SCC strength Figs. 5, 6 and 7 and Table 1.

4.18 Gaps in knowledge

1. For AI algorithms to efficiently identify patterns and generate precise predictions, vast volumes of high-quality data are needed. However, due to variables including variations in material composition, curing conditions, and testing methodologies, it might be difficult to gather enough information about self-compacting concrete (SCC) qualities, particularly compressive strength.
2. Developing precise prediction models requires identifying the most pertinent features or input variables that affect the compressive strength of SCC. To find out which elements have the most effects on strength, researchers may need to investigate different concrete mix designs, admixtures, curing times, and environmental conditions.
3. While AI methods like deep learning can provide strong prediction skills, model interpretability is sometimes sacrificed in the process. Understanding how input features contribute to the anticipated compressive strength and learning about the underlying relationships in the data require striking a balance between interpretability and model complexity.
4. AI models that have been trained on historical data could find it difficult to generalize to novel situations or scenarios that greatly deviate from the training set. To make sure that prediction models can reliably handle differences in material qualities, environmental conditions, and construction processes, researchers need to investigate methods including transfer learning, domain adaptation, and robust optimization.

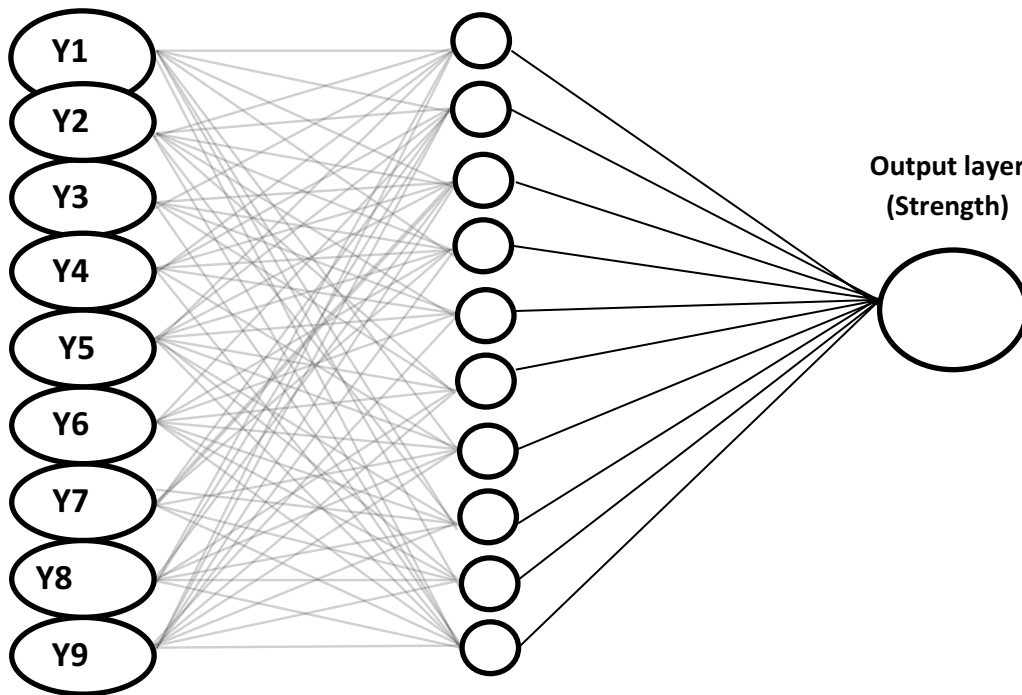


Fig. 4 ANN Architecture

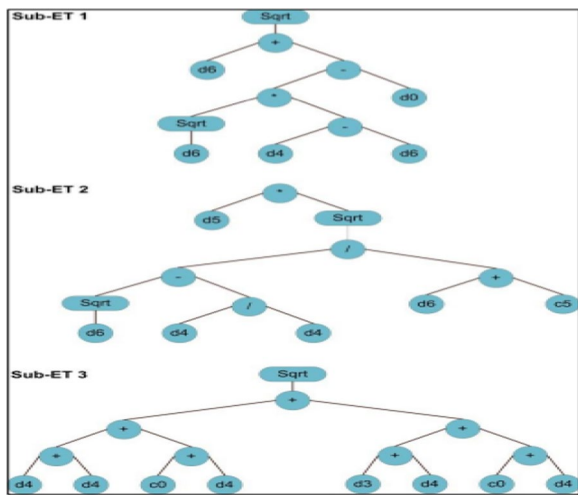


Fig. 5 GEP Models in Concrete Strength Prediction. Source: Milad, B and Valiollah, A: *Civil Engineering Journal*, Vol. 4, No. 7, July 2018

