Magazine of Civil Engineering

journal homepage: http://engstroy.spbstu.ru/

ISSN
2712-8172

Research article

UDC 666.972:004.8

DOI: 10.34910/MCE.141.10

Adaptive cost-constrained optimization of concrete mixtures using machine learning-guided genetic algorithms

M.B. Beigi

Islamic Azad University, Semnan, Iran

mohsenbeigi59@gmail.com

Keywords: concrete mix optimization, machine learning, genetic algorithm, compressive strength prediction, cost constraint, android application

Abstract. This study presents an adaptive framework for optimizing high-performance concrete mixtures by integrating machine learning with a genetic algorithm under cost constraints. An experimental dataset was used to train an XGBoost model, which accurately predicts 28-day compressive strength. The trained machine learning model was embedded as the objective function within the genetic algorithm to maximize compressive strength while incorporating cost limitations defined by user-provided unit prices and budget. Unlike most previous studies that treat cost and strength as two separate objectives in multi-objective formulations, this work introduces cost as a constraint and strength as the sole optimization objective, thereby simplifying the decision-making process. To bridge the gap between theory and practice, an Android application was developed. The application enables users to input real-time material prices and budget limits, which are transmitted to a server hosting the machine learning model and genetic algorithm. The server computes optimized mix proportions and returns both the predicted compressive strength and the optimal design to the user interface. The proposed adaptive optimization framework was shown to effectively adjust to market price fluctuations and varying budget scenarios, providing a practical and flexible solution for real-world applications. Furthermore, the single-objective formulation ensures a unique optimal solution, avoiding the complexity of selecting among multiple Pareto-optimal alternatives.

Citation: Beigi, M.B. Adaptive cost-constrained optimization of concrete mixtures using machine learning-guided genetic algorithms. Magazine of Civil Engineering. 2026. 19(1). Article no. 14110. DOI: 10.34910/MCE.141.10

1. *Introduction*

Concrete is one of the most widely used construction materials due to its versatility, durability, and cost-effectiveness. It is a composite material composed primarily of cement, aggregates, water, and chemical admixtures. Cement acts as the binding agent, holding the mixture together, while aggregates contribute to bulk and significantly influence the mechanical and durability properties of both fresh and hardened concrete. Water facilitates the hydration of cement and affects the porosity, strength, and durability of the hardened material. Chemical admixtures are employed to regulate physical properties and improve workability and performance [1, 2].

The 28-day compressive strength of concrete is a critical performance indicator in both research and practice. It is affected by multiple factors including mix proportions, water-to-cement ratio, curing conditions, and the properties of individual components. Accurate prediction of this property is essential for ensuring safety, performance, and compliance with standards [3, 4].

Cost is another significant factor in concrete production. Raw material expenses, such as cement, aggregates, and admixtures, can constitute more than half of the total production cost. Rising material prices, particularly for cement, can substantially impact project budgets. Optimizing concrete mixtures with supplementary materials like fly ash or slag can reduce costs while maintaining or enhancing performance [5, 6].

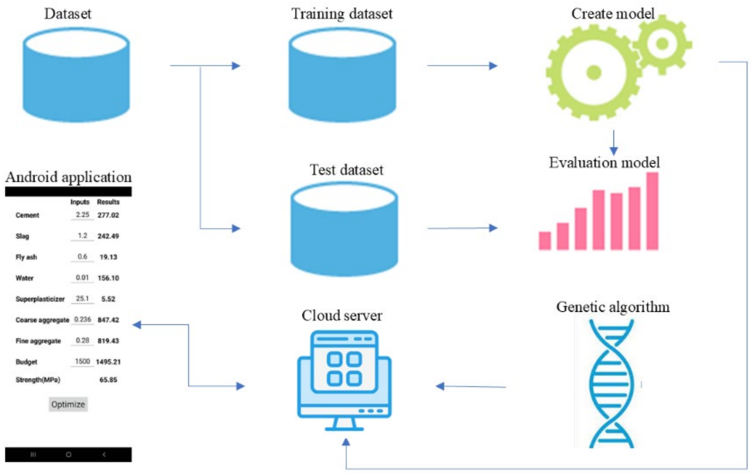
However, compressive strength and production cost are generally conflicting objectives, since an increase in strength is often associated with higher costs, and vice versa. In many studies, this trade-off has been addressed through multi-objective optimization approaches, where compressive strength and cost are treated as two separate objectives to be optimized simultaneously [7–17].

Predicting concrete compressive strength is a complex task due to nonlinear relationships between input variables, such as mix proportions and curing conditions, and output variables. Traditional empirical models often fall short in capturing these interactions. Machine learning techniques, such as extreme gradient boosting and artificial neural networks, have demonstrated high accuracy in estimating compressive strength, effectively managing complex datasets and improving prediction reliability [18, 19].

Integrating genetic algorithms with machine learning models has proven highly effective for optimizing concrete mix designs to achieve maximum compressive strength under specified constraints. Genetic algorithm explores vast solution spaces and identifies optimal solutions by mimicking natural selection. When combined with machine learning models predicting compressive strength, genetic algorithms can fine-tune mix proportions for enhanced performance. Studies have demonstrated that integrating machine learning estimators with genetic algorithms enables optimization of concrete mixtures, balancing strength, cost, and durability under quality constraints [20, 21].

In this study, an experimental dataset was employed to develop a machine learning model for predicting the 28-day compressive strength of concrete. The trained model was subsequently coupled with a genetic algorithm to optimize concrete mixture proportions subject to both technical and economic constraints. In contrast to previous studies, construction cost is not formulated as a secondary optimization objective; instead, it is explicitly imposed as a constraint. This formulation allows market price fluctuations to be flexibly incorporated into the optimization framework. By enabling users to specify material unit prices and a maximum allowable budget, the proposed approach ensures that compressive strength is maximized while strictly satisfying the prescribed cost limitation.

The novelty of this work lies in two main aspects. First, the optimization problem is formulated as a single-objective genetic algorithm, where the objective function is the compressive strength predicted by the machine learning model, while cost and proportional limitations are treated as constraints. This design offers a more flexible and practical framework compared to traditional multi-objective approaches. Second, to facilitate real-world application, an Android application was developed that allows users to input unit prices and budget information. These inputs are transmitted to a server hosting the machine learning model and genetic algorithm, which computes the optimal mixture proportions and returns the results to the application (Fig. 1). This user-oriented design makes the framework not only technically innovative but also highly accessible for practical decision-making in concrete mix design.



