Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi

Pamukkale University Journal of Engineering Sciences

Optimization of hyper parameters of artificial neural networks for prediction of elastic modulus of normal and high strength concrete

Normal ve yüksek dayanımlı betonların elastisite modülünün tahmini için yapay sinir ağlarının hiper parametrelerinin optimizasyonu

Fatih Ahmet Şenel[1*]

[1]Department of Computer Engineering, Faculty of Engineering and Natural Sciences, Süleyman Demirel University, Isparta, Türkiye.
fatihsenel@sdu.edu.tr

Received/Geliş Tarihi: 27.08.2024 Revision/Düzeltme Tarihi: 03.04.2025 doi: 10.5505/pajes.2025.20265
Accepted/Kabul Tarihi: 23.06.2025 Research Article/Araştırma Makalesi

Abstract

In this study, while modeling the concrete elasticity modulus with Artificial Neural Networks (ANN), the optimal determination of the parameters of ANNs was carried out with the help of meta-heuristic algorithms. The hyperparameters of ANNs are the number of hidden layers, the number of neurons in hidden layers, and the activation functions in hidden layers. ANNs have been successfully used in classification and regression problems. But determining hyperparameters is a time-consuming process. Therefore, in this study, hyperparameters were determined using meta-heuristic algorithms. Whale Optimization Algorithm, Ant Lion Optimizer and Particle Swarm Optimization algorithms were used because they are successful in solving many engineering problems. The elastic modulus of normal and high strength concrete was estimated using ANN, whose hyperparameters were determined. The results obtained were compared with previous studies in literature. The proposed method outperformed the previous methods by showing better or equal results in most experiments. In the training process, for high strength concrete, it was more successful in 44.9%, equal in 34.8% and less successful in 20.3%. Overall, it performed equal to or better than the previous methods in 79.7% of the training process and 76.4% in the testing process. For normal strength concrete, the proposed method performed better or equal in 59.6% of the training process and 69.2% of the testing process, proving its effectiveness in both cases. As a result, better modeling results were obtained than in previous studies. As a result of modeling with different datasets, the R^2 value was found to be the highest 0.98. It has been shown that better results can be obtained from ANN used without tuning the hyperparameter.

Keywords: Artificial neural network, Elastic modulus, Optimization, Meta-Heuristic

Öz

Bu çalışmada, beton elastisite modülü Yapay Sinir Ağları (YSA) kullanılarak modellenmiştir. YSA'ın yapısal parametrelerinin optimum olarak belirlenmesi ise meta-sezgisel algoritmalar yardımıyla gerçekleştirilmiştir. YSA'ların hiper parametreleri; gizli katman sayısı, gizli katmanlardaki nöron sayıları ve gizli katmanlarda kullanılan aktivasyon fonksiyonlarıdır. YSA, sınıflandırma ve regresyon problemlerinde başarılı sonuçlar elde edebilen bir yöntemdir. Ancak hiper parametrelerinin belirlenmesi zaman alıcıdır. Bu nedenle bu çalışmada meta-sezgisel algoritmalar kullanılarak hiper parametreler belirlenmiştir. Birçok mühendislik probleminin çözümünde Balina Optimizasyon Algoritması, Karınca Aslanı Optimizasyonu ve Parçacık Sürü Optimizasyon algoritmaları başarılı sonuçlar elde edebildikleri için bu çalışmada tercih edilmişlerdir. Normal ve yüksek dayanımlı betonların elastisite modülü, hiper parametreleri belirlenen YSA kullanılarak tahmin edilmiş ve elde edilen sonuçlar literatürdeki önceki çalışmalarla karşılaştırılmıştır. Önerilen yöntem, çoğu deneyde daha iyi veya eşit sonuçlar göstererek önceki yöntemlerden daha iyi performans göstermiştir. Eğitim aşamasında, yüksek dayanımlı beton için %44,9'nda daha başarılı, %34,8'nde aynı başarıyı göstermiş ve %20,3'nde ise daha az başarılı olmuştur. Genel olarak, eğitim aşamasının %79,7'sinde önceki yöntemlere eşit veya daha iyi, test aşamasında ise %76,4 başarı göstermiştir. Normal dayanımlı beton için amaçlanan yöntem, eğitim aşamasının %59,6'sında ve test aşamasının %69,2'sinde daha iyi veya aynı performansı göstermiş ve her iki durumda da etkinliğini kanıtlamıştır. Sonuç olarak önceki çalışmalara göre daha iyi modelleme sonuçları elde edilmiştir. Farklı veri kümeleri ile yapılan modelleme sonucunda R^2 değeri en yüksek 0.98 olarak bulunmuştur. Hiper parametreler bulunmadan kullanılan YSA'dan daha iyi sonuçlar elde edilebileceği gösterilmiştir.

Anahtar kelimeler: Yapay sinir ağı, Elastisite modülü, Optimizasyon, Meta-Sezgisel

1 Introduction

The Elastic Modulus (EM) is the measure of the elastic deformation of the material under force. Due to the force applied to a material, the shape of the material changes. The ability of the material to return to its original shape after this applied force is removed is expressed as EM. The EM is calculated by taking a sample under ideal conditions and measuring it. However, measurements made by taking samples are not always practical. Therefore, with the developing computer technology, a surrogate model can be created for EM. With these surrogate models, predictions can be made for the necessary calculations without causing any physical damage by taking samples. When the literature is examined, there are many studies for predictive modeling of the EM of concrete. Demir [1] aimed a model that predicts elastic modulus of normal and high strength concrete using Artificial Neural Networks (ANN). He aimed to develop the best prediction model by using different ANN architectures. He tested the method with different datasets and compared the results. It was able to make high-accuracy predictions with the ANN method. However, while tuning the ANN architecture, time wasted due to manual adjustment. Farooq et al. made a modeling for prediction the compressive strength of high strength concrete in their study [2]. They developed a method using random forest and gene expression programming methods together.

*Corresponding author/Yazışılan Yazar

Farooq et al. tested their methods with datasets consisting of 357 pieces of data. They also compared their developed methods with ANN and decision tree. The random forest method best modeled the compressive strength of high strength concrete. García et al. developed a model that predicts compressive strengths and slump flow parameters using the ANN method [3]. They used a total of 265 data, using 200 data for training and 65 data for testing. The ANN method was analyzed with different numbers of hidden layers and neurons. They achieved the best modeling result by using a low number of hidden layers and neurons to avoid over-fitting. Abdulla developed a model with ANN to predict the axial compression capacity and axial strain of concrete-filled plastic tubular specimens [4]. The ANN architecture he uses has four neurons in the input layer. These are the water-cement ratio, aggregate cement ratio, slump, and the bias. There is also a single hidden layer and an output layer consisting of a single neuron. He selected six-neuron hidden layer for strain models, and seven-neuron hidden layer for force models. When the studies are examined, it is seen that the hidden layer numbers and neuron numbers of the ANN models used are determined by the user without any optimization. Manually tuning the ANN architecture is a time-consuming process. In addition, it is not always possible to set the best architecture. At this stage, tuning the control parameters of optimal prediction models using developing soft computing algorithms will be the best selection for obtaining accurate results.

