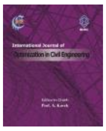
INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2018; 8(4):547-563



DEVELOPMENT OF ANFIS-PSO, SVR-PSO, AND ANN-PSO HYBRID INTELLIGENT MODELS FOR PREDICTING THE COMPRESSIVE STRENGTH OF CONCRETE

M. Torkan and M. Naderi Dehkordi[*, †]
Faculty of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran

ABSTRACT

Concrete is the second most consumed material after water and the most widely used construction material in the world. The compressive strength of concrete is one of its most important mechanical properties, which highly depends on its mix design. The present study uses the intelligent methods with instance-based learning ability to predict the compressive strength of concrete. To achieve this objective, first, a set of data pertaining to concrete mix designs containing fly ash was collected. Then, mix design parameters were used as the inputs of the artificial neural network (ANN), support vector machine (SVM), and adaptive neuro-fuzzy inference system (ANFIS) developed for predicting the compressive strength. In all these models, prediction accuracy largely depends on the parameters of the learning model. Hence, the particle swarm optimization (PSO) algorithm, as a powerful population-based algorithm for solving continuous and discrete optimization problems, was used to determine the optimal values of algorithm parameters. The hybrid models were trained and tested with 426 experimental data and their results were compared by statistical criteria. Comparing the results of the developed models with the real values showed that the ANFIS-PSO hybrid model has the best performance and accuracy among the assessed methods.

Keywords: concrete; compressive strength; artificial neural networks (ANN); support vector machine (SVM); adaptive neural-fuzzy inference system (ANFIS).

Received: 20 October 2017; Accepted: 15 January 2018

1. INTRODUCTION

Concrete plays a primary role in construction materials. Compressive strength is a property

[*] Corresponding author: Faculty of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran
Naderi@iaun.ac.ir (M. Naderi Dehkordi)

of concrete that nearly always is of concern [1]. The properties such as compressive strength of concrete depend on multiple parameters including water-cement ratio, the quantity of fine and coarse aggregates, additives, etc. The high number and nature of these parameters make concrete strength very difficult to predict. In this regard, a general rule to describe this complex system has not been provided yet and the currently available relationships are mostly empirical. Traditional methods, which are based on a generalization of past experiences, are not accurate enough to give satisfactory relationships for this purpose [2]. Therefore, fast and reasonably accurate prediction of concrete strength can benefit both design and quality control procedures [3]. Prediction of concrete strength before actual construction enables the engineers to improve upon the existing planning and quality control efforts. Also, this can lead to significant time and cost savings in the construction of large concrete structures. One method to predict the compressive strength of concrete is the use of intelligent methods with instance-based learning ability [2]. The main purpose of such modeling systems is to use a large set of experimental data for different concrete mixtures to reflect the nature of certain physical properties of concrete such as its compressive strength [4]. This prediction allows the designer to modify the mix design so as to achieve outcomes such as improved quality, rapid construction, or lower costs [3].

Data mining is the process of analyzing data from different perspectives and summarizing them into useful information (Fig. 1). Technically, data mining is the process of finding relationships or patterns between dozens of data items in a large database [5]. The process of knowledge discovery in databases consists of three steps: pre-processing, data mining, and post-processing. In the data mining process, an algorithm is used to detect patterns in the data. Generally, data mining techniques operate with a set of training examples and a set of test cases. The purpose of training examples is to train the algorithm for its target task and the purpose of test cases is to evaluate its performance. The purpose of the training process is to readjust the algorithm parameters so as to optimize the results. Eventually, validity and accuracy of the algorithm can be assessed with various error-based criteria. After validation, the algorithm can be used as a model to predict output variables [2].

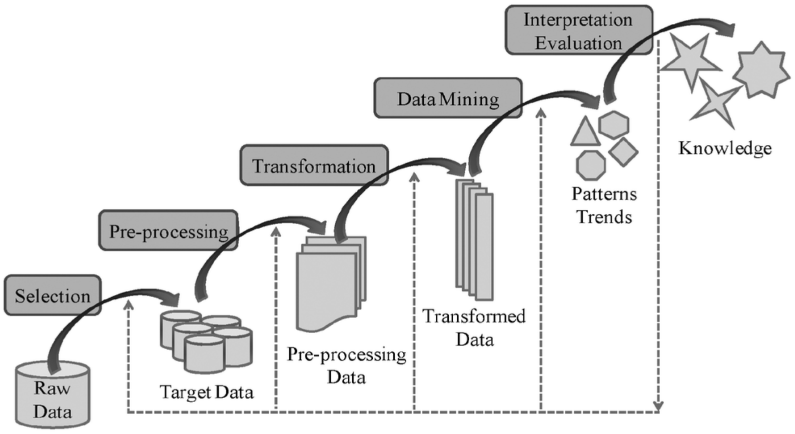


Figure 1. Data mining as a step in the process of knowledge discovery [6]

Review of literature shows that researchers who sought to predict the 28-day compressive strength of concrete with different mix designs have used different data mining techniques such as artificial neural networks (ANNs), support vector machines (SVMs), and decision trees for this purpose [7-9]. Zain and Abd [10] used multiple nonlinear regression to predict the compressive strength of high-performance concrete. They presented a multivariable regression equation for forecasting the strength of high-performance concrete and stated that the proposed model can predict the compressive strength of concrete at different ages with a high correlation coefficient. Saridemir [11] used the results of different tests reported in the literature to develop an ANN for predicting the compressive strength of mortars containing metakaolin with different mix designs and curing time. Atici [12] used multiple regression and ANN techniques to estimate the compressive strength of concrete containing different quantities of blast furnace slag and fly ash. Chou et al. [4] proposed a hybrid method composed of hierarchical classification and regression (HCR) techniques for estimating the compressive strength of high-performance concrete (HPC). Madandoust et al. [13] used the adaptive neuro-fuzzy inference system (ANFIS) and GMDH ANNs to predict the compressive strength of concrete based on cylindrical specimens. They reported that both ANFIS and GMDH neural network have a relatively good potential for modeling and prediction based on experimental data, but overall ANFIS can provide better predictions. Chou et al. [14] used basic learning methods including multilayer perceptron (MLP), SVM, regression and classification trees, and linear regression as the components of hybrid models, and showed that the combined use of learning techniques is better than using them alone. Nikoo et al. [15] used self-organizing (SOFM) networks to predict the compressive strength of concrete based on 173 different specimens. Yuan et al. [16] reported that conventional regression models that predict the concrete strength do not provide accurate results, and then proposed two genetic-based algorithms and ANFIS models to deal with this situation. Their results showed that ANFIS model, which is a combination of ANN and fuzzy logic, has a greater ability to predict the 28-day compressive strength of concrete.