	Inputs	Results
Cement	2.25	277.02
Slag	1.2	242.49
Fly ash	0.6	19.13
Water	0.01	156.10
Superplasticizer	25.1	5.52
Coarse aggregate	0.236	847.42
Fine aggregate	0.28	819.43
Budget	1500	1495.21
Strength(MPa)		65.85

Figure 1. Schematic representation of the developed framework.

2. Materials and Methods

2.1. Problem Formulation

The present study addresses the optimization of high-performance concrete mixture proportions using a single-objective genetic algorithm. The objective function (cost function) is to maximize the 28-day compressive strength predicted by a trained machine learning model:

$$f\left(X_1, X_2, \ldots, X_7\right) = \hat{f}_{ML}\left(X_1, X_2, \ldots, X_7\right), \quad (1)$$

where $\hat{f}_{ML}$ represents the output of the machine learning model and the input variables X_i (all in kg/m^3) the concrete constituents as defined in Table 1.

Table 1. Concrete components, specific gravities, and constraint ratios.

Component	Symbol	Specific Gravity (kg/m³)	Min Ratio	Max Ratio	Notes / Constraint Type
Cement	X_1	3.15	—	—	Part of binder
Slag	X_2	2.80	0	0.61	Slag-to-binder ratio
Fly Ash	X_3	2.50	0	0.61	Fly Ash-to-binder ratio
Water	X_4	1.00	0.23	0.90	Water-to-binder ratio
Superplasticizer	X_5	1.35	0	0.13	SP-to-cement ratio
Coarse Aggregate	X_6	2.50	1.18	5.62	Coarse-to-binder ratio
Fine Aggregate	X_7	2.65	0.35	0.54	Fine-to-total aggregate ratio

The genetic algorithm aims to determine the optimal mixture proportions under a set of general, proportional, and cost constraints.

2.1.1. Volume constraint

The total volume of the concrete mixture must equal 1 m^3, enforced using the specific gravities G_i of each component:

$$\sum_{i=1}^{7} \frac{X_i}{G_i} = 1000. \quad (2)$$

This ensures that the genetic algorithm-generated mixture corresponds to a physically feasible 1 m^3 concrete volume.

2.1.2. Ratio constraints

To maintain logical and practical mixture proportions, the ratio constraints listed in Table 1 are imposed [16].

2.1.3. Cost constraint

A user-defined budget constraint is introduced as an innovative aspect of this study. The user specifies the unit price P_i (per kg) for each material and the maximum allowable budget $\sum_{i=1}^{7} P_i X_i \le Budget$ is ensures that the optimized mixture maximizes compressive strength without exceeding the user's budget.

2.2. Implementation

2.2.1. Framework description

As illustrated in Fig. 2, the proposed framework operates as follows. The user, through an Android application, inputs the unit prices of the concrete constituents (cement, slag, fly ash, water, superplasticizer, coarse aggregate, and fine aggregate) along with the maximum allowable budget. This information is transmitted to a web server via an API. On the server, a machine learning model – trained on an experimental dataset – is hosted to predict the 28-day compressive strength of concrete for given inputs X_1 through X_7. Simultaneously, a genetic algorithm is implemented on the server, where the output of the machine learning model serves as the fitness function to be maximized. The genetic algorithm searches for optimal mixture proportions while satisfying the volume constraint, ratio constraints (e.g., water-to-binder ratio, slag-to-binder ratio, fly ash-to-binder ratio, coarse-to-total aggregate ratio, fine-to-binder ratio, and superplasticizer-to-cement ratio), and the budget.

Finally, the genetic algorithm computes the optimal values of $X_1, \ldots, X_7$ that maximize the predicted compressive strength while ensuring that all constraints are satisfied. The optimized mix proportions are then returned to the Android application, enabling the user to conveniently view and utilize the results without requiring advanced knowledge of optimization methods.

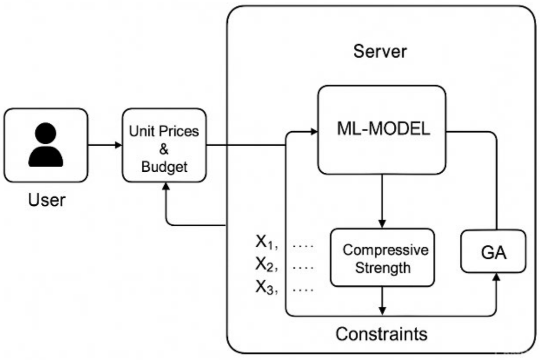


Figure 2. Framework of the proposed methodology.

2.2.2. Modeling

In this study, a dataset of 425 samples of high-performance concrete was employed for developing the machine learning model. The input features consisted of seven variables – cement, slag, fly ash, water, superplasticizer, coarse aggregate, and fine aggregate – while the output variable was the 28-day compressive strength. The experimental data were originally obtained from the program reported by Yeh [22], in which ASTM Type I Portland cement was used. Aggregate properties were described qualitatively, indicating that the coarse aggregate consisted of crushed natural stone with a maximum size of 10 mm, whereas the fine aggregate was washed river sand with a fineness modulus of approximately 3 mm. This dataset provided sufficient diversity to capture the variability in mixture proportions and their influence on compressive strength.

To incorporate economic considerations, an additional column representing the cost of each mixture was computed using unit prices ($/kg) for all constituents: cement (0.11), slag (0.06), fly ash (0.055), water (0.000024), superplasticizer (2.94), coarse aggregate (0.01), and fine aggregate (0.006). Descriptive statistics for the dataset used in this study are presented in Table 2. The distributions of the input variables compressive strength and cost are shown in Fig. 3, illustrating the range and variability of the experimental data.