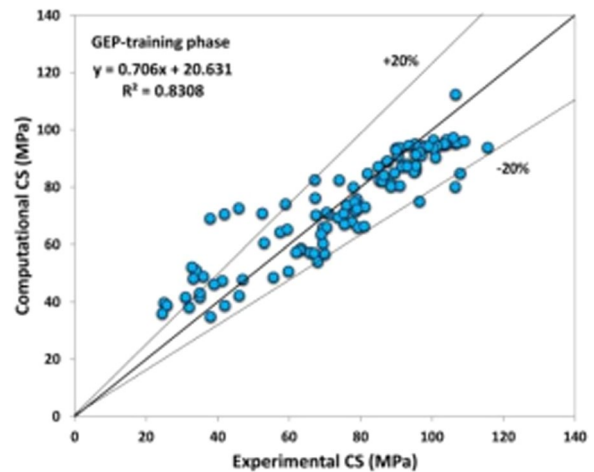


Fig. 6 Scatter Plot of Observed and Predicted Compressive Strength during the Training Phase of the GEP Model. Source: Milad, B and Valiollah, A: *Civil Engineering Journal*, Vol. 4, No. 7, July 2018

5. To evaluate AI models' dependability and generalizability, proper validation is essential. To properly assess the precision and resilience of their prediction models, researchers must use the right performance measures (such as mean absolute error, root mean square error) and validation strategies (such as hold-out validation, cross-validation).

6. Measurement inaccuracy and other kinds of variability make predicting concrete strength fundamentally questionable. AI approaches that use uncertainty quantification techniques, like ensemble learning or Bayesian neural networks, can produce more accurate forecasts and uncertainty estimations, which are necessary for engineering applications to make well-informed decisions.

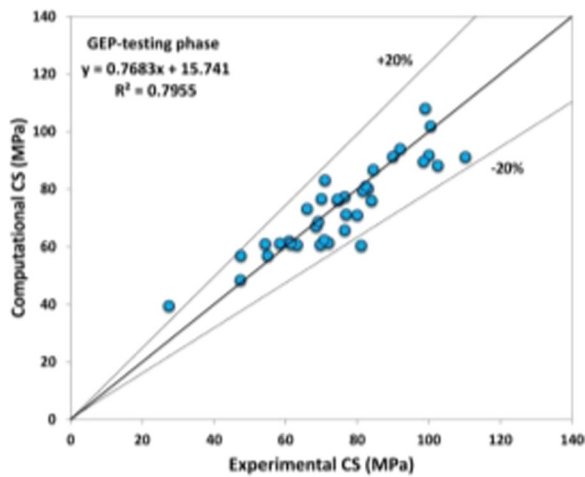


Fig. 7 Scatter Plot of Observed and Predicted Compressive Strength during the Testing Phase of the GEP Model. Source: Milad, B and Valiollah, A: *Civil Engineering Journal*, Vol. 4, No. 7, July 2018

Table 1 Parameters of optimized GEP models

Parameters	Description of parameters	Setting of parameters
P ₁	Function set	+, -, ×, /, exp, power
P ₂	Mutation rate	0.138
P ₃	Inversion rate	0.546
P ₄	One point and two point recombination rate	0.277
P ₅	Gene recombination rate	0.277
P ₆	Gene transportation rate	0.277
P ₇	Maximum tree depth	6
P ₈	Number of gene	3
P ₉	Number of chromosomes	30

Source: Milad, B and Valiollah, A: *Civil Engineering Journal*, Vol. 4, No. 7, July 2018