There are analytical methods developed for calculation of elastic modulus of strength concrete besides soft computing algorithms. Analytical formulas used in the literature to express the relationship between compressive strength and elastic modulus are shown in Equations 1, 2, 3, 4 and 5. Equations 1 and 2 refer to normal strength concrete, while Equations 3, 4 and 5 refer to high strength concrete.

$$E_c = 4.73(f_c)^{1/2}, \text{ACI 318-95} \quad (1)$$

$$E_c = 3.25(f_c)^{1/2} + 14, \text{TS-500} \quad (2)$$

$$E_c = 3.32(f_c)^{1/2} + 6.9, \text{ACI 363} \quad (3)$$

$$E_c = 10(f_c + 8)^{1/3}, \text{CEB90} \quad (4)$$

$$E_c = 9.5(f_c)^{0,3}, \text{NS 3473} \quad (5)$$

These methods are also frequently used in studies in literature. However, it cannot give as successful results as soft computing algorithms. Soft computing algorithms are used in civil engineering as well as in solving problems in many other fields. Al-Gburi et al. developed a neural network model for concrete compressive strength prediction [5]. They used 13 input parameters. They determined the optimum neural network structure with different neural network structures and different training-test ratios. Zeng et al. proposed a model that predicts the 28-day compressive strength of concrete using constituent material information parameters [6]. They generated a dataset of 380 groups of concrete mixes. They compared the success of their proposed deep neural network model with other known modeling methods for testing. The results show that the intended model is successful. Mohamad Ali et al. developed a model that predicts the compressive strength of concrete using three different ANN models [7]. They developed an ANN model with eight inputs. Their model is consisting of cement, blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, fine aggregate, and age inputs. They showed that concrete compressive strength can be modeled successfully with ANN.

When the studies are examined, it has been observed that artificial neural network-based methods give successful results. However, determining the structure of the ANN is important to obtaining successful results at the desired level. Determining the optimum neural network structure manually is difficult. Early studies on hyperparameter tuning were generally performed by manual tuning. However, due to the complexity of the ANN architecture, the applicability of this method has been difficult. Later, random search techniques were found to be more effective in hyperparameter tuning, paving the way for the use of meta-heuristic algorithms [8]. This modeling method, known as hyperparameter optimization, has been used in this study and hyperparameter optimization method is used in many different fields today [9], [10], [11], [12], [13], [14], [15], [16], [17], [18]. Notably, the application of these techniques in the context of predicting concrete properties is highlighted by Haseli et al. [19]. Their work on forecasting strength parameters of concrete using optimized ANN approaches reported minimal discrepancies between predicted and experimental values, underscoring the practical significance of hyperparameter tuning in reducing prediction errors. Franchini [20] introduced a “green” neural architecture search method that minimizes the environmental impact of extensive hyperparameter sweeps—an increasingly important consideration as ANN-based approaches are scaled in practical engineering applications. In this study, three different optimization algorithms have been used to determine the optimum ANN structure. These algorithms are Whale Optimization Algorithm (WOA) [21], Ant Lion Optimizer (ALO) [22] and Particle Swarm Optimization (PSO) [23] algorithms.

The used optimization and ANN methods are given in the Section 2. Section 3 is experimental results, and the last section is the conclusion of the study.

2 Material and Methods

The dataset for the model to be developed in this study were obtained from studies [24], [25], [26], [27]. This dataset consists of 156 experimental data. Since there are studies on this dataset, I chose this dataset. Thus, the success of the model could be discussed. In this study, ANN was used for modelling. The ANN's hyperparameters are tunned with three meta-heuristic algorithms. The ANN and these algorithms are detailed below.

2.1 Artificial Neural Network

Artificial Neural Network (ANN) is a supervised learning method that establishes a relationship between input and output. There are layers and neurons in the ANN. These layers consist of three groups as input, hidden and output layers (Figure 1). The number of neurons in the input layer is equal to the number of features. The number of neurons in the output layer is equal to the output number of the dataset. The number of hidden layers and the number of neurons are important parameters of the ANN that need to be adjusted. The hidden layer structure differs according to the problem.

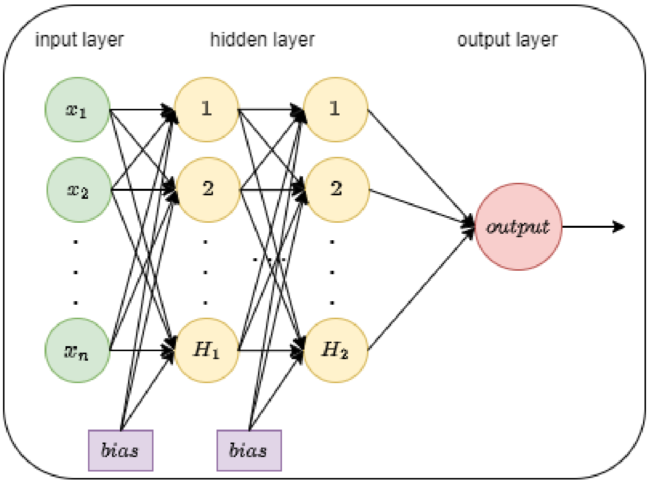

Figure 1. ANN architecture; layers and neurons

The neurons in each layer are multiplied by a weight coefficient, then summed and passed through an activation function. Different activation functions can be preferred. Just as the hidden layer structure needs to be adjusted, the activation function is also a hyperparameter that needs to be adjusted.

2.2 Whale Optimization Algorithm

Whale Optimization Algorithm (WOA) is a meta-heuristic optimization algorithm inspired by the hunting techniques of humpback whales [21]. Humpback whales create a bubble net to hunt small schools of fish. So, they create a bubble barrier that prevents small fish from escaping. They, also, hide themselves with this barrier. Figure 2 illustrates this behavior.