Table 2. Descriptive statistics for the concrete dataset.

	mean	**std**	**min**	**25 %**	**50 %**	**75 %**	**max**
Cement	265.44	104.67	102.00	160.20	261.00	323.70	540.00
Slag	86.29	87.83	0.00	0.00	94.70	160.50	359.40
Fly Ash	62.80	66.23	0.00	0.00	60.00	120.00	200.10
Water	183.06	19.33	121.80	171.00	185.00	193.30	247.00
Superplasticizer	7.00	5.39	0.00	0.00	7.80	10.30	32.20
Coarse Aggregate	956.06	83.80	801.00	882.60	953.20	1013.20	1145.00
Fine Aggregate	764.38	73.12	594.00	712.00	769.30	811.50	992.60
Compressive Strength	36.75	14.71	8.54	26.23	33.76	44.39	81.75
Cost	72.55	19.05	34.93	59.24	72.07	83.67	166.86

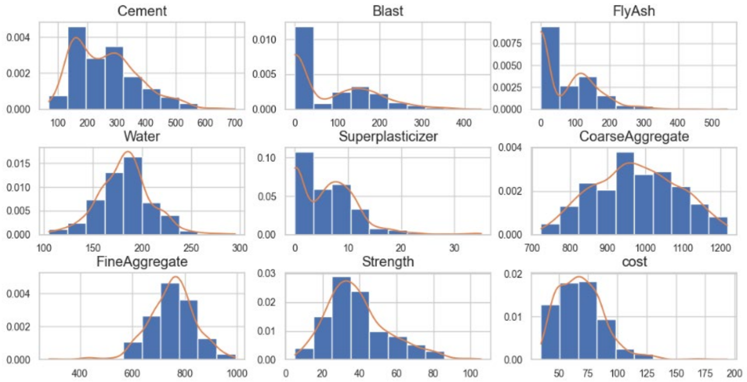
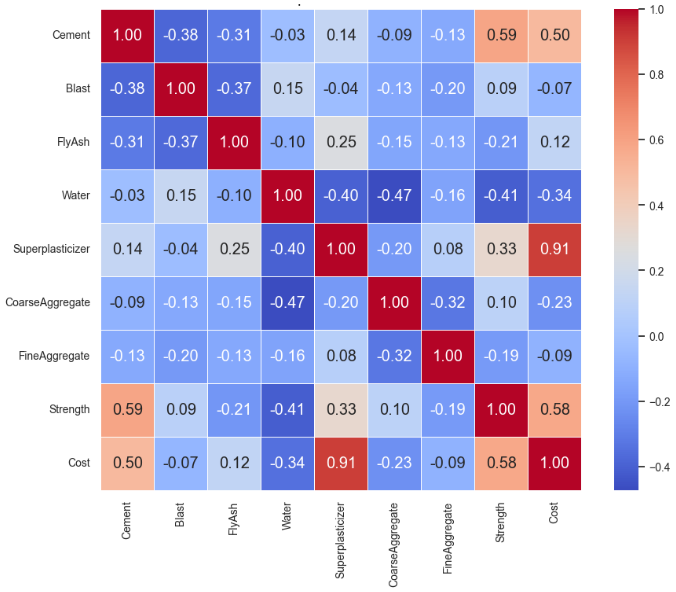


Figure 3. Histograms of input and output variables used in the study.

The correlation map (Fig. 4) displays the pairwise Pearson correlation coefficients between the main concrete mix variables and cost. Strong positive correlations are observed between superplasticizer and cost, compressive strength and cement, and compressive strength and cost, indicating that higher amounts of superplasticizer and cement lead to increased compressive strength and higher overall cost [23–26].



	Cement	Blast	FlyAsh	Water	Superplasticizer	CoarseAggregate	FineAggregate	Strength	Cost
Cement	1.00	-0.38	-0.31	-0.03	0.14	-0.09	-0.13	0.59	0.50
Blast	-0.38	1.00	-0.37	0.15	-0.04	-0.13	-0.20	0.09	-0.07
FlyAsh	-0.31	-0.37	1.00	-0.10	0.25	-0.15	-0.13	-0.21	0.12
Water	-0.03	0.15	-0.10	1.00	-0.40	-0.47	-0.16	-0.41	-0.34
Superplasticizer	0.14	-0.04	0.25	-0.40	1.00	-0.20	0.08	0.33	0.91
CoarseAggregate	-0.09	-0.13	-0.15	-0.47	-0.20	1.00	-0.32	0.10	-0.23
FineAggregate	-0.13	-0.20	-0.13	-0.16	0.08	-0.32	1.00	-0.19	-0.09
Strength	0.59	0.09	-0.21	-0.41	0.33	0.10	-0.19	1.00	0.58
Cost	0.50	-0.07	0.12	-0.34	0.91	-0.23	-0.09	0.58	1.00

Figure 4. Correlation map of concrete mix variables and cost.

Three different machine learning models were considered: extreme gradient boosting [19], multilayer perceptron neural network [27], and random forest [28]. To ensure robust evaluation, a 10-fold cross-validation approach was employed during training. Model performance was compared using the mean squared error metric.

2.2.3. Optimization

Genetic algorithm is an evolutionary optimization technique inspired by the process of natural selection [29]. It operates by encoding potential solutions, known as chromosomes, and evolving them over successive generations to find the optimal solution. In the present study, each chromosome represents a possible concrete mixture, defined by the seven input variables $X_1 - X_7$ (cement, slag, fly ash, water, superplasticizer, coarse aggregate, and fine aggregate). The fitness of each chromosome is evaluated using the trained machine learning model, which predicts the 28-day compressive strength, while budget and proportional constraints are simultaneously enforced. As illustrated in Fig. 5, the genetic algorithm applies selection to retain high-performing chromosomes, crossover to recombine parent solutions, and mutation to introduce diversity and avoid premature convergence. Through iterative search of the solution space, the genetic algorithm gradually improves the population until the optimal mix proportions that maximize compressive strength without exceeding the budget are obtained.

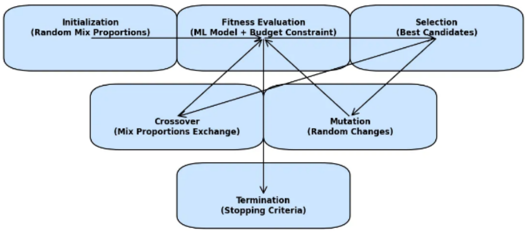


Figure 5. Schematic representation of the genetic algorithm framework.

2.2.4. Implementation and deployment

After the machine learning model was trained and saved, it was deployed on the PythonAnywhere hosting platform. Along with the trained model, the genetic algorithm code was also implemented on the same host. To enable practical use, an Android application was developed using App Inventor, through which users could provide unit prices of concrete components and the maximum allowable budget. These inputs were transmitted via an API to the server. The Flask framework was employed to handle communication, ensuring efficient transfer of requests and responses between the application and the server. On the server side, the genetic algorithm utilized the user-provided information together with the trained machine learning model to compute the optimal mix proportions $\left(X_1 - X_7\right)$. The optimization process was carried out with the objective of maximizing compressive strength without exceeding the specified budget. Finally, the computed optimal proportions and the corresponding compressive strength were returned to the Android application through the API, enabling the user to conveniently access and interpret the results.