5 Conclusions

From the findings of this review, it was observed that the application of artificial intelligence techniques in predicting the compressive strength of self-compacting concrete yields approximate strength values closely aligned with experimental investigations. The research indicates that models achieving $R^2 > 0.8$ demonstrate a significant correlation between predicted values and experimental outcomes in the dataset. The review further noted that across all evaluated techniques, R^2 consistently exceeded 0.8 for 28-day compressive strength, affirming the suitability of these models for accurate predictions. Particularly, the ANN model exhibited notable consistency when compared to experimental results of concrete compressive strength, thereby establishing ANN as a robust predictive tool suitable for both in situ and experimental predictions. Consequently, it is recommended for formulating

various civil engineering properties requiring predictive capabilities. Artificial intelligence models offer significant time and resource savings by obviating the need for experimental tests, which can occasionally delay construction projects. Reinforcement learning techniques like Deep Deterministic Policy Gradient (DDPG) are more responsive to selections of hyperparameters than others. Through hyperparameter optimization, a set of parameters can be identified that optimizes model performance on test data, minimizing predefined loss functions. Overfitting, a common issue in machine learning where models perform poorly on new data due to over-adaptation to training data, can be mitigated through strategies such as cross-validation, data augmentation, dropout techniques, and careful feature selection.

Abbreviations

ANN	Artificial neural network
ANNs	Artificial neural networks
GP	Genetic programming
GEP	Genes expression programming
HMP	Hybrid multilayer perceptron
EANNs	Evolutionary artificial neural networks
GAs	Genetic algorithms
AI	Artificial intelligence
ANFIS	Adaptive network based fuzzy inference systems
FL	Fuzzy logic
GP	Genetic programming
DDPG	Deep deterministic policy gradient
KNN	K-nearest neighbor
ML	Machine learning
R	Correlation coefficient
MAE	Mean absolute error
RSE	Relative squared error
RRMSE	Relative root mean square error
SCC	Self-compacting concrete

Acknowledgements

The authors express gratitude to the Department of Civil Engineering, University of Nigeria Nsukka for the kind support to carry out this research.

Author contributions

All of the authors have contributed substantially to the article. Further, all of the authors have validated and approved the final manuscript.

Funding

This research received no external funding. All costs were incurred by the authors.

Data availability

All data used for this research are available in the manuscript.

Declarations

Ethics approval and consent to participate

The authors declare that the content of this paper is original and all the relevant and consulted materials are duly cited.

Competing interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Received: 29 January 2024 Revised: 29 April 2024 Accepted: 4 July 2024
Published online: 28 August 2024