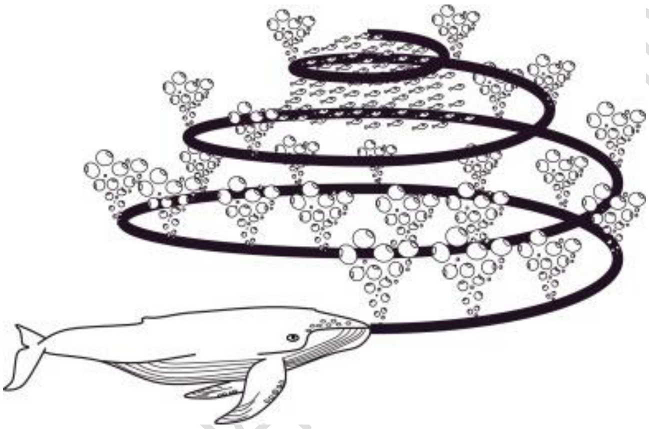

Figure 2. Hunting behavior of humpback whales [21]

The WOA is a three-step algorithm. These are encircling, bubble-net attacking and searching for prey steps.

Humpback whales encircle their prey to identify their location. Other whales update their positions considering the position of the whale with the best fitness. The positions of the whales are updated at each iteration. And the update is modeled according to Equations 6 and 7.

$$\vec{D} = |\vec{C} \cdot \vec{X} * (t) - \vec{X}(t)| \tag{6}$$

$$\vec{X}(t+1) = \vec{X^*}(t) - \vec{A} \cdot \vec{D} \tag{7}$$

where t is the current iteration, X^* is the best whale position vector, and A and C are the coefficient vectors.

Equations (8) and (9) show the calculation of A and C coefficients.

$$\vec{A} = 2 \cdot \vec{a} \cdot \vec{r} - \vec{a} \tag{8}$$

$$\vec{C} = 2 \cdot \vec{r} \tag{9}$$

Here, r is the random number in the range of [0,1] and a is the coefficient decreasing from 2 to 0 as the iteration progresses. In WOA, humpback whales attack their prey with two methods: Shrinking encircling and Spiral position update. The shrinking encircling method is performed by decreasing the number a. The spiral position update method is operated according to Equation 10.

$$\vec{X}(t+1) = \vec{D'} \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X^*}(t) \tag{10}$$

Here, vector D' represents the distance between X^* and $X(t)$. l is the random number in the range of $[-1,1]$. b is a constant for the formation of the spiral shape. The WOA algorithm decides with equal probability which of the Shrinking encircling and spiral position update methods to use. This probability procedure is expressed in Equation 11.

$$\vec{X}(t+1) = \begin{cases} \vec{X^*}(t) - \vec{A} \cdot \vec{D} & , p < 0.5 \\ \vec{D'} \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X^*}(t) & , p \geq 0.5 \end{cases} \tag{11}$$

p is the random number in the range of [0,1].

2.3 Ant Lion Optimizer

Ant Lion Optimizer (ALO) has been developed inspired by the hunting strategies of ant lions during their larval stages [22]. The ALO algorithm is a meta-heuristic algorithm developed based on the ant lion larva digging a cone-shaped hole with its large jaw. The ant lion hides at the bottom of the cone-shaped pit and waits for the ants to fall into the pit. Figure 3 shows the trap dug by the ant lion. The trapped ants easily slide towards the bottom of the pit. Because the trap has slippery and sharp edges. Ants are struggling to get out. The ant lion, on the other hand, throws sand from the bottom to the top of the pit when an ant falls into the trap, and slides the feet of the ants. Eventually, the ants fall to the bottom and are hunted by the ant lion.

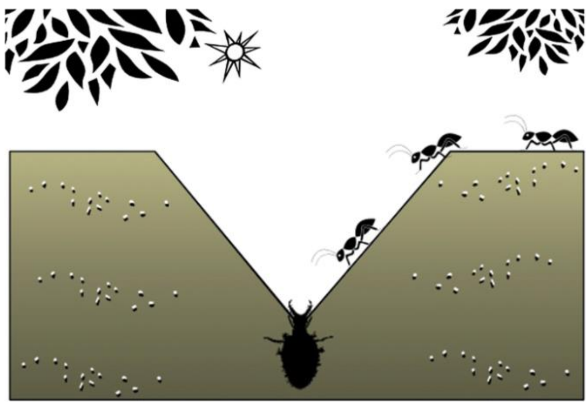

Figure 3. Ant lion's pit and ants [22]

The ALO algorithm starts with the random walks of the ants. In Equations 12, 13 and 14, the random walks of the ants are expressed mathematically.

$$X(t) = [0, c(2r(t_1) - 1), c(2r(t_2) - 1), \dots, c(2r(t_n) - 1)] \tag{12}$$

$$r(t) = \begin{cases} 1 & rand(0,1) > 0.5 \\ 0 & rand(0,1) \leq 0.5 \end{cases} \tag{13}$$

$$X_i^t = \frac{(X_i^t - a_i) \times (d_i^t - c_i^t)}{(b_i - a_i)} + c_i^t \tag{14}$$

where, n represents the maximum number of iterations, t represents the steps of random walks, function c represents the cumulative sum, and $r(t)$ represents random walks. c_i^t and d_i^t values represent the minimum and maximum values of the i^{th} dimension at t^{th} iteration, respectively. Ants are prevented from leaving the search space with Equation 14. The sliding of ants towards the bottom of the trap is modeled in Equation 15 and 16.

$$c_i^t = Antlion^t + c^t, d_i^t = Antlion^t + d^t \tag{15}$$

$$c^t = \frac{c^t}{J}, d^t = \frac{d^t}{J} \tag{16}$$

$Antlion^t$ represents the selected ant lion at the current iteration. J sliding ratio is calculated according to Equation 17.

$$J = \begin{cases} 1 + 10^2 \frac{t}{T_{max}}, & 0.1T_{max} < t \leq 0.5T_{max} \\ 1 + 10^3 \frac{t}{T_{max}}, & 0.5T_{max} < t \leq 0.75T_{max} \\ 1 + 10^4 \frac{t}{T_{max}}, & 0.75T_{max} < t \leq 0.9T_{max} \\ 1 + 10^5 \frac{t}{T_{max}}, & 0.9T_{max} < t \leq 0.95T_{max} \\ 1 + 10^6 \frac{t}{T_{max}}, & 0.95T_{max} < t \leq T_{max} \\ 1, & otherwise \end{cases} \tag{17}$$

The positions of the ants are calculated according to Equations 18 and 19.

$$Ant_i^t = \frac{R_A + R_E}{2} \tag{18}$$

$$\text{Antlion}^t = \begin{cases} Ant_i^t & f(Ant_i^t) < f(\text{Antlion}^t) \\ \text{Antlion}^t & otherwise \end{cases} \tag{19}$$

R_A and R_E represent an ant lion and the elite ant lion in the population selected by roulette wheel method, respectively.