3. Results and Discussion

3.1. Modeling Results

The experimental dataset was divided into two subsets: 75 % for training and 25 % for testing. The training data were further used in a 10-fold cross-validation scheme to ensure robust model evaluation. Three different machine learning models were considered: Extreme gradient boosting (XGBoost), random forest, and a multi layer perceptron neural network. After training, the models were tested on the 25 % test set, and their predictive performances were compared using the mean squared error as the evaluation criterion. The results demonstrated that XGBoost achieved the best performance with a mean squared error of 17.56 MPa^2, followed by random forest with a mean squared error of 17.90 MPa^2, and the multi layer perceptron with a mean squared error of 18.22 MPa^2. These findings indicate that XGBoost provides superior predictive accuracy for 28-day compressive strength compared to the other tested models. Previous studies have also confirmed the strong capability of XGBoost in predicting the compressive strength of concrete [19, 30–55].

The specifications of the XGBoost model are summarized in Table 3. This table highlights the model's performance with the hyperparameters set to a maximum depth of 4 and the number of estimators set to 500. The maximum depth parameter, fixed at 4, was selected to balance model complexity and mitigate overfitting. Meanwhile, the use of 500 decision trees (estimators) was chosen to improve predictive accuracy. The results demonstrate that the model achieved high predictive performance, with low error values in estimating the 28-day compressive strength of concrete.

Table 3. Performance metrics and hyperparameters of the optimized XGBoost model.

Metrics and hyperparameters	Value
max_depth	4
n_estimators	500
Mean Squared Error	17.56
Root Mean Squared Error	4.19
Mean Absolute Error	2.82
R-squared	0.93

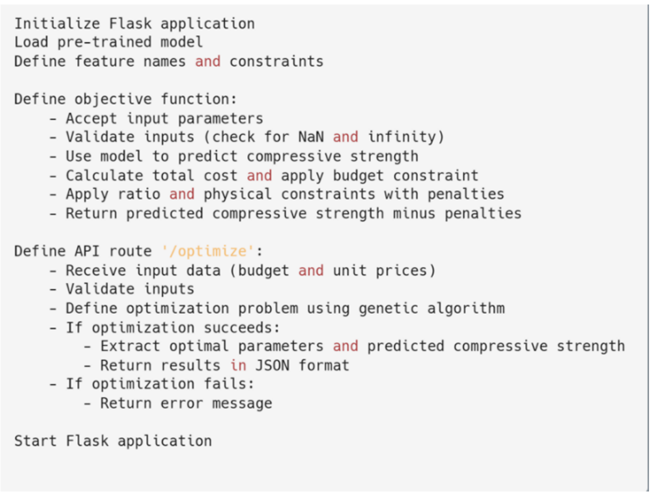
3.2. Genetic Algorithm

In this study, the implementation of the genetic algorithm was designed within an adaptive framework to account for varying unit prices of materials and budget limitations provided by the user. Unlike traditional static optimization approaches, the proposed genetic algorithm incorporates economic constraints in an adaptive manner during the optimization process. The cost function, defined as the 28-day compressive strength predicted by the trained machine learning model, is maximized while simultaneously satisfying proportional mixture constraints and the user-defined budget constraint. Since unit prices and budget vary depending on market conditions and project-specific requirements, the optimization problem is adaptively solved for each input scenario. The trained machine learning model, deployed on the server, evaluates candidate solutions during the genetic algorithm search, ensuring that only feasible mixtures are explored. This integration of genetic algorithm with machine learning and adaptive cost constraints enables flexible and practical optimization, making the approach suitable for real-world applications where cost and material availability fluctuate.

The pseudocode shown in Fig. 6 represents the implementation of the genetic algorithm on the server. First, a Flask application is initialized to handle communication between the Android client and the optimization framework. The pre-trained machine learning model, which predicts the 28-day compressive strength of concrete, is loaded into memory. Feature names and all associated constraints are then defined, including both proportional mixture rules and the user-defined budget constraint.

The core of the algorithm is the objective function. This function accepts input parameters (mixture proportions), validates them by checking for invalid values, such as NaN or infinity, and predicts compressive strength using the machine learning model. It also calculates the total cost of the mixture and applies the budget constraint. Any violations of the ratio constraints are penalized, ensuring that only feasible solutions are promoted. The genetic algorithm therefore aims to maximize the predicted compressive strength while minimizing penalties due to constraint violations.

Next, an API route /optimize is defined within the Flask application. This route receives input data from the Android application, including the maximum budget and unit prices of concrete components. After validating the inputs, the genetic algorithm-based optimization problem is solved. If optimization is successful, the algorithm extracts the optimal mixture proportions and the corresponding predicted compressive strength, and returns these results in JSON format to the client. If optimization fails, an appropriate error message is generated. Finally, the Flask application is launched, enabling real-time interaction between the client and the optimization framework.



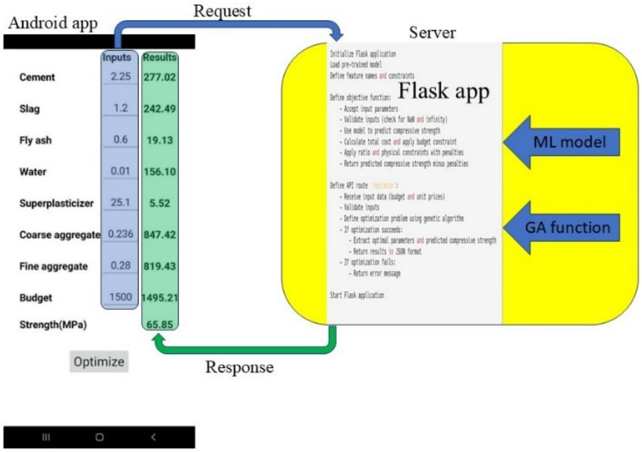
```
Initialize Flask application
Load pre-trained model
Define feature names and constraints

Define objective function:
    - Accept input parameters
    - Validate inputs (check for NaN and infinity)
    - Use model to predict compressive strength
    - Calculate total cost and apply budget constraint
    - Apply ratio and physical constraints with penalties
    - Return predicted compressive strength minus penalties

Define API route '/optimize':
    - Receive input data (budget and unit prices)
    - Validate inputs
    - Define optimization problem using genetic algorithm
    - If optimization succeeds:
        - Extract optimal parameters and predicted compressive strength
        - Return results in JSON format
    - If optimization fails:
        - Return error message

Start Flask application
```

Figure 6. Pseudocode of the server-side implementation of the genetic algorithm integrated with the trained machine learning model and Flask framework.