In this paper, three intelligent methods including ANN, SVM, and ANFIS were optimized using particle swarm optimization (PSO) algorithm. The resulting hybrid models were used to predict the compressive strength of concrete based on experimental data.

2 MATERIALS AND METHODS

In this section, methods of modeling including ANN, SVR, and ANFIS are briefly discussed. Moreover, the methodology of the optimization (PSO) is presented. In addition, this section is devoted to explaining the required data for training and testing of proposed models.

2.1 Artificial neural networks (ANN)

ANN is an information processing system inspired by functional characteristics of biological neural networks. ANNs can be counted as generalized mathematical models of the human brain or neural biology. The ANN does not really solve the problem in a strictly mathematical sense, but it demonstrates information processing characteristics that gives an approximate solution to a given problem [17]. These networks are constructed based on the following assumptions:

1- Information processing is done in a number of basic elements called neurons. 2-

Signals between neurons are transmitted over communications links. 3- Each link has a weight, which (in a normal ANN) is the factor by which transmitted signal must be multiplied. 4- Each neuron applies a usually nonlinear activation function to input (the sum of weighted input signals) to determine the output signal. Neural networks are characterized by (1) the pattern of communication between neurons (called architecture), (2) how the weights of links are calculated (called training or algorithm), and (3) the activation function. ANN models are particularly useful for simulating the processes for which there is no exact knowledge or definition. Another characteristic of these models that distinguish them from their rivals is their lower sensitivity to the presence of an error in inputs, which can be attributed to the extensive processing of distributed information. In these systems, complex functions are carried out in a highly parallel structure that replaces the idea of putting a heavy computation load on a single fast computing unit with the idea of using a large number of simple computing units performing a common task. This division of labor has another positive outcome; since many neurons are involved simultaneously, the contribution of a single neuron is not particularly important, so an error or failure in one neuron has not much effect on other computing units or overall outcome [18, 19].

A typical ANN architecture consists of three or more layers: one input layer, one output layer, and one or several hidden layers, whose neurons are connected by weighted links. This architecture also contains a bias that is connected (through weighted links) to the output and hidden neurons. The number of neurons in each layer varies with application and design. Fig. 2 shows a schematic diagram of a neural network [4]. Over the years in all areas of civil engineering were undertaken different studies addressing various problems using ANNs with varying degrees of success [20, 21].

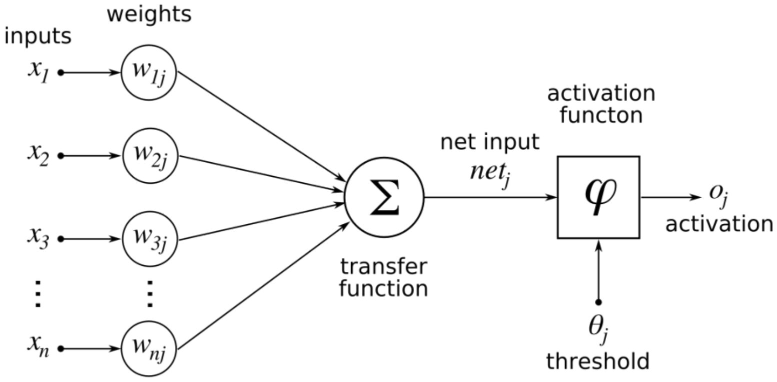


Figure 2. Schematic view of a neural network [1]

2.2 Support vector machine (SVM)

Introduced in 1995 by Vapnik, SVM is a supervised learning model in data analysis for classification (SVC) and regression problems (SVR) [22]. SVM is, in fact, a linear learning method for finding the optimal hyperplane separating two classes. As a supervised classification method, SVM seeks to maximize the distance to the nearest training point from any class (that distance is called margin) in order to optimize the generalization/classification performance on test data. In SVM, the solution is based only on those training data that are positioned on the margin's boundary. These points, which are called the support vectors, are shown in Fig. 3a. As presented in Fig. 3b, when the classes cannot be separated linearly, the space of input data must be transformed to a higher-dimension to allow the linear SVM formulation to be properly utilized. This transformation is often carried out using a kernel function (h). This function allows determining a nonlinear decision boundary, which is linear in higher-dimensional space, without the need to obtain the parameters of an optimal hyperplane in that higher-dimensional space. Thus, the solution can be written as the weighted sum of values of the kernel function employed in the support vectors [23].