2.4 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a meta-heuristic algorithm inspired by the social behavior of bird flocking or fish schooling while searching for food [23]. In the PSO algorithm, the particles in the population have position and velocity vectors. The positions of the particles are updated, considering the best positions of the particles themselves and the best position obtained during all iterations at each iteration step. Updates of particles in the population are expressed in Equations 20 and 21.

$$\vec{V}(t+1) = w * \vec{V}(t) + c_1 * r_1 * \left(\vec{P}(t) - \vec{X}(t)\right) + c_2 * r_2 * \left(\vec{P}^g(t) - \vec{X}(t)\right) \tag{20}$$

$$\vec{X}(t+1) = \vec{X}(t) + \vec{V}(t+1) \tag{21}$$

Here t is iteration. X and V are the positions and velocities of the particles, respectively. $w = 0.7298$, $c_1 = c_2 = 1.49445$ are constant coefficients [28] while r_1, r_2 represent random numbers in the range of [0,1].

2.5 Cost Function

In this study, meta-heuristic algorithms are used to optimize the hyperparameters of the ANN. The fitness values of the individuals in the population of the optimization algorithms are calculated by the prediction success of the candidate ANN structure. The prediction success of the ANN was measured with the Mean Absolute Error (MAE) metric. The MAE calculation formula is given in Equation 22.

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |y - \hat{y}| \tag{22}$$

Here N is the number of samples. y is the actual value and $\hat{y}$ is the predicted value. As the MAE value decreases to 0, the prediction success of the ANN increases.

3 Experimental Results

In this study, a dataset consisting of experimental data of the concrete elasticity modulus obtained from different studies was used. There are 156 experimental samples in this dataset. There is a study that models this dataset using ANN [1]. However, different ANN architectures without hyperparameter optimization have been tried. In this study, hyperparameter optimization was performed and the modeling results were compared with previous studies. The ANN consists of 1 to 3 hidden layers, with each layer having up to 10 neurons. The activation function for each layer is selected from 'tansig', 'logsig', or 'purelin'. These hyperparameters are optimized using a meta-heuristic algorithm to improve the model's performance. Table 1 shows the ranges of the parameters tuned.

Table 1. Hyperparameter ranges for ANN architecture

Parameter	Range
The Number of Hidden Layers	1 - 3
Neurons Per Layer	1 - 10
Activation Function	'tansig', 'logsig', 'purelin'

All experiments in this study were performed on a computer with the following specifications: Intel i7-10870H CPU, 16 GB RAM, 64-bit operating system. Additionally, the coding operations were carried out using MATLAB.

To ensure a fair comparison of all the models, the initial weight values and bias values of ANN were set to 0. ANN's learning rate was set to 0.01. By preventing random initialization, the comparison was made in an objective manner. Similarly, all three meta-heuristic algorithms used were run with a population size of 10 and for 30 iterations.

In this study, modeling was performed with high strength concrete and normal strength concrete data. From the data in Table 2~5, f_c(MPa) is used as the input parameter and E_c(GPa) is used as the output parameter of the model. The references from which the data were obtained are shown in the first column of the tables. The accuracy of the model's predictions was evaluated by calculating the ratio of the actual value to the predicted value. A ratio close to 1 indicates that the model's prediction is close to the actual value. When the ratio is less than 1, the model has under-predicted value, and when the ratio is greater than 1, the model has over-predicted value. A ratio equal to 1 signifies a perfect match between the predicted and actual values. The data in the table show the prediction ratios of the calculation methods in the literature and the prediction ratio of this study. In the modeling conducted in this study, the MAE value was used as the loss function. The goal was to minimize the MAE value as much as possible, aiming for a value close to 0. Additionally, the R^2 value was also considered to understand the success of the modeling. The pseudocode for the software developed for hyperparameter optimization applied in this study is provided in Figure 4.

The dataset is divided into two groups as training and testing. In order to make an objective comparison, the training-test data were divided as in previous studies. The high strength concrete estimation results are given for the training data in Table 2. In Table 3, the test data estimation results are presented.

```
Algorithm 1 Hyperparameter Optimization using Metaheuristic Algorithms
Define the problem parameters
Set number_of_layers = [1, 3]
Set neurons_per_layer_range = [1, 10]
Set activation_functions = {tansig, logsig, purelin}
Initialize Metaheuristic Algorithm Parameters
Set population_size = 10
Set max_iterations = 30
Set learning_rate = 0.01
Define Fitness Function (Error Measurement)
function FITNESS_FUNCTION(hyperparameters)
    Build the Neural Network with given hyperparameters
    network = build_network(hyperparameters)
    Train the neural network
    trained_network = train_network(network, train_data, train_labels, learning_rate)
    Predict using the trained network
    predictions = test_network(trained_network, test_data)
    Calculate the error metrics (MAE, R²)
    error = calculate_error(test_labels, predictions)
    return error
end function
Particle Swarm Optimization (PSO)
function PSO_OPTIMIZATION
    Initialize particles with random values
    for iteration = 1 to max_iterations do
        for each particle do
            Calculate the fitness of each particle using fitness_function
            Update particle's best-known position and global best position
            Update particle velocities and positions using PSO equations
        end for
        Update swarm's global best solution
    end for
    return best solution
end function
Ant Lion Optimizer (ALO)
function ALO_OPTIMIZATION
    Initialize antlions with random values
    for iteration = 1 to max_iterations do
        for each antlion do
            Calculate the fitness of each antlion using fitness_function
            Update the position of antlions using ALO equations
        end for
        Update the best solution found
    end for
    return best solution
end function
Whale Optimization Algorithm (WOA)
function WOA_OPTIMIZATION
    Initialize whales with random values
    for iteration = 1 to max_iterations do
        for each whale do
            Calculate the fitness of each whale using fitness_function
            Update position using WOA equations
        end for
        Update the best solution found
    end for
    return best solution
end function
Main Optimization Process
Set chosen_algorithm = "PSO" or "ALO" or "WOA"
if chosen_algorithm == "PSO" then
    best_hyperparameters = PSO_optimization()
else if chosen_algorithm == "ALO" then
    best_hyperparameters = ALO_optimization()
else if chosen_algorithm == "WOA" then
    best_hyperparameters = WOA_optimization()
end if
```

Figure 4. Pseudocode for the software used in hyperparameter optimization.