3.3. Practical Implementation

To bridge the gap between theoretical research and practical implementation, an Android application was developed using the App Inventor platform. Considering the rapid advancement of smart technologies and the widespread use of smartphones in daily life, deploying an Android-based solution provides an effective means of operationalizing the outcomes of this study. The application receives user inputs, such as unit costs of raw materials and the available budget, and communicates with a server hosting the pre-trained machine learning model and the implemented genetic algorithm. Based on these inputs, it computes the optimal mixture proportions and the maximum achievable compressive strength within the specified budget and presents the results to the user. In this way, the theoretical contributions of the research are translated into a practical tool with direct applicability in the concrete industry.

Fig. 7 illustrates the overall architecture of the developed framework. On the left, a screenshot of the Android application (designed in MIT App Inventor) is shown. The application interface consists of three main columns: (i) labels indicating input parameters, such as cement, aggregate, and budget, (ii) input fields for entering unit prices and the allowable budget, and (iii) result fields that display the optimized mix design and predicted compressive strength returned from the server. An "Optimize" button allows the user to send the entered data to the server. On the right, an icon representing the PythonAnywhere hosting platform is illustrated, where the trained machine learning model, the genetic algorithm code, and the Flask framework are deployed. The Android app sends a request (containing unit prices and budget) to the server, and in return, the server sends a response (optimized mix proportions and compressive strength), which is displayed in the application interface.

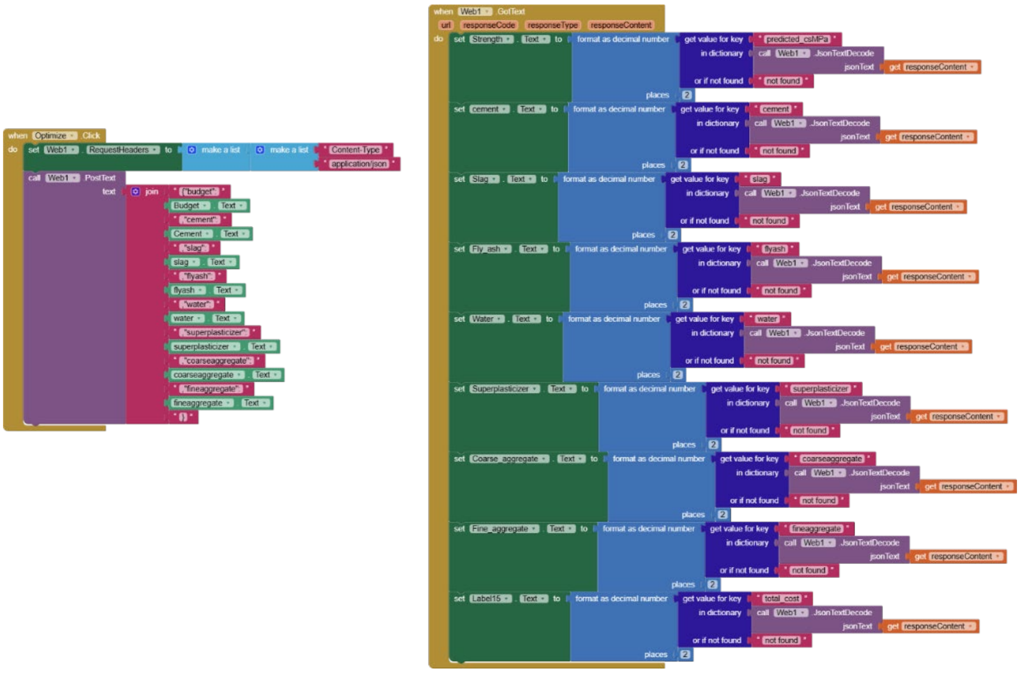


	Inputs	Results
Cement	2.25	277.02
Slag	1.2	242.49
Fly ash	0.6	19.13
Water	0.01	156.10
Superplasticizer	25.1	5.52
Coarse aggregate	0.236	847.42
Fine aggregate	0.28	819.43
Budget	1500	1495.21
Strength(MPa)		65.85

```
Initialize Flask application
Load pre-trained model
Define feature names and constraints

Define objective function:
    - Accept input parameters
    - Validate inputs (check for NaN and infinity)
    - Use model to predict compressive strength
    - Calculate total cost and apply budget constraint
    - Apply ratio and physical constraints with penalties
    - Return predicted compressive strength minus penalties

Define API route '/optimize':
    - Receive input data (budget and unit prices)
    - Validate inputs
    - Define optimization problem using genetic algorithm
    - If optimization succeeds:
        - Extract optimal parameters and predicted compressive strength
        - Return results in JSON format
    - If optimization fails:
        - Return error message

Start Flask application
```

Figure 7. Schematic representation of the developed framework integrating the Android application with the Flask-based server on PythonAnywhere.

In Fig. 8, the essential App Inventor blocks required for developing the Android application are illustrated. Two groups of blocks can be observed. The blocks on the left are responsible for sending the input information provided by the user – including the unit prices of concrete components and the maximum allowable budget – to the server. The blocks on the right handle the response received from the server, which contains the optimized mixture proportions of the seven components within the specified budget. Additionally, these blocks display the corresponding 28-day compressive strength predicted by the trained machine learning model for the optimized mixture design.



```
when Optimize .Click
do  set Web1 . RequestHeaders to  make a list  make a list  "Content-Type"
                                                            "application/json"
    call Web1 .PostText
                   text  join  "{"budget":"
                               Budget . Text
                               ","cement":"
                               Cement . Text
                               ","slag":"
                               slag . Text
                               ","flyash":"
                               flyash . Text
                               ","water":"
                               water . Text
                               ","superplasticizer":"
                               superplasticizer . Text
                               ","coarseaggregate":"
                               coarseaggregate . Text
                               ","fineaggregate":"
                               fineaggregate . Text
                               "}"
```

```
when Web1 .GotText
 url  responseCode  responseType  responseContent
do  set Strength . Text to  format as decimal number  get value for key  "predicted_csMPa"
                                                        in dictionary  call Web1 .JsonTextDecode
                                                                                  jsonText  get responseContent
                                                        or if not found  "not found"
                                                places  2
    set cement . Text to  format as decimal number  get value for key  "cement"
                                                      in dictionary  call Web1 .JsonTextDecode
                                                                                jsonText  get responseContent
                                                      or if not found  "not found"
                                              places  2
    set Slag . Text to  format as decimal number  get value for key  "slag"
                                                    in dictionary  call Web1 .JsonTextDecode
                                                                              jsonText  get responseContent
                                                    or if not found  "not found"
                                            places  2
    set Fly_ash . Text to  format as decimal number  get value for key  "flyash"
                                                       in dictionary  call Web1 .JsonTextDecode
                                                                                 jsonText  get responseContent
                                                       or if not found  "not found"
                                               places  2
    set Water . Text to  format as decimal number  get value for key  "water"
                                                     in dictionary  call Web1 .JsonTextDecode
                                                                               jsonText  get responseContent
                                                     or if not found  "not found"
                                             places  2
    set Superplasticizer . Text to  format as decimal number  get value for key  "superplasticizer"
                                                                in dictionary  call Web1 .JsonTextDecode
                                                                                          jsonText  get responseContent
                                                                or if not found  "not found"
                                                        places  2
    set Coarse_aggregate . Text to  format as decimal number  get value for key  "coarseaggregate"
                                                                in dictionary  call Web1 .JsonTextDecode
                                                                                          jsonText  get responseContent
                                                                or if not found  "not found"
                                                        places  2
    set Fine_aggregate . Text to  format as decimal number  get value for key  "fineaggregate"
                                                              in dictionary  call Web1 .JsonTextDecode
                                                                                        jsonText  get responseContent
                                                              or if not found  "not found"
                                                      places  2
    set Label15 . Text to  format as decimal number  get value for key  "total_cost"
                                                       in dictionary  call Web1 .JsonTextDecode
                                                                                 jsonText  get responseContent
                                                       or if not found  "not found"
                                               places  2
```