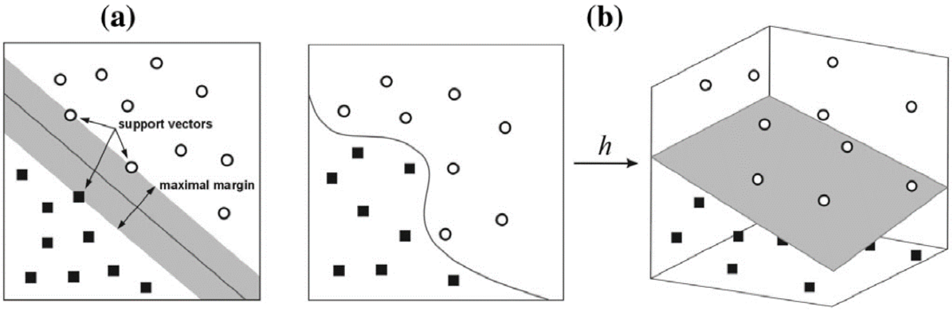


Figure 3. Geometric principle of the SVM algorithm. (a) Linear SVM in a separable classification problem. (b) Nonlinear SVM [23]

Let the training samples be denoted as XY = {(x, y)|(X_1 , Y_1),. . ., (X_{nd} , Y_{nd})}, where nd is the number of training samples. In linear SVR, the relation between input variable x_k and the predicted variable $\hat{y}_k$ can be described by the linear function f(x) taking the form of:

$$\hat{y}_k = f(x_k\ ) = \langle w\,, x_k\ \rangle + b \tag{1}$$

where $\langle .,. \rangle$denotes the dot product, w and b are weight vector and bias vector, respectively. The aim is to find a pair of unknown vectors of (w, b) which minimizes the prediction error for training samples and has at most ε deviation from actual target y_k. The latter implies that there would be no penalty during optimization for the pairs when $|y_k\ - f(x)| \leq \varepsilon$ and is defined by the ε-intensive loss function, $\boldsymbol{l_\varepsilon}$, which can be expressed as follows [24]:

$$l_\varepsilon = |y - f(x)|_\varepsilon = \max\{0, |y - f(x)| - \varepsilon\}. \tag{2}$$

One way to ensure that the minimal complexity risk would be obtained, in order to have optimal SRM, is to minimize the norm of w, $||w||^2 = \langle w, w \rangle$. Thus, in mathematical terms, the constrained regression problem can be written as a convex optimization problem as follows:

$$min_{w,b,\xi_k,\xi_k^*}\ \frac{1}{2}\ ||w||^2 + C\sum_{k=1}^{nd}(\xi_k + \xi_k^*), \tag{3}$$

$$subject\ to \begin{cases} y_k - \langle w\ ,x_k \rangle - b \ \le \ \varepsilon + \ \xi_k \\ \langle w\ ,x_k \rangle + b - \ y_k \le \ \ \varepsilon + \xi_k^* \\ \xi_k\ ,\xi_k^* \ge 0 \end{cases} \tag{4}$$

where $\xi_k and\ \xi_k^*$ are slack variables. In Eq. (3), the constant regularization parameter C≥ 0 determines the trade-off between the complexity of function and the deviation from the tolerable error ε chosen in prior. The problem, represented in Eq. (3) and (4), is a convex quadratic programming optimization which can be converted to a Lagrange function by introducing a dual set of positive Lagrange multiplier variables. This Lagrange function could be solved by maximizing its dual optimization problem. The final solution of the optimization problem is given by:

$$\mathrm{W} = \sum_{k=1}^{nd} (\alpha_k - \alpha_k^*) x_k \xrightarrow{yields} \hat{y}_{new} = f(x_{new}) = \sum_{k=1}^{nd} (\alpha_k - \alpha_k^*) \langle x_k\ ,x_{new} \rangle + b\ , \tag{5}$$

where $\alpha_k \ge 0\ \ and\ \alpha_k^*\ \ge$ are Lagrange multipliers. As seen in Eq. (5), w can be completely described as a linear combination of the training vectors and the Lagrange multipliers. The samples lie inside the ε -intensive tube make both Lagrange multipliers zero, and w actually is represented by only some of training vectors, called support vectors (SVs), which lie outside the ε-intensive tube. Thus, the complexity of the solution is not dependant on the dimensionality of the problem whereas SVs define the complexity of the function.

For enriching SVR algorithm to deal with the models with complex non-linear relation between input and output domains, one can implement some pre-processing procedures of training patterns. This can be done by mapping input vectors into a higher-dimensional feature space by the means of kernel functions, which yields the non-linear SVR for the kernel function of k⟨.,.⟩. Its solution is given by

$$\hat{y}_{new} = f(x_{new}) = \sum_{k=1}^{nd} (\alpha_k - \alpha_k^*) k \langle x_k\ ,x_{new} \rangle + b\ . \tag{6}$$

Kernel functions map the input space into the feature space by giving the weights of nearby data points in making an estimate. Therefore, they are important to control the complexity of the final solution. One may choose any arbitrary kernel functions, e.g., linear kernel function $k\langle x_t\ ,x_k \rangle = \langle x_t\ ,x_k \rangle$, polynomial kernel function $k\langle x_t\ ,x_k \rangle = (\langle x_t\ ,x_k \rangle + 1)^d, d > 0$, radial basis function (RBF) $k\langle x_t\ ,x_k \rangle =\ \exp(-y \| x_t\ ,x_k \|^2)\ ,y > 0$, etc. In highly non-linear spaces, RBF kernel usually yields more promising results in comparison with other mentioned kernels [25]. Consequently, we use only RBF kernel functions in this paper.

2.3 Adaptive network-based fuzzy inference system (ANFIS)

Introduced by Jang, ANFIS is a well-known and popular method for combined use of fuzzy inference system and ANN learning power for modeling of complex phenomena [26]. Fuzzy inference system is a system of structured knowledge in which each fuzzy rule describes the system's behavior in a locality. The inability of this system to adapt to changes in external environment has led to the incorporation of learning concept of neural networks into this system, which has resulted in the development of the ANFIS. Accordingly, ANFIS combines the learning capability of a neural network with inference capabilities provided by fuzzy logic [16]. The basic learning rule of ANFIS is the back-propagation gradient descent, which computes the signal error from the last layer (the output node) back toward the first layer (the input node). This learning rule is exactly like the back-propagation learning rule commonly used in feedforward neural networks [27]. Recent ANFIS designs have utilized a fast learning method called hybrid learning which is based on combined use of the gradient descent and the least squares method for finding a suitable set of antecedent and consequent parameters [9]. The architecture of ANFIS with two input variables is shown in Fig. 4. Also, the fuzzy-reasoning mechanism is illustrated as follows:

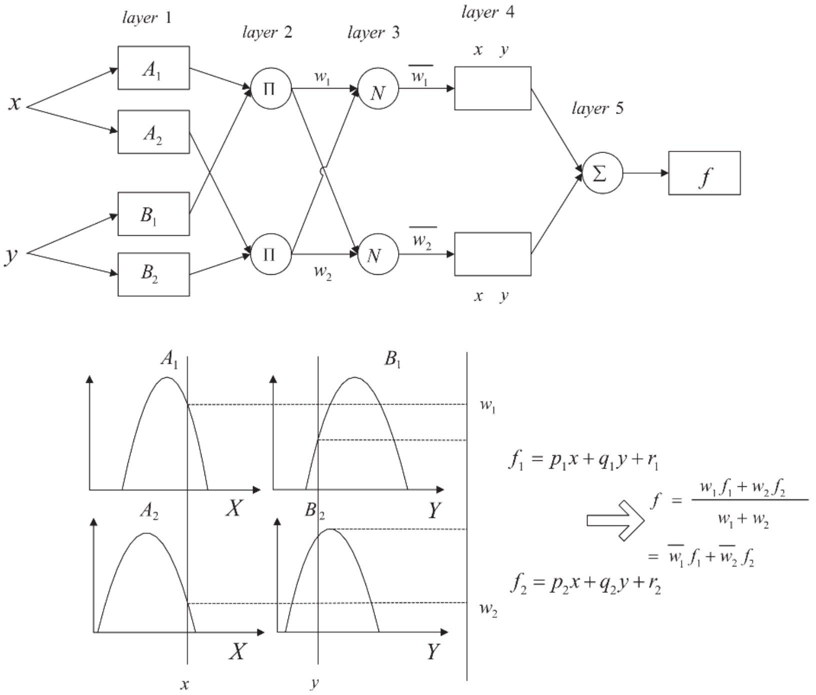

layer 1
layer 2
layer 3
layer 4
layer 5
x
y
A_1
A_2
B_1
B_2
Π
N
w_1
w_2
$\overline{w_1}$
$\overline{w_2}$
x y
Σ
f
X
Y
$f_1 = p_1x + q_1y + r_1$
$f = \frac{w_1f_1 + w_2f_2}{w_1 + w_2}$
$= \overline{w_1}f_1 + \overline{w_2}f_2$
$f_2 = p_2x + q_2y + r_2$

Figure 4. Architecture of ANFIS and Fuzzy-reasoning scheme of ANFIS [16]

$Rule1: IF\ x\ is\ A_1\ and\ y\ is\ B_1, THEN\ f_1 = p_1 + q_1 + r_1$
$Rule2: IF\ x\ is\ A_2\ and\ y\ is\ B_2, THEN\ f_2 = p_2 + q_2 + r_2$

The function of each layer is described subsequently:

Layer 1

The first layer of this architecture is the fuzzy layer. Each node of this layer makes the membership grad of a fuzzy set. Every node in this layer is an adaptive node with a node function.

$$O_i^1 = \mu_{A_i}(x) \tag{7}$$

where x is the input to node I, and A_i is the linguistic label associated with this node function. Premise parameters change the shape of the membership function.

Layer 2

Every node in this layer is a circle node labeled ∏, representing the firing strength of each rule, which multiplies the incoming signals and sends the product out. i.e. ∏-norm operation:

$$O_i^2 = \mu_{A_i}(x) \times \mu_{B_i}(y), \quad i = 1,2 \tag{8}$$

Layer 3

Every node in this layer is a circle node labeled N, representing the normalized firing strength of each rule. The ith node calculated the ratio of the ith rule's firing weight to the sum of all rule's firing weights:

$$O_i^3 = \overline{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1,2 \tag{9}$$

The outputs of this layer are called normalized firing strengths.

Layer 4

Every node in this layer is an adaptive node with a node function, indicating the contribution of the ith rule towards the overall output.

$$O_i^4 = \overline{w}_i f_i = \overline{w}_i(P_i x + q_i y + r_i), \quad i = 1,2 \tag{10}$$

Where $\overline{w}_i$ is the output of layer 3, and $\{P_i x + q_i y + r_i\}$ is the parameter set.

Layer 5

The signal node in this layer is a circle node labeled ∑, indicating the overall output as the summation of all incoming signals calculated, i.e.

$$O_i^5 = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{11}$$

There were five layers in this model, including input, input membership function, rule, output membership function and output.

2.4 Particle swarm optimization (PSO)

First introduced by Eberhart and Kennedy, PSO is an optimization tool inspired by the behavior of a swarm of birds and other social animals [28]. PSO is a swarm intelligence algorithm for solving numerical optimization problems, which has become very popular because of its effectiveness in science and engineering applications. Like genetic algorithm, PSO operation starts with a randomly initialized population, which is updated through a search procedure aimed at determining a better solution. In PSO, each bird (particle) moves at a certain pace based on a velocity vector. The velocity vector of each bird (particle) is governed and updated by two behavioral variables: memory (cognitive behavior) and current perception (social behavior). After sufficient time (iterations), birds (particles) can be expected to swarm toward the locations where their needs are met at an optimum level (optimum solutions).