Table 2. The high strength concrete estimation results and comparisons for training data

Data	f_c(MPa)	E_c(GPa)	$E_{ACI\ 363}/E_c$	E_{CEB}/E_c	$E_{NS\ 3473}/E_c$	Demir [1]	This Study
[27]	63.2	41.8	0.80	0.99	0.79	1.02	0.98
[27]	70.2	43.0	0.81	0.99	0.79	1.02	0.97
[27]	65.1	41.5	0.81	1.01	0.80	1.03	0.99
[27]	70.5	40.4	0.86	1.06	0.84	1.09	1.04
[27]	71.5	41.4	0.84	1.04	0.83	1.07	1.02
[27]	63.6	42.6	0.78	0.97	0.78	1.00	0.96
[27]	85.9	45.0	0.84	1.01	0.80	1.01	0.99
[27]	90.2	44.4	0.87	1.04	0.83	1.03	1.01
[27]	85.9	44.3	0.85	1.03	0.82	1.03	1.00
[27]	81.2	43.9	0.84	1.02	0.81	1.03	1.01
[27]	88.1	44.5	0.86	1.03	0.82	1.03	1.00
[27]	81.6	43.8	0.84	1.02	0.81	1.03	1.01
[27]	84.8	47.2	0.79	0.96	0.76	0.96	0.94
[27]	85.6	45.6	0.82	1.00	0.79	1.00	0.97
[27]	96.2	46.6	0.85	1.01	0.80	1.02	0.98
[27]	46.4	35.2	0.84	1.08	0.85	1.04	1.02
[27]	73.9	41.6	0.85	1.04	0.83	1.07	1.03
[27]	87.6	44.5	0.85	1.03	0.82	1.03	1.00
[27]	93.1	45.4	0.86	1.03	0.82	1.05	1.00
[27]	95.3	45.2	0.87	1.04	0.82	1.05	1.02
[27]	102.1	46.1	0.88	1.04	0.83	1.03	1.01
[27]	102.8	46.7	0.87	1.03	0.82	1.02	1.01
[27]	106.3	48.4	0.85	1.00	0.80	0.99	1.01
[27]	104.2	46.3	0.88	1.04	0.83	1.03	1.03
[27]	94.6	47.3	0.83	0.99	0.79	1.00	0.97
[27]	94.0	46.3	0.84	1.01	0.80	1.03	0.99
[27]	96.6	46.5	0.85	1.01	0.80	1.02	0.98
[27]	91.5	45.9	0.84	1.01	0.80	1.03	0.98
[27]	91.7	46.0	0.84	1.01	0.80	1.03	0.98
[27]	119.9	49.1	0.88	1.03	0.81	1.00	1.00
[27]	125.6	50.9	0.87	1.00	0.80	1.00	0.91
[24]	77.2	47.1	0.77	0.93	0.74	0.95	1.02
[24]	66.5	46.8	0.73	0.90	0.72	0.92	1.01
[24]	70.7	47.3	0.74	0.91	0.72	0.93	1.02
[24]	61.8	45.4	0.73	0.91	0.72	0.93	0.99
[24]	68.9	47.6	0.72	0.89	0.71	0.92	1.01
[24]	62.2	45.4	0.73	0.91	0.72	0.93	0.99
[24]	75.8	43.0	0.83	1.02	0.81	1.04	1.12
[24]	67.7	48.2	0.71	0.88	0.70	0.90	0.99
[24]	53.6	46.2	0.68	0.85	0.68	0.86	0.97
[24]	92.9	46.4	0.84	1.00	0.80	1.02	1.05
[24]	94.0	48.3	0.81	0.97	0.77	0.98	1.02
[24]	97.7	47.0	0.85	1.01	0.80	1.01	1.06
[24]	102	48.8	0.83	0.98	0.78	0.98	1.02
[24]	86.2	47.1	0.80	0.97	0.77	0.97	0.97
[24]	87.9	43.0	0.88	1.06	0.85	1.06	1.06
[24]	82.7	45.4	0.82	0.99	0.79	1.00	1.01
[24]	79.1	44.7	0.81	0.99	0.79	1.01	1.07
[24]	86.6	46.1	0.82	0.99	0.79	0.99	0.99

[24]	85.5	44.3	0.85	1.02	0.81	1.03	1.03
[24]	91.1	46.8	0.82	0.99	0.79	1.01	1.01
[24]	96.7	53.2	0.74	0.89	0.70	0.89	0.93
[24]	91.2	49.3	0.78	0.94	0.75	0.96	0.96
[24]	83.8	45.9	0.81	0.98	0.78	0.99	0.99
[24]	87.1	47.7	0.79	0.96	0.76	0.96	0.96
[24]	93.2	46.2	0.84	1.01	0.80	1.03	1.06
[24]	86.9	46.1	0.82	0.99	0.79	0.99	0.99
[24]	90.7	48.1	0.80	0.96	0.76	0.98	0.98
[24]	89.5	47.6	0.80	0.97	0.77	0.96	0.97
[24]	87.8	45.4	0.84	1.01	0.80	1.01	1.01
[24]	95.2	50.8	0.77	0.92	0.73	0.94	0.98
[24]	92.2	50.0	0.78	0.93	0.74	0.95	0.97
[24]	97.6	49.3	0.81	0.96	0.76	0.97	1.01
[24]	87.5	48.5	0.78	0.94	0.75	0.94	0.94
[24]	80.4	43.2	0.85	1.03	0.82	1.04	1.09
[24]	86.5	44.2	0.85	1.03	0.82	1.03	1.03
[24]	83.9	44.3	0.84	1.02	0.81	1.03	1.03
[24]	80.9	44.6	0.82	1.00	0.80	1.01	1.05
[24]	85.7	45.1	0.83	1.01	0.80	1.01	1.01

Table 3. The high strength concrete estimation results and comparisons for test data

Data	f_c(MPa)	E_c(GPa)	$E_{ACI\ 363}/E_c$	E_{CEB}/E_c	$E_{NS\ 3473}/E_c$	Demir [1]	This Study
[27]	69.70	41.50	0.83	1.03	0.82	1.06	1.00
[27]	78.30	44.30	0.82	1.00	0.79	1.01	0.99
[27]	82.60	44.20	0.84	1.02	0.81	1.03	1.00
[27]	65.80	40.80	0.83	1.03	0.82	1.06	1.01
[27]	100.60	45.80	0.88	1.04	0.83	1.04	0.99
[27]	92.80	45.80	0.85	1.02	0.81	1.04	0.99
[27]	93.60	47.10	0.83	0.99	0.79	1.01	0.97
[24]	71.50	48.00	0.73	0.90	0.71	0.92	1.00
[24]	59.10	40.90	0.79	0.99	0.79	1.01	1.08
[24]	57.90	44.50	0.72	0.91	0.72	0.92	0.99
[24]	93.70	50.50	0.77	0.92	0.73	0.94	0.97
[24]	85.30	45.00	0.83	1.01	0.80	1.01	1.01
[24]	99.70	47.60	0.84	1.00	0.79	1.00	1.04
[24]	85.10	44.70	0.84	1.01	0.81	1.02	1.02
[24]	90.30	45.00	0.85	1.03	0.82	1.02	1.04
[24]	87.20	41.10	0.92	1.11	0.88	1.11	1.11
[24]	84.50	45.30	0.83	1.00	0.79	1.00	1.00