References

- Aci, M., Aci, 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>
- 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 J Civil Eng*, 18, 1753–1758. <https://doi.org/10.1007/s12205-014-0524-0>
- Alade, I. O., Bagudu, A., Oyeohan, T. A., Rahman, M. A. 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 Programmes in Biomedicine*, 163, 135–142. <https://doi.org/10.1016/j.cmpb.2018.05.029>
- Algaifi, H. A., Bakar, S. A., Alyousef, R., Mohd-Sam, A. R., Alqarni, A. S., Ibrahim, M. H. W., Shahidan, S., Ibrahim, M., Salami, B. A., Algaifi, H. A., Bakar, S. A., Alyousef, R., Moh, A. R. S., Alqarni, A. S., Ibrahim, M. H. W., Shahidan, S., Ibrahim, M., & Salami, B. A. (2021). Machine learning and RSM models for prediction of compressive strength of smart bio-concrete. *Smart Structure Systems*, 28, 535. <https://doi.org/10.12989/SSS.2021.28.4.535>
- Ashish, D. K., & Verma, S. K. (2019). An overview on mixture design of self-compacting concrete. *Structural Concrete*, 20, 1–547. <https://doi.org/10.1002/suco.201700279>
- 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>
- Babajanzandeh, M., & Azizifa, V. (2018). Compressive strength prediction of self compacting concrete incorporating silica fume using artificial intelligence techniques. *Civil Engineering Journal*, 4(7), 1542.
- Babatunde, A. S., Mudassir, I., Abdulazeez, A., Fazal, E., Jalal, W. A., Arshad, J., Tafsirojijaman, T., Yue, L., & Abidhan, B. (2022). Estimating compressive strength of lightweight foamed concrete using neural, genetic and ensemble machine learning approaches. *Cement and Concrete Composites*. <https://doi.org/10.1016/j.cemconcomp.2022.104721>
- Bayer, I.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>
- Behnood, A., Behnood, V., Modiri, G. M., & Alyamac, K. E. (2017). Prediction of the compressive strength of normal and high-performance concretes using MSP model tree algorithm. *Construction and Building Materials*, 142, 199–207. <https://doi.org/10.1016/j.conbuildmat.2017.03.061>
- 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>
- 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>
- Ben, A. M., Al-Asri, Y., Zaher, M., Hafidi, A. A., & Burttschell, 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>
- Chakravarthy, H. G. 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, 13621. <https://doi.org/10.3390/su151813621>
- Chiew, F. H., Ng, C. K., Chai, K. C., & Tay, K. M. (2017). A fuzzy adaptive resonance theorybased model for mix proportion estimation of high-performance concrete. *Computer-Aided Civil and Infrastructure Engineering*, 32, 772–786. <https://doi.org/10.1111/MICE.12288>
- Chou, J.A.P. Materials, undefined, enhanced artificial intelligence for ensemble approach to predicting high performance concrete compressive strength, Elsevier, 2013 (n.d.), <https://www.sciencedirect.com/science/article/pii/S0950061813008088>.
- Danish, P and Ganesh, G.M. Self-compacting concrete—optimization of mix design procedure by the modifications of rational method, Springer Int. Publ. (2021), https://doi.org/10.1007/978-3-030-51485-3_25.
- Demir, V., & Doğu, R. (2023). 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>
- Dias, W.P.S., Pooliyadda, S.P. Neural networks for predicting properties of concretes with admixtures. *Construction and Building Materials*. Volume 15, Issue 7, October 2001, Pages 371–379. [https://doi.org/10.1016/S0950-0618\(01\)00006-X](https://doi.org/10.1016/S0950-0618(01)00006-X) Get rights and content
- 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>
- EFNARC, Specification and Guidelines for Self-Compacting Concrete, Rep. from EFNARC, 2002, p. 32.
- Erzin, Y. (2007). Artificial neural networks approach for swell pressure versus soil suction behaviour. *Canadian Geotechnical Journal*, 44, 1215–1223. <https://doi.org/10.1139/T07-052>
- Esmailkhanian, 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>
- Falliano, D., 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>
- 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 Assessment*. <https://doi.org/10.1007/s10661-018-6719-y>
- Ferreira, C. Gene expression programming in problem solving. *Soft Computing and Industry* (2002) 635–653, https://doi.org/10.1007/978-1-4471-0123-9_54.
- Ferreira, C. Gene Expression Programming Mathematical Modeling by an Artificial Intelligence, vol. 21, Springer, 2006, p. 478.
- Gagg, C. R. (2014). Cement and concrete as an engineering material: An historic appraisal and case study analysis. *Engineering Failure Analysis*, 40, 114–140. <https://doi.org/10.1016/j.engfailanal.2014.02.004>
- 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>
- Gesoglu, M., Güneyisi, E., Özturan, T., & Özbay, E. (2010). Modeling the mechanical properties of rubberized concretes by neural network and genetic programming. *Materials and Structures Construct*, 43, 31–45. <https://doi.org/10.1617/s11527-009-9468-0>
- 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>
- Golafshani, E. M., Behnood, A., & Arashpour, M. (2020a). 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>
- 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.
- 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 Construct*, 48, 1581–1602. <https://doi.org/10.1617/s11527-014-0256-0>
- Günel, A. Y., & Mehdi, R. (2023). 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>
- Imam, B.A., Salami, B.A., Oyeohan, T.A. Predicting the compressive strength of a quaternary blend concrete using Bayesian regularized neural network, 2021.
- Iqbal, M. F., Feng, L. Q., 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>