The aforementioned behaviors, fundamentals of PSO, are formularized as follows:

$$v_i(j+1) = w(j)v_i(j) + \varphi_1(j)\big(pbest_i(j) - x_i(j)\big) + \varphi_2(j)\big(gbest_i(j) - x_i(j)\big) \tag{12}$$

$$\varphi_1(j) = C_1 r_1(j), \varphi_2(j) = C_2 r_2(j) \tag{13}$$

$$x_i(j+1) = x_i(j) + v_i(j+1) \tag{14}$$

where i denotes particle index; j represents iteration index; x_i indicates particle position. Corresponding particle velocity is represented by v_i; $pbest_i$ is its own previous best position; gbest is the previous best position of the entire swarm; and χ is a parameter called constriction coefficient that handles magnitude of the velocity. In Eq. (12), the second term on the right-hand side is a cognitive term and the third is a social term. C_1 and C_2 are, respectively, cognitive and social acceleration constants. r_1 and r_2 are two random variables ranged in [0, 1] uniformly. In some references, gbest has been defined individually for each particle with a defined neighborhood. Each particle is considered to have a neighborhood consisting of a number of other particles that impacts on the particle's movement. Neighborhoods can be defined in different ways. Different criteria lead to different topologies and directly act upon results [29, 30].

2.5 Data collection

Dataset was obtained from UC Irvine repository [31]. This dataset includes the data of 1030 specimens of normally cured ordinary Portland concrete with different additives, collected from laboratories of various universities. All compressive strength tests have been performed on 15 cm cylindrical samples in accordance with standards [4]. The present study used only the data pertaining to compressive strength at the age of 28 days. Overall, 428 data points (mix designs) of this dataset were employed. Of these 428 data points, 90% (383 data points) was used for training and the remaining 10% (43 data points) was used for testing the models. Table 1 shows the characteristics of the data used.

Table 1: Basic descriptive statistics for the original database

Parameters	**Unit**	**Min.**	**Max.**	**Avg.**
Cement	kg/m^3	102	540	265.16
Blast furnace slag	kg/m^3	0	359.4	86.08
Fly ash	kg/m^3	0	200.1	62.96
Water	kg/m^3	121.8	247	183.05
Superplasticizer	kg/m^3	0	32.2	7
Fine aggregates	kg/m^3	801	1145	956.11
Coarse aggregates	kg/m^3	594	922.6	764.49
28-day compressive strength	MPa	8.54	81.75	36.69

3 MODELS AND RESULTS

In this section, the procedures to develop intelligent models to predict the 28-day compressive strength using ANN-PSO, SVR-PSO, and ANFIS-PSO are presented. The results of these models with the actual data are also compared.

3.1 Modelling with ANN-PSO

As mentioned, this study was conducted with 428 mix designs, divided into two groups of training (383 mix designs) and test (43 mix designs). The first step in the development of an ANN for predicting the compressive strength is to normalize the input data using Eq. 15.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{15}$$

The ANN model developed in this study is a feed-forward network with the back-propagation algorithm. The back-propagation algorithm is the most successfully and widely used algorithm among artificial neural networks [32, 33]. Fig. 5 shows a schematic diagram of the developed network. This network consists of one hidden layer composed of sigmoid neurons and an output layer (purelin). The use of only one hidden layer reduces the complexity of the model. The network inputs include the quantities of cement, blast furnace slag, fly ash, water, superplasticizer, fine aggregates, and coarse aggregates, while its output is the 28-day compressive strength of concrete. Determining the number of hidden neurons is a critical decision concerning ANN architecture. In this connection, several equations have been proposed [34]. Assuming $2N_i + 1$ as the maximum number of neurons required in the hidden layer, a trial and error process was carried out and the number yielding the best network performance was selected. The final outcome of this selection process was the use of 15 neurons in the hidden layer. The ANN's ability to process information is closely related to its architecture and weights. In this study, to improve the ANN performance, its weights and biases were optimized using PSO. The primary parameters of PSO algorithm were the number of particles, the maximum number of iterations, inertia weight, and velocity coefficients, which are given in the table 2. The resulting model was developed with MATLAB (version 7.10.0.499(R2010a)) software. Figs. 6 and 7 compare the results of the ANN-PSO model with the actual values.