Table 4. The normal strength concrete estimation results and comparisons for training data

Data	f_c(MPa)	E_c(GPa)	$E_{ACI\ 318}/E_c$	$TS500/E_c$	Demir [1]	This Study
[26]	31.4	30.4	0.87	1.06	0.97	1.00
[26]	27.8	29.1	0.86	1.07	0.97	0.92
[26]	28.5	26.8	0.94	1.17	1.06	1.04
[26]	29.4	31.5	0.81	1.00	0.91	0.92
[26]	26.4	30.0	0.81	1.02	0.92	0.88
[26]	28.5	29.0	0.87	1.08	0.98	0.96
[26]	32.6	32.4	0.83	1.00	0.93	0.93
[26]	29.9	30.2	0.86	1.05	0.96	0.99
[26]	29.8	27.5	0.94	1.15	1.05	1.08
[26]	28.0	30.8	0.81	1.01	0.92	0.88

[26]	27.3	26.5	0.93	1.17	1.05	0.99
[26]	27.5	25.2	0.98	1.23	1.11	1.05
[26]	27.0	27.2	0.90	1.14	1.02	0.97
[26]	28.5	27.3	0.92	1.15	1.04	1.02
[26]	26.4	26.5	0.92	1.16	1.04	0.99
[26]	27.1	23.9	1.03	1.29	1.17	1.10
[26]	26.3	24.0	1.01	1.28	1.15	1.10
[26]	26.1	24.9	0.97	1.23	1.10	1.06
[26]	27.8	25.3	0.99	1.23	1.11	1.06
[26]	25.7	25.7	0.93	1.19	1.06	1.02
[26]	27.8	26.0	0.96	1.20	1.08	1.03
[26]	28.6	27.5	0.92	1.14	1.04	1.01
[26]	27.9	26.2	0.95	1.19	1.08	1.03
[26]	18.4	21.9	0.93	1.28	1.14	1.07
[26]	23.4	26.3	0.87	1.13	1.01	0.93
[26]	29.9	30.4	0.85	1.05	0.95	0.98
[26]	22.9	26.5	0.85	1.12	0.99	0.88
[26]	23.7	27.2	0.85	1.10	0.98	0.93
[26]	27.4	27.1	0.91	1.14	1.03	0.97
[26]	14.0	15.6	1.13	1.68	1.53	1.44
[26]	16.9	20.5	0.95	1.33	1.20	1.10
[25]	17.1	26.3	0.74	1.04	0.94	0.85
[25]	18.0	28.8	0.70	0.96	0.86	0.78
[25]	18.5	30.1	0.68	0.93	0.83	0.75
[25]	21.8	20.9	1.06	1.40	1.24	1.15
[25]	25.8	28.6	0.84	1.07	0.96	0.90
[25]	27.3	32.9	0.75	0.94	0.85	0.79
[25]	30.3	35.9	0.73	0.89	0.81	0.89
[25]	29.6	36.8	0.70	0.86	0.78	0.87
[25]	19.6	23.1	0.91	1.23	1.10	0.98
[25]	19.4	30.3	0.69	0.93	0.84	0.75
[25]	20.9	23.9	0.90	1.21	1.08	0.98
[25]	21.2	26.5	0.82	1.09	0.97	0.89
[25]	23.6	32.1	0.72	0.93	0.83	0.79
[25]	24.2	33.6	0.69	0.89	0.80	0.76
[25]	31.8	25.5	1.05	1.27	1.17	1.21
[25]	32.2	27.4	0.98	1.18	1.09	1.12
[25]	30.6	28.6	0.91	1.12	1.03	1.11
[25]	29.6	31.6	0.81	1.00	0.91	1.01
[25]	35.0	35.6	0.79	0.93	0.88	0.89
[25]	32.8	36.7	0.74	0.89	0.82	0.84
[25]	38.4	26.6	1.10	1.28	1.24	1.21
[25]	35.7	30.1	0.94	1.11	1.05	1.05
[25]	42.7	34.1	0.91	1.03	1.03	0.95
[25]	36.8	29.3	0.98	1.15	1.10	1.09
[25]	40.1	28.4	1.05	1.22	1.19	1.14
[25]	47.7	29.6	1.10	1.23	1.26	1.37

Table 5. The normal strength concrete estimation results and comparisons for test data

Data	f_c(MPa)	E_c(GPa)	$E_{ACI\ 318}/E_c$	$TS500/E_c$	Demir [1]	This Study
[26]	29.40	33.02	0.78	0.96	0.87	0.88
[26]	28.80	28.97	0.88	1.08	0.98	0.97
[26]	27.70	25.64	0.97	1.22	1.10	1.04
[26]	22.10	21.80	1.02	1.34	1.20	1.06
[26]	28.90	26.83	0.95	1.17	1.07	1.05
[26]	20.60	23.87	0.90	1.20	1.07	0.97
[26]	25.30	28.09	0.85	1.08	0.97	0.93
[25]	16.20	23.26	0.82	1.16	1.05	0.96
[25]	23.20	23.88	0.95	1.24	1.11	1.05
[25]	17.90	17.99	1.11	1.54	1.38	1.25
[25]	23.90	30.48	0.76	0.98	0.88	0.83
[25]	27.10	24.67	1.00	1.25	1.13	1.05
[25]	37.50	32.61	0.89	1.04	1.00	0.99

When the training process for high strength concrete estimation is examined, it will be seen that the proposed method is much more successful than previous studies. In 31 of 69 observations, the intended method is better. While it showed the same success in 24 of all the experiments, it fell behind in 14 of them. In other words, the intended method was the same or more successful in 79.7% of the experiments.

If the results obtained with the test data of the trained model are examined, it is seen that the developed model is more successful in more than half of the experiments. The model developed in 76.4% of the experiments has the same or better performance.

The developed model was performed for normal strength concrete estimation after high strength concrete estimation. The normal strength concrete estimation results are presented in Tables 4 and 5. Just as with high strength concrete estimation, the proposed model is also successful for normal strength concrete estimation. The better or the same results were obtained with this model in 59.6% of the train experiments. In addition, the better results were obtained in 69.2% of the test data.

The results of hyperparameter tuning using meta-heuristic algorithms are presented in Table 6. For high strength concrete, PSO achieved the best performance with the highest R^2 (0.98) and the lowest MAE (0.36), suggesting the model is highly accurate and reliable. For normal strength concrete, the performance of the three algorithms is very similar, with all achieving an R^2 of 0.75 and MAE values ranging between 1.59 and 1.64.