Figure 8. App Inventor blocks used in the Android application.

4. Conclusion

In this study, a machine learning-guided optimization framework was developed to design cost-effective and high-performance concrete mixtures. An XGBoost model was trained on an experimental dataset of 425 samples to accurately predict the 28-day compressive strength of concrete. The trained model was then integrated with a genetic algorithm to optimize mixture proportions under adaptive cost constraints, allowing flexible consideration of varying market conditions. Unlike conventional multi-objective approaches that simultaneously optimize strength and cost, this study formulated the problem as a single-objective optimization with compressive strength maximization, while cost was incorporated as a constraint. This approach enables practical adaptability while ensuring budget limits are not exceeded.

To bridge the gap between theory and practice, an Android application was developed using App Inventor. The application allows users to input unit prices of components and defines budget limits, which are then transmitted to a server hosting the machine learning model and genetic algorithm implemented via Flask. The server computes optimized mixture proportions and returns both the mix design and predicted compressive strength to the user interface. This practical implementation demonstrates the usability of the framework even for non-expert users, supporting data-driven decision-making in construction material design.

An important advantage of the proposed framework lies in its adaptability to real-world economic conditions. In scenarios where market prices of materials fluctuate, the system can seamlessly adjust optimization results according to updated unit prices. Similarly, when project budgets vary – ranging from generous allocations to highly cost-sensitive cases – the framework can readily adapt to provide optimized designs that meet the specified financial constraints. This flexibility makes the proposed approach highly practical and valuable for real construction applications where both material costs and budget availability are inherently dynamic.

Moreover, unlike multi-objective approaches that typically generate a large set of non-dominated solutions (Pareto front), which may cause confusion for practitioners in selecting the most suitable mix design, the single-objective formulation adopted here provides a unique optimal solution. This simplifies the decision-making process by directly identifying the best mix design under the specified budget, thereby enhancing the usability of the framework in practical scenarios.

Workability, commonly quantified by slump, is a critical performance requirement for concrete mixtures, as it directly affects placing, compaction, and overall constructability, particularly in high-strength concrete. Future work may extend the proposed framework by incorporating concrete workability as a constraint rather than as an additional optimization objective. This can be achieved by developing a dedicated machine learning model to predict slump based on mixture proportions. During the optimization process, the objective function would remain the maximization of compressive strength, while a minimum target slump value would be imposed as a constraint. Such a constrained optimization strategy would ensure that the optimized concrete mixtures achieve high strength while maintaining adequate workability, thereby enhancing the practical applicability of the proposed framework.

References

1. Neville, A.M. Properties of concrete. 4th edn. Longman. Essex, 1995. 844 p.
2. Mehta, P.K., Monteiro, P.J. Concrete microstructure, properties, and materials.: McGraw-Hill. 3rd edn. McGraw-Hill. New York, 2006. 684 p.
3. Abolpour, B., Afsahi, M.M., Hosseini, S.G. Statistical analysis of the effective factors on the 28 days compressive strength and setting time of the concrete. Journal of Advanced Research. 2015. 6(5). Pp. 699–709. DOI: 10.1016/j.jare.2014.03.005
4. Abd Elaty, M.A.A. Compressive strength prediction of Portland cement concrete with age using a new model. HBRC Journal. 2014. 10(2). Pp. 145–155. DOI: 10.1016/j.hbrcj.2013.09.005
5. Amin, M.N., Iftikhar, B., Kaffayatullah, Kh., and Qadir, M.T. Analyzing the compressive strength, eco-strength, and cost–strength ratio of agro-waste-derived concrete using advanced machine learning methods. Reviews on Advanced Materials Science. 2025. 64(1). Article no. 20250091. DOI: 10.1515/rams-2025-0091
6. Ramseyer, C.C., Kiamanesh, R. Optimizing Concrete Mix Designs to Produce Cost Effective Paving Mixes. Oklahoma Department of Transportation. 2009. 122 p.
7. Zheng, W., Zheng, W., Shui, Zh., Xu, Zh., Gao, Xu, Zhang, Sh. Multi-objective optimization of concrete mix design based on machine learning. Journal of Building Engineering. 2023. 76. Article no. 107396. DOI: 10.1016/j.jobe.2023.107396
8. Tipu, R.K., Panchal, V., Pandya, K. Multi-objective optimized high-strength concrete mix design using a hybrid machine learning and metaheuristic algorithm. Asian Journal of Civil Engineering. 2023. 24(3). Pp. 849–867. DOI: 10.1007/s42107-022-00535-8
9. Wang, M., Wang, M., Du, M., Zhuang, X., x Lv, X., Wang, Ch., Zh. Multi-objective optimization of ultra-high performance concrete based on life-cycle assessment and machine learning methods. Frontiers of Structural and Civil Engineering. 2025. 19(1). Pp. 143–161. DOI: 10.1007/s11709-025-1152-0
10. Zhang, F., Wen, B., Niu, D., Li, A., & Guo, B. Optimized Design of Low-Carbon Mix Ratio for Non-Dominated Sorting Genetic Algorithm II Concrete Based on Genetic Algorithm-Improved Back Propagation. Materials. 2024. 17(16). Article no. 4077. DOI: 10.3390/ma17164077
11. DeRousseau, M., Kasprzyk, J., Srubar III, W. Multi-objective optimization methods for designing low-carbon concrete mixtures. Frontiers in Materials. 2021. 8. Article no. 680895. DOI: 10.3389/fmats.2021.680895
12. Wei, J., Zhang, H., Yang, Y., Zhang W., Liu, X. Multi-objective optimization of compressive strength and slump in MPCM-integrated concrete using machine learning. Materials Today Communications. 2025. 46. Article no. 112619. DOI: 10.1016/j.mtcomm.2025.112619