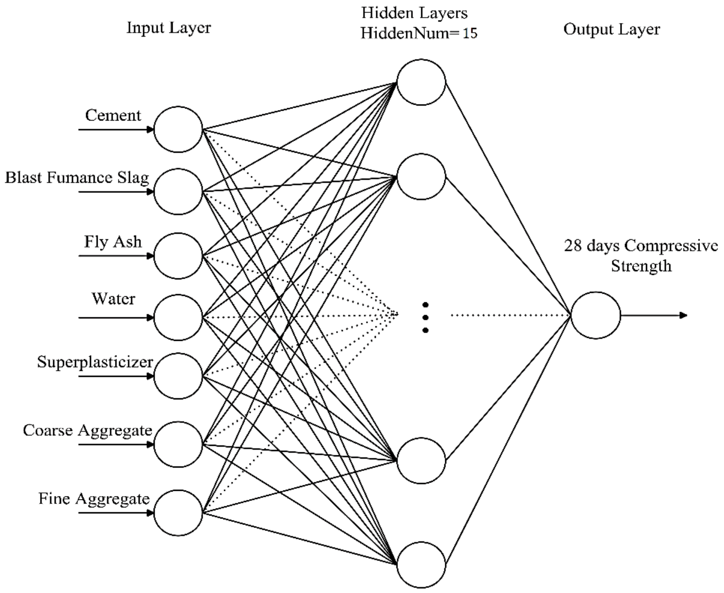


Figure 5. Architecture of the developed Neural network

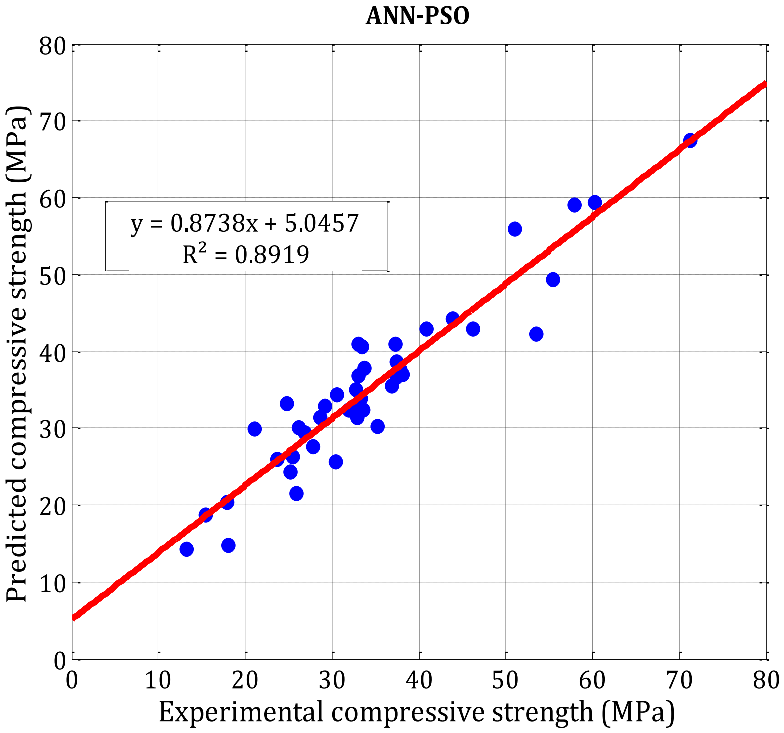


Figure 6. The relationship between actual and predicted values

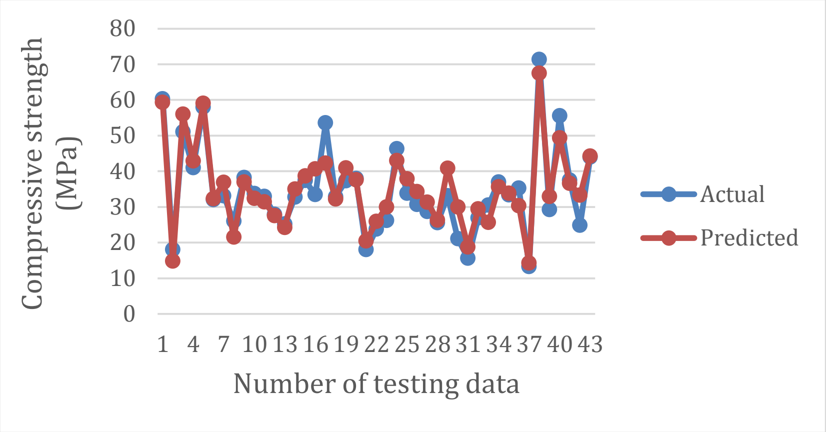


Figure 7. Comparison of measured and predicted values by ANN-PSO model

3.2 Modelling with SVR-PSO

The same 383 and 43 data points used for the ANN model were also used for training and test of the SVR model. Again, model inputs were the quantities of cement, blast furnace slag, fly ash, water, superplasticizer, fine aggregates, and coarse aggregates, and the output was the 28-day compressive strength of concrete. The developed SVR model is based on radial basis function (RBF) kernel. SVR’s generalization capability heavily depends on its learning parameters including penalty factor (C) and RBF kernel deviation (ɣ). The nonlinear behavior of SVR model with respect to these parameters makes finding their best combination more difficult. Thus, in this study, the PSO algorithm was used to determine the optimal value of these two parameters and thereby improve the accuracy of the SVR model. The primary parameters considered for the PSO algorithm are given in the following table. Like before, modeling was carried out using MATLAB (version 7.10.0.499(R2010a)) software. Fig. 8 shows the linear relationship between the actual value and the model estimates. Also, Fig. 9 compares the predicted values with measured ones.

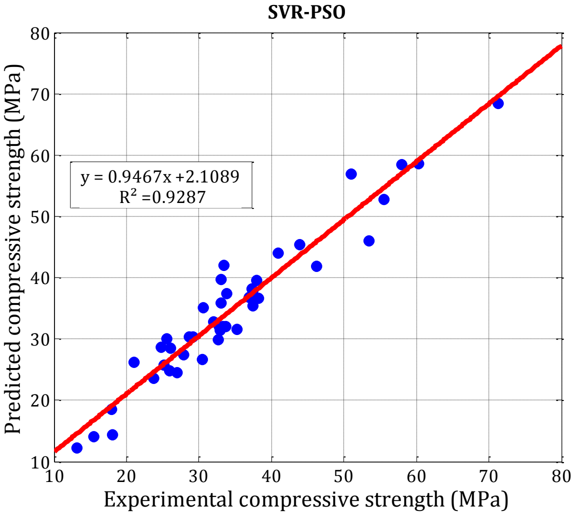


Figure 8. The relationship between actual and predicted values

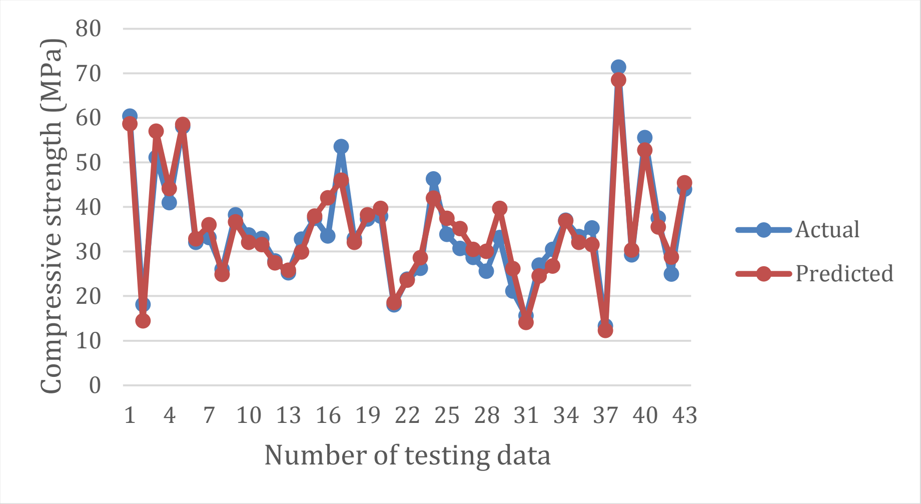


Figure 9. Comparison of measured and predicted values by SVR-PSO model

3.3 Modelling with ANFIS-PSO

Compressive strength was also predicted using an ANFIS model optimized with PSO algorithm. The ANFIS-PSO model was based on a Gaussian membership function and was coded using MATLAB (version 7.10.0.499 (R2010a)) software. The inputs of this model were again the quantities of cement, blast furnace slag, fly ash, water, superplasticizer, fine aggregates, and coarse aggregates. The primary parameters of the PSO algorithm, which were obtained through trial and error, are shown in Table 2. Figs. 10 and 11 compare the actual values with the estimations of the ANFIS-PSO model.