- Jovic, S., Radulovic, R., Kovacevic, M., Bozovi, C. R., & Sarkocecic, Z. (2019). Estimation of important variables for strength of concrete with high performance based on neuro fuzzy logic approach. *Structural Concrete*. <https://doi.org/10.1002/suco.201900330>
- Kandiri, A., Mohammadi, G. E., & 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. *Constructing and Building Materials*, 248, 118676. <https://doi.org/10.1016/j.conbuildmat.2020.118676>
- 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>
- 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, 1–23. <https://doi.org/10.3390/MA14051106>
- Khandelwal, M., Shirani, F. R., Monjezi, M., Armaghani, D. J., Majid, M. Z. B., & Yagiz, S. (2017). Function development for appraising brittleness of intact rocks using genetic programming and non-linear multiple regression models. *Engineering Computations*, 33, 13–21. <https://doi.org/10.1007/s00366-016-0452-3>
- 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, 04016092. [https://doi.org/10.1061/\(ASCE\)MT.1943-5533.0001602](https://doi.org/10.1061/(ASCE)MT.1943-5533.0001602)
- Kovačević, M., Lozančić, S., Nyarko, E.K., Hadzima-nyarko, M. Modeling of compressive strength of self-compacting rubberized concrete using machine learning. *Materials* 14 (2021) 4346
- 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(2021), 17331–17351.
- Madani, H., Mohammad, K., & 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, 04020021.
- Mehmannavaz, T., Khalilikhorrām, 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, 1872.
- Mogaraju, J. K. (2023). 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>
- Nehdi, M., Djebbar, Y., & Khan, A. (2001). Neural network model for preformed-foam cellular concrete. *Materials Journal*, 98, 402–409. <https://doi.org/10.14359/10730>
- Neira, P., Bennun, L., Pradena, M., & Gomez, J. (2020). Prediction of concrete compressive strength through artificial neural network. *Gradjevinar*, 72, 585–592. <https://doi.org/10.14256/JCE.2438.2018>
- 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 Infrastructure Engineering*, 34, 316–332. <https://doi.org/10.1111/mice.12422>
- 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 hydratedlime 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>
- 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>
- Öztaş, A., Pala, M., Özbay, E., Kanca, E., Çag, L. N., & Bhatti, M. A. (2006). Predicting the compressive strength and slump of high strength concrete using neural network. *Construction and Building Materials*, 20, 769–775. <https://doi.org/10.1016/j.conbuildmat.2005.01.054>
- Prasad, B. K. R., Eskandari, H., & Reddy, V. (2009). Prediction of compressive strength of SCC and HPC with high volume fly ash using ANN. *Construction and Building Materials*, 23, 117–128. <https://doi.org/10.1016/j.conbuildmat.2008.01.014>
- Rahmat, M.S. and Yasin Mousavi. Fresh and hardened properties of self-compacting concrete containing metakaolin. *Construction and Building Materials*. Volume 35, 2012, Pages 752–760. <https://doi.org/10.1016/j.conbuildmat.2012.04.109>
- Rana, F. T., Muhammad, H. N., Jawad, A., Hafiz, W., Ali, M., Danish, F., Ahsen, M., & Faheem, B. (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>
- Rob, G. and Noel, D. Self-compacting concrete. *Advanced Concrete Technology*. Elsevier. Volume 4, 2003, Pages 1–23. <https://doi.org/10.1016/B978-075065686-3/50295-0>
- Saridemir, M. (2010). Genetic programming approach for prediction of compressive strength of concretes containing rice husk ash. *Construction and Building Materials*, 24, 1911–1919. <https://doi.org/10.1016/J.CONBUILDMAT.2010.04.011>
- Sergio, L., & Mauro, S. (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](https://doi.org/10.1016/S0950-0618(97)00007-X)
- Seung-Chang L. Prediction of concrete strength using artificial neural networks, *Engineering Structures*, Volume 25, Issue 7, 003, Pages 849–857, [https://doi.org/10.1016/S0141-0296\(03\)00004-X](https://doi.org/10.1016/S0141-0296(03)00004-X). Elsevier
- Seung-Chang, L. (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](https://doi.org/10.1016/S0141-0296(03)00004-X)
- 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, 50–74. <https://doi.org/10.25073/JAEC.202151.308>
- Shi, C., Wu, Z., Lv, K., & Wu, L. (2015). A review on mixture design methods for self-compacting concrete. *Constructing and Building Materials*, 84, 387–398.
- 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, 780–786. <https://doi.org/10.1016/j.advengsoft.2011.05.016>
- 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, 44. <https://doi.org/10.3390/BUILDINGS11020044>
- Sonebi, M. (2004). Development of statistical models for mixture design of high-volume fly ash self-consolidating concrete. *ACI Materials Journal*, 101, 339–346. <https://doi.org/10.14359/13363>
- Surya, G. L., Tejaswini, A., & Venkateswara, R. (2020). A detailed report on various behavioral aspects of self-compacting concrete. *Materials Today Processing*, 33, 839–844. <https://doi.org/10.1016/j.matpr.2020.06.273>
- Syed, M. I., Saad, S., Ansari, S., & Danish, H. (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>
- 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. *Cement Based Composites*, 4(1), 5756. <https://doi.org/10.36937/cebacom.2023.5756>
- Zhou, Q., Wang, F., & Zhu, F. (2016). Estimation of compressive strength of low 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>

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.