13. Gu, Y., Fan, R., Li, Y., Zhao, J., Song, Z., Chu, H. Multi-Objective Optimization for Nano-Silica-Modified Concrete Based on Explainable Machine Learning. Nanomaterials. 2025. 15(18). Article no. 1423. DOI: 10.3390/nano15181423
14. Chen, F., Xu, W., Wen, Q., Zhang, G., Xu, L., Fan, D., Yu, R. Advancing concrete mix proportion through hybrid intelligence: A multi-objective optimization approach. Materials. 2023. 16(19). Article no. 6448. DOI: 10.3390/ma16196448
15. Fan, M., Li, Y., Shen., J., J., Kaikai, Shi, J. Multi-objective optimization design of recycled aggregate concrete mixture proportions based on machine learning and NSGA-II algorithm. Advances in Engineering Software. 2024. 192. Article no. 103631. DOI: 10.1016/j.advengsoft.2024.103631
16. Zhang, J., Huang, Y., Wang, Y., Ma, G. Multi-objective optimization of concrete mixture proportions using machine learning and metaheuristic algorithms. Construction and Building Materials. 2020. 253. Article no. 119208. DOI: 10.1016/j.conbuildmat.2020.119208
17. Tipu, R.K., Rathi, P., Pandya, K., Panchal, V.R. Optimizing sustainable blended concrete mixes using deep learning and multi-objective optimization. Scientific Reports. 2025. 15(1). Article no. 16356. DOI: 10.1038/s41598-025-00943-1
18. Paudel, S., Pudasaini, A., Shrestha, R.K., Kharel, E. Compressive strength of concrete material using machine learning techniques. Cleaner Engineering and Technology. 2023. 15. Article no. 100661. DOI: 10.1016/j.clet.2023.100661
19. Abbas, Y.M., Khan, M.I. Robust Machine Learning Framework for Modeling the Compressive Strength of SFRC: Database Compilation, Predictive Analysis, and Empirical Verification. Materials. 2023. 16(22). Article no. 7178. DOI: 10.3390/ma16227178
20. Oviedo, A.I., Londoño, J.M., Vargas, J.F., Zuluaga, C., Gómez, A. Modeling and Optimization of Concrete Mixtures Using Machine Learning Estimators and Genetic Algorithms. Modelling. 2024. 5(3). Article no. 642–658. DOI: 10.3390/modelling5030034
21. Zuo, S., Liu, B. Optimization design of concrete mix proportion based on support vector machine regression and enhanced genetic algorithm. Discover Applied Sciences. 2025. 7(3). Article no. 200. DOI: 10.1007/s42452-025-06603-3
22. Yeh, I.-C. Design of high-performance concrete mixture using neural networks and nonlinear programming. Journal of Computing in Civil Engineering. 1999. 13(1). Pp. 36–42.
23. Ma, M., Tam, V.W., Le, Kh.N., Osei-Kyei, R. Analysing the impacts of key factors on the price of recycled concrete: A system dynamics model. Journal of Building Engineering. 2023. 80. Article no. 108123. DOI: 10.1016/j.jobe.2023.108123
24. Khan, A., Do, J., Kim, D. Cost effective optimal mix proportioning of high strength self compacting concrete using response surface methodology. Computers and Concrete. 2016. 17(5). Pp. 629–638. DOI: 10.12989/cac.2016.17.5.629
25. Cavusoglu, I. Superplasticizer Dosage Effect on Strength, Microstructure and Permeability Enhancement of Cementitious Paste Fills. Minerals. 2024. 14(12). Article no. 1242. DOI: 10.3390/min14121242
26. Alsadey, S., Omran, A. Effect of superplasticizers to enhance the properties of concrete. Design, Construction, Maintenance. 2022. 2. Pp. 84–91. DOI: 10.37394/232022.2022.2.13
27. Asteris, P.G., et al. AI-powered GUI for prediction of axial compression capacity in concrete-filled steel tube columns. Neural Computing and Applications. 2024. 36(35). Article no. 22429–22459. DOI: 10.1007/s00521-024-10405-w
28. Li, H., Lin, J., Lei, X., Wei, T. Compressive strength prediction of basalt fiber reinforced concrete via random forest algorithm. Materials Today Communications. 2022. 30. Article no. 103117. DOI: 10.1016/j.mtcomm.2021.103117
29. Holand, J.H. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. The University of Michigan Press. Ann Arbor, MI, 1975. 232 p.
30. Nukah, P.D., Abbey, S.J., Booth, C.A. Development of a learner model tool for predicting strength and embodied carbon for lightweight concrete production. Journal of Building Engineering. 2024. 95. Article no. 110330. DOI: 10.1016/j.jobe.2024.110330
31. Abdennaji, T.S., Tipu, R.K., Alassaf, Y. Predicting compressive and tensile strength of concrete with different sand types using machine learning. Ain Shams Engineering Journal, 2025. 16(8). Article no. 103474. DOI: 10.1016/j.asej.2025.103474
32. Golafshani, E., Khodadadi, N., Ngo, T., Nanni, A., Behnood, A. Modelling the compressive strength of geopolymer recycled aggregate concrete using ensemble machine learning. Advances in Engineering Software. 2024. 191. DOI: 10.1016/j.advengsoft.2024.103611
33. Fathy, I.N., Dahish, H.A., Alkharisi, M.K., Mahmoud, A.A., Fouad, H.E.E. Predicting the compressive strength of concrete incorporating waste powders exposed to elevated temperatures utilizing machine learning. Scientific Reports. 2025. 15(1). Article no. 25275. DOI: 10.1038/s41598-025-11239-9
34. Shaaban, M., Amin, M., Selim, S., Riad, I.M. Machine learning approaches for forecasting compressive strength of high-strength concrete. Scientific Reports. 2025. 15(1). Article no. 25567. DOI: 10.1038/s41598-025-10342-1
35. Chou, J.-S., Chen, L.-Y., Liu, C.-Y. Forensic-based investigation-optimized extreme gradient boosting system for predicting compressive strength of ready-mixed concrete. Journal of Computational Design and Engineering. 2023. 10(1). Pp. 425–445. DOI: 10.1093/jcde/qwac133
36. Cihan, M.T., Cihan, P. Bayesian-Optimized Ensemble Models for Geopolymer Concrete Compressive Strength Prediction with Interpretability Analysis. Buildings. 2025. 15(20). Article no. 3667. DOI: 10.3390/buildings15203667
37. Philip, S., Nidhi, N. Compressive strength prediction and feature analysis for GGBS-Based geopolymer concrete using optimized XGBoost and SHAP: A comparative study of optimization algorithms and experimental validation. Journal of Building Engineering. 2025. 108. Article no. 112879. DOI: 10.1016/j.jobe.2025.112879
38. Khan, M.I., Abbas, Y.M., Fares, G., Alqahtani, F. Strength prediction and optimization for ultrahigh-performance concrete with low-carbon cementitious materials – XG boost model and experimental validation. Construction and Building Materials. 2023. 387. Article no. 131606. DOI: 10.1016/j.conbuildmat.2023.131606
39. Tabani, A., Sharma, A., Biswas, R., Sivenas, T., Asteris, P.G. Revealing the nature of Ultra-High-Performance concrete using computational intelligence. Construction and Building Materials. 2025. 492. Article no. 143082. DOI: 10.1016/j.conbuildmat.2025.143082
40. Al-Naghi, A.A., Ahmad, A., Amin, M.N., Algassem, O., Alnawmasi, N. Sustainable optimisation of GGBS-based concrete: De-risking mix design through predictive machine learning models. Case Studies in Construction Materials. 2025. 23. Article no. e04900. DOI: 10.1016/j.cscm.2025.e04900