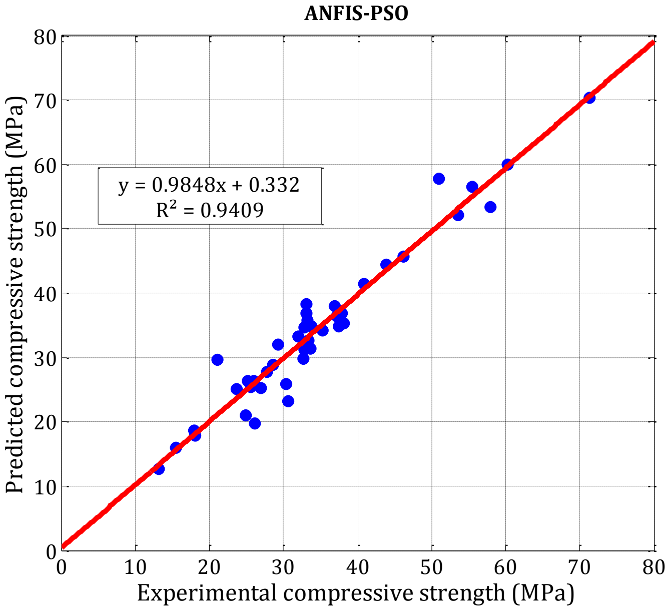


Figure 10. The relationship between actual and predicted values

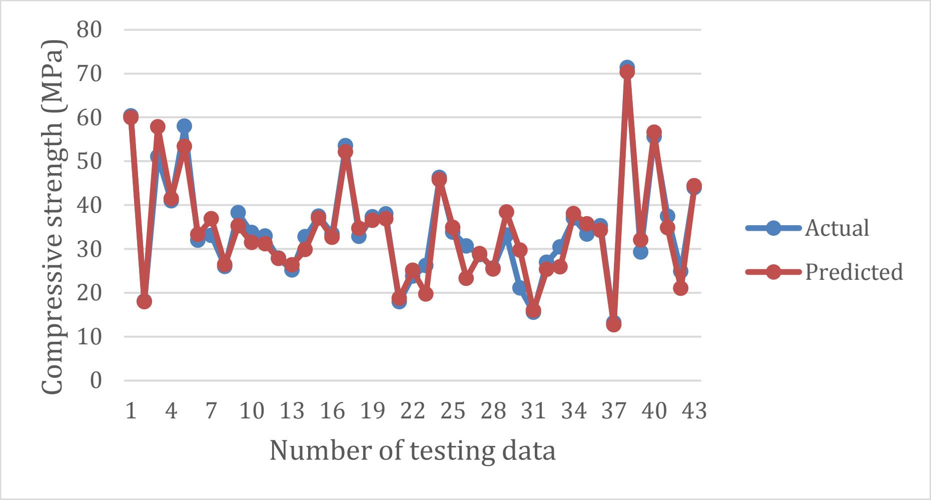


Figure 11. Comparison of measured and predicted values by ANFIS-PSO model

Table 2: Main parameters of the PSO

Parameters	Value		
Model	ANN	SVR	ANFIS
Number of particles	50	30	20
Number of iterations	300	150	250
Cognitive acceleration (C1)	2	2	2
Social acceleration (C2)	2	2	2
inertia weight	0.8	0.8	0.9

4. PERFORMANCE ASSESSMENT OF MODELS

In this section, performances of constructed models (ANN–PSO, SVR-PSO, and ANFIS-PSO) are evaluated using 43 testing datasets, which were not incorporated into the training models. To evaluate the accuracy of the mentioned models, three criteria are used: the coefficient of determination (R^2; Eq. 16), mean absolute percentage error (MAPE; Eq. 17), and root mean square error (RMSE; Eq. 18). A predictive model is accepted as excellent when R^2 is 1 and MAPE and RMSE are 0.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(T_i - P_i)^2}{\sum_{i=1}^{N}(T_i - \bar{T})^2} \tag{16}$$

$$MAPE(\%) = \frac{1}{N}\sum_{i=1}^{N}\frac{| T_i - P_i |}{T_i} \times 100 \tag{17}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(T_i - P_i)^2} \tag{18}$$

where T_i, P_i, $\bar{T}$ and N are target/measured values, predicted values, mean of target values, and total number of input data, respectively[35].

The values of performance indices (R^2, MAPE, and RMSE) for ANN–PSO, SVR-PSO, and ANFIS-PSO models are listed in Table 3.

Table 3: Performance indices for various models

	ANN+PSO	SVR+PSO	ANFIS-PSO
R^2	0.8919	0.9287	0.9409
MAPE (%)	10.0526	7.9363	6.7203
RMSE	4.0281	3.2362	2.9861

As presented in Table 3, the ANN model exhibits a poorer performance than the other two models. The SVR model shows a better performance than the ANN model in terms of not only accuracy, but also execution time and memory use. However, the ANFIS model shows the best performance among the assessed models. Note that errors of all three intelligent methods are acceptably low and they all can be counted as good alternatives to time-consuming and costly strength tests. Nevertheless, it should be reminded that, given the characteristics of these intelligent methods, a change in materials or conditions can have a negative effect on their accuracy. Also, using a more distributed dataset and a higher number of data items for training can make the results substantially closer to reality.