Based on the analyses, the best result was obtained with the PSO algorithm. MAE was used as the cost function during the optimization process. Figure 5 shows the change in MAE value over 30 iterations of PSO. Upon examining the graph, it can be seen that the optimum point is reached and the number of 30 iterations is sufficient.

Figure 6 presents the error graph of the neural network trained with the optimum hyperparameters over the epochs. In the MATLAB program, the neural network tool uses Mean Squared Error (MSE) as the loss function. The training process is automatically stopped when no further improvement is observed in the error. The training error graph, showing an R^2 value of 0.98 achieved through hyperparameter optimization, is displayed in Figure 6. This graph also confirms that the training process has been successfully completed.

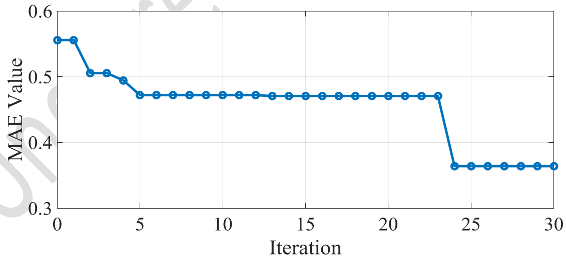

Figure 5. MAE values of PSO across iterations.

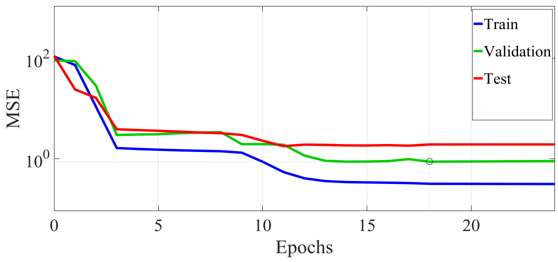

Figure 6. Change in MSE value during training of ANN with optimized hyperparameters.

The results indicate that the normal strength concrete estimation problem is more challenging compared to high strength concrete estimation. Despite using different meta-heuristic algorithms, the R^2 values remain relatively low, and the MAE values are relatively high. This suggests that the model may be underfitting the data, possibly due to limitations in the selected hyperparameter ranges. To improve the performance of the model for normal strength concrete, it would be beneficial to expand the search space for hyperparameters, such as increasing the number of neurons per layer. The results indicate that the use of three layers generally yields the best performance across all algorithms for both high strength concrete and normal strength concrete estimation. This suggests that a deeper network structure is more effective in capturing complex relationships within the data. Therefore, employing a three-layer architecture seems to be the most effective approach for this study, highlighting the importance of depth in neural network performance for concrete estimation.

Table 6. Results of hyperparameter tuning using meta-heuristic algorithms

Data	Algorithms	The number of Layer	Neurons per Layer	Activation Function	R^2	MAE
The high strength concrete	WOA	3	[9,8,8]	['purelin', 'tansig', 'tansig']	0.95	0.47
	ALO	3	[9,3,7]	['logsig', 'purelin', 'logsig']	0.94	0.46
	PSO	3	[4,7,10]	['logsig', 'logsig', 'tansig']	0.98	0.36
The normal strength concrete	WOA	3	[1,8,5]	['tansig', 'logsig', 'purelin']	0.75	1.59
	ALO	3	[1,8,5]	['tansig', 'logsig', 'purelin']	0.75	1.59
	PSO	3	[1,8,5]	['tansig', 'logsig', 'purelin']	0.75	1.59

4 Conclusion

In this study, the modeling of the concrete elasticity modulus has been carried out using the ANN. ANN is a method that has been used for many years and has achieved successful results for many engineering problems. However, when using ANN or other modeling methods, the biggest challenge is to determine the optimum parameters of the modeling methods. Because although the method is suitable for the problem, if the optimum parameters are not set, unfortunately, good results cannot be obtained. Different methods have different numbers of parameters that need to be set. The parameters of the ANN to be adjusted are the number of hidden layers, the number of neurons in the hidden layers and the activation functions of the hidden layers. Although ANN has 3 different parameters, it is difficult and time consuming to manually adjust these parameters because the range of each parameter is wide. Sometimes the most suitable parameters cannot be found. In this case, the hyperparameter optimization process, which is among the popular topics of recent years, is used. Parameters that need to be adjusted with hyperparameter optimization are determined with the help of a meta-heuristic optimization algorithm. This saves time and takes into account all possible search space. In this study, Whale Optimization Algorithm, Ant Lion Optimizer and Particle Swarm Optimization algorithms were used to determine the hyperparameters of the ANN. These algorithms have been selected among the current and successful algorithms in the literature. As a result of the analysis made, the most successful results were obtained with the PSO. The pair of PSO and ANN has been the best choice for modeling the concrete elasticity modulus. The modeling of the concrete elasticity modulus was compared with previous studies. In 71% of the comparisons, the method in this study achieved the same or better results than previous studies. More successful results were obtained in 51% of these comparisons. The successful results obtained all indicate that a three-layered neural network should be used. Although the problem addressed in this study involves a small amount of data, it is evident from the results that it is a challenging modeling task. Especially in modeling normal strength concrete, all algorithms achieved the same result. However, the R^2 and MAE values did not reach the desired levels. Despite this, better results were obtained compared to previous studies in literature. Therefore, it can be concluded that the proposed model is successful. As a result, the method aimed with this study was more successful than previous studies. It has been shown that better results can be obtained with hyperparameter optimization. It can be a guide for the use of hyperparameter optimization in future studies.

5 Author contribution declaration

In the study, Author 1 carried out the formation of the idea, literature review, evaluation of the results obtained, interpretation of the results and all writing stages.

6 Ethics committee approval and conflict of interest statement

"There is no need to obtain permission from the ethics committee for the article prepared".

"There is no conflict of interest with any person/institution in the article prepared".

7 References

[1] F. Demir, "Prediction of elastic modulus of normal and high strength concrete by artificial neural networks," *Constr. Build. Mater.*, vol. 22, pp. 1428–1435, 2008, doi: 10.1016/J.CONBUILDMAT.2007.04.004.

[2] F. Farooq *et al.*, "A Comparative Study of Random Forest and Genetic Engineering Programming for the Prediction of Compressive Strength of High Strength Concrete (HSC)," *Appl. Sci.*, vol. 10, p. 7330, 2020, doi: 10.3390/APP10207330.