41. Cakiroglu, C., Batool, F., Sangi, A.J., Fatima, B., Nehdi, M.L. Explainable machine learning predictive model for mechanical strength of recycled ceramic tile-based concrete. Materials Today Communications. 2025. 44. Article no. 112139. DOI: 10.1016/j.mtcomm.2025.112139
42. Ehsan, K., Mohamed, A.H., Inqiad, W.B., Javed, M.A., Iqbal, I. Multi expression programming and interpretable machine learning for determining compressive strength of coral sand aggregate concrete. Materials Today Communications. 2025. 45. Article no. 112370. DOI: 10.1016/j.mtcomm.2025.112370
43. Hamed, A.K., Elshaarawy, M.K., Alsaadawi, M.M. Stacked-based machine learning to predict the uniaxial compressive strength of concrete materials. Computers and Structures. 2025. 308. Article no. 107644. DOI: 10.1016/j.compstruc.2025.107644
44. Abushanab, A., Vimonsatit, V. Compressive strength, flexural strength, and slump of recycled aggregate fibre-reinforced fly ash concrete using explainable extreme gradient boosting machine learning model with prediction tool. Powder Technology. 2025. 469(1). Article no. 121710. DOI: 10.1016/j.powtec.2025.121710
45. Alahmari, T.S., Arif, K. Machine learning approaches to predict the strength of graphene nanoplatelets concrete: Optimization and hyper tuning with graphical user interface. Materials Today Communications. 2024. 40. Article no. 109946. DOI: 10.1016/j.mtcomm.2024.109946
46. Kapil, A., Jadda, K., Jee, A.A. Developing machine learning models to predict the fly ash concrete compressive strength. Asian Journal of Civil Engineering. 2024. 25(7). Article no. 5505–5523. DOI: 10.1007/s42107-024-01125-6
47. Javed, M.F., Fawad, M., Lodhi, R., Najeh, T., Gamil, Y. Forecasting the strength of preplaced aggregate concrete using interpretable machine learning approaches. Scientific Reports. 2024. 14(1). Article no. 8381. DOI: 10.1038/s41598-024-57896-0
48. Khatoon, S., K, K.A., Sapkota, S.C. Experimental insights and hybridized ensemble machine learning validation of fiber reinforced geopolymer concrete strength. Asian Journal of Civil Engineering. 2026. 27. Pp. 1289–1312. DOI: 10.1007/s42107-025-01562-x
49. Li, T., Yang, J., Jiang, P., Abuhussain, M.A., Zaman, A., Fawad, M., Farooq, F. Forecasting the strength of nanocomposite concrete containing carbon nanotubes by interpretable machine learning approaches with graphical user interface. Structures. 2024. 59. Article no. 105821. DOI: 10.1016/j.istruc.2023.105821
50. Liu, X., Mei, Sh., Wang, X., Li, X. Estimation of compressive strength of concrete with manufactured sand and natural sand using interpretable artificial intelligence. Case Studies in Construction Materials. 2024. 21. Article no. e03840. DOI: 10.1016/j.cscm.2024.e03840
51. Nguyen, N.H., Abellán-García, J., Lee, S., Garcia-Castano, E, Vo. Th.P. Efficient estimating compressive strength of ultra-high performance concrete using XGBoost model. Journal of Building Engineering. 2022. 52. Article no. 104302. DOI: 10.1016/j.jobe.2022.104302
52. Sun, Y. Explainable Prediction of Compressive Strength and Elastic Modulus for Concrete Containing Waste Foundry Sand Using Bayesian-Optimized XGBoost with 10-Fold Cross-Validation. Journal of Sustainable Metallurgy. 2024. 10(1). Pp. 335–359. DOI: 10.1007/s40831-024-00790-w
53. Singh, S., Patro, S.K., Parhi, S.K. Evolutionary optimization of machine learning algorithm hyperparameters for strength prediction of high-performance concrete. Asian Journal of Civil Engineering. 2023. 24(8). Pp. 3121–3143. DOI: 10.1007/s42107-023-00698-y
54. Sun, Y. Estimation of compressive strength for spiral stirrup-confined circular concrete column using optimized machine learning with interpretable techniques. Mechanics of Advanced Materials and Structures. 2024. 31(28). Pp. 10839–10858. DOI: 10.1080/15376494.2023.2298232
55. Ullah, A., Yang,Y., Ullah, W., Ayub, B., Alzlfawi, A., Iqbal, I. Toward transparent AI: Predicting strength of fly ash foam composite concrete using explainable ML models. Structural Concrete. 2025. 27(1). Pp. 595–624. DOI: 10.1002/suco.70302

Information about the authors:

***Mohsen Beigi**,*
ORCID: https://orcid.org/0000-0002-0644-2657
E-mail: mohsenbeigi59@gmail.com

Received 29.09.2025. Approved after reviewing 05.02.2026. Accepted 05.02.2026.