5 CONCLUSION

The present study was conducted to develop hybrid intelligent models for predicting the 28-day compressive strength of concrete. For this purpose, the PSO algorithm was used to improve the performance of ANN, SVR, and ANFIS models developed for prediction of the compressive strength of concrete. This improvement was carried out by optimizing the parameter setting of the mentioned models with the PSO algorithm. Comparing the results of models with the actual values showed the acceptable accuracy of these hybrid intelligent data mining techniques in the prediction of compressive strength. Comparing the results showed that the ANFIS-PSO model provides more reliable predictions than the other methods. The results also indicated the superiority of the SVR-PSO over the ANN-PSO in terms of accuracy, execution time, and memory use. The best R^2 achieved with the test dataset was 0.9409, which was obtained by ANFIS-PSO model. Moreover, the R^2 achieved by SVR-PSO and ANN-PSO models were respectively 0.9287 and 0.8919. Finally, it can be concluded that the use of data mining techniques for analyzing the available experimental data and creating a model to predict the compressive strength of concrete can be of great assistance for avoiding the repeat of experiments. In addition, optimizing the parameters of data mining models with PSO leads to certain improvement in their performance and reliability and generalizability of their results.

Acknowledgments: The authors would like to express their thanks to Mr. Hamid Kalhori for his kind help during the preparation of manuscript.

REFERENCES

1. Kaveh A, Khaleghi HA. Prediction of strength for concrete specimens using artificial neural network, *Asian J Civil Eng* 2000; **1**(2): 1-13.
2. Martins FF, Camões A. Prediction of compressive strength of concrete containing fly ash using data mining techniques, *Cement Wapno Beton* 2013; (1): 39-51.
3. Chou JS, Chiu CK, Farfoura M, Al-Taharwa I. Optimizing the prediction accuracy of concrete compressive strength based on a comparison of data-mining techniques, *J Comput Civil Eng* 2010; **25**(3): 242-53.
4. Chou JS, Tsai CF. Concrete compressive strength analysis using a combined classification and regression technique, *Automat Construct* 2012; **24**: 52-60.
5. Yücel KT, Özel C. Modeling of mechanical properties and bond relationship using data mining process, *Adv Eng Softw* 2012; **45**(1): 54-60.
6. Tinoco J, Correia AG, Cortez P. Application of data mining techniques in the estimation of the uniaxial compressive strength of jet grouting columns over time, *Construct Build Mater* 2011; **25**(3): 1257-62.
7. Safarzadegan Gilan S, Bahrami Jovein H, Ramezanianpour AA. Hybrid support vector regression - Particle swarm optimization for prediction of compressive strength and RCPT of concretes containing metakaolin, *Construct Build Mater* 2012; **34**: 321-9.
8. Khademi F, Akbari M, Jamal SM, Nikoo M. Multiple linear regression, artificial neural network, and fuzzy logic prediction of 28 days compressive strength of concrete, *Front Struct Civil Eng* 2017; **11**(1): 90-9.
9. Sadrmomtazi A, Sobhani J, Mirgozar M. Modeling compressive strength of EPS lightweight concrete using regression, neural network and ANFIS, *Construct Build Mater* 2013; **42**: 205-16.
10. Zain MFM, Abd SM. Multiple regression model for compressive strength prediction of high performance concrete, *J Appl Sci* 2009; **9**(1): 155-60.
11. Sarıdemir M. Predicting the compressive strength of mortars containing metakaolin by artificial neural networks and fuzzy logic, *Adv Eng Softw* 2009; **40**(9): 920-7.
12. Atici U. Prediction of the strength of mineral admixture concrete using multivariable regression analysis and an artificial neural network, *Expert Syst Applicat* 2011; **38**(8): 9609-18.
13. Madandoust R, Bungey JH, Ghavidel R. Prediction of the concrete compressive strength by means of core testing using GMDH-type neural network and ANFIS models, *Comput Mater Sci* 2012; 51(1): 261-72.
14. Chou JS, Tsai CF, Pham AD, Lu YH. Machine learning in concrete strength simulations: Multi-nation data analytics, *Construct Build Mater* 2014; **73**: 771-80.
15. Nikoo M, Zarfam P, Sayahpour H. Determination of compressive strength of concrete using Self Organization Feature Map (SOFM), *Eng Comput* 2015; **31**(1): 113-21.
16. Yuan Z, Wang LN, Ji X. Prediction of concrete compressive strength: Research on hybrid models genetic based algorithms and ANFIS, *Adv Eng Softw* 2014; **67**: 156-63.
17. Iranmanesh A, Kaveh A. Structural optimization by gradient base neural networks, *Int J Numer Meth Eng* 1999; **46**(2): 297-311.
18. Fausett L. *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*, Prentice-Hall, 1994.
19. Beale R, Jackson T. *Neural Computing-an Introduction*, CRC Press, 1990.
20. Rofooei R, Kaveh A, Farahani FM. Estimating the vulnerability of concrete moment resisting frame structures using artificial neural networks, *Int J Operat Res* 2011; **1**(3): 433-8.

21. Kaveh A, Iranmanesh A. Comparative study of backpropagation and improved counterpropagation neural nets in structural analysis and optimization, *Int J Space Struct* 1998; **13**(4): 177-85.
22. Cortes C, Vapnik V. Support-vector networks, *Mach Learn* 1995; **20**(3): 273-97.
23. Moraes R, Valiati JF, Neto WPG. Document-level sentiment classification: An empirical comparison between SVM and ANN, *Expert Syst Applicat* 2013; **40**(2): 621-33.
24. Vapnik VN, Vapnik V. *Statistical Learning Theory*, Wiley