[3] J. A. Garcia, J. F. Gomez, N. T. Castellanos, J. I. Abell An Garc Ia, and J. Fernandez G Omez C, "Properties prediction of environmentally friendly ultra-high-performance concrete using artificial neural networks," *Constr. Build. Mater.*, vol. 26, pp. 2319–2343, 2020, doi: 10.1080/19648189.2020.1762749.

[4] N. A. Abdulla, "Using the Artificial Neural Network to Predict the Axial Strength and Strain of Concrete-Filled Plastic Tube," *J. Soft Comput. Civ. Eng.*, vol. 4, pp. 63–84, 2020, doi: 10.22115/SCCE.2020.225161.1198.

[5] S. N. A. Akpinar and P. A. Al-Gburi, "Machine learning in concrete's strength prediction," *Comput. Concr.*, vol. 29, pp. 433–444, 2022, doi: 10.12989/CAC.2022.29.6.433.

[6] Z. Zeng *et al.*, "Accurate prediction of concrete compressive strength based on explainable features using deep learning," *Constr. Build. Mater.*, vol. 329, p. 127082, 2022, doi: 10.1016/J.CONBUILDMAT.2022.127082.

[7] B. K. A. Mohamad Ali Ridho, C. Ngamkhanong, Y. Wu, and S. Kaewunruen, "Recycled Aggregates Concrete Compressive Strength Prediction Using Artificial Neural Networks (ANNs)," *Infrastructures*, vol. 6, p. 17, 2021, doi: 10.3390/INFRASTRUCTURES6020017.

[8] B. K. Khotimah *et al.*, "Random Search Hyperparameter Optimization for BPNN to Forecasting Cattle Population," *E3s Web Conf.*, vol. 499, p. 01017, 2024, doi: 10.1051/e3sconf/202449901017.

[9] J. Du, H. Ma, D. Sun, and P. Pan, "Data driven strength and strain enhancement model for FRP confined concrete using Bayesian optimization," *Structures*, vol. 41, pp. 1345–1358, 2022, doi: 10.1016/J.ISTRUC.2022.05.093.

[10] J. Han, C. Gondro, K. Reid, and J. P. Steibel, "Heuristic hyperparameter optimization of deep learning models for genomic prediction," *G3 Genes Genomes Genet.*, vol. 11, 2021, doi: 10.1093/G3JOURNAL/JKAB032.

[11] Z. Liu, X. Gu, H. Yang, L. Wang, Y. Chen, and D. Wang, "Novel YOLOv3 Model With Structure and Hyperparameter Optimization for Detection of Pavement Concealed Cracks in GPR Images," *IEEE Trans. Intell. Transp. Syst.*, 2022, doi: 10.1109/TITS.2022.3174626.

[12] M. W. Newcomer and R. J. Hunt, "NWTOPT – A hyperparameter optimization approach for selection of environmental model solver settings," *Environ. Model. Softw.*, vol. 147, p. 105250, 2022, doi: 10.1016/J.ENVSOFT.2021.105250.

[13] Y. Peng, D. Gong, C. Deng, H. Li, H. Cai, and H. Zhang, "An automatic hyperparameter optimization DNN model for precipitation prediction," *Appl. Intell.*, vol. 52, pp. 2703–2719, 2022, doi: 10.1007/S10489-021-02507-Y/TABLES/11.

[14] Z. Qu, J. Xu, Z. Wang, R. Chi, and H. Liu, "Prediction of electricity generation from a combined cycle power plant based on a stacking ensemble and its hyperparameter optimization with a grid-search method," *Energy*, vol. 227, p. 120309, 2021, doi: 10.1016/J.ENERGY.2021.120309.

[15] B. Senel and F. A. Senel, "Novel neural network optimization approach for modeling scattering and noise parameters of microwave transistor," *Int. J. Numer. Model. Electron. Netw. Devices Fields*, vol. 35, p. e2930, 2022, doi: 10.1002/JNM.2930.

[16] C. W. Tsai and Z. Y. Fang, "An Effective Hyperparameter Optimization Algorithm for DNN to Predict Passengers at a Metro Station," *ACM Trans. Internet Technol. TOIT*, vol. 21, 2021, doi: 10.1145/3410156.

[17] Y. Wei, Z. Chen, C. Zhao, Y. Tu, X. Chen, and R. Yang, "A BiLSTM hybrid model for ship roll multi-step forecasting based on decomposition and hyperparameter optimization," *Ocean Eng.*, vol. 242, p. 110138, 2021, doi: 10.1016/J.OCEANENG.2021.110138.

[18] Z. E. Aydin, B. I. Erdem, and Z. I. E. Cicek, "Prediction bike-sharing demand with gradient boosting methods," *PAMUKKALE Univ. J. Eng. Sci.-PAMUKKALE Univ. MUHENDISLIK Bilim. Derg.*, vol. 29, no. 8, pp. 824–832, 2023, doi: 10.5505/pajes.2023.39959.

[19] B. Haseli, G. Nouri, M. Mardi, E. Adili, and M. Bahari, "Prediction of Strength Parameters of Concrete Containing Different Additives Using Optimized Neural Network Algorithm.," vol. 7, no. 4, pp. 12–23, 2023, doi: 10.61186/nmce.2022.410.

[20] G. Franchini, "GreenNAS: A Green Approach to the Hyperparameters Tuning in Deep Learning," *Mathematics*, vol. 12, no. 6, p. 850, 2024, doi: 10.3390/math12060850.

[21] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, 2016.

[22] S. Mirjalili, "The ant lion optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, 2015.

[23] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1995, pp. 1942–1948. doi: 10.1109/ICNN.1995.488968.

[24] M. Gesoglu, E. Guneyisi, and T. Ozturan, "Effects of end conditions on compressive strength and static elastic modulus of very high strength concrete," *Cem. Concr. Res.*, vol. 32, pp. 1545–1550, 2002, doi: 10.1016/S0008-8846(02)00826-8.

[25] T. Ozturan, "An investigation of concrete abrasion as two phase material," Master's Thesis, Faculty of Civil Engineering, Istanbul Technical University, 1984.

[26] M. Turan and M. Iren, "Strain stress relationship of concrete," *J. Eng. Archit.*, vol. 12, pp. 76–81, 1997.

[27] T. H. Wee, M. S. Chin, and M. A. Mansur, "Stress-Strain Relationship of High-Strength Concrete in Compression," *J. Mater. Civ. Eng.*, vol. 8, pp. 70–76, 1996, doi: 10.1061/(ASCE)0899-1561(1996)8:2(70).

[28] F. A. Senel, F. Gokce, A. S. Yuksel, and T. Yigit, "A novel hybrid PSO–GWO algorithm for optimization problems," *Eng. Comput.*, vol. 35, pp. 1359–1373, 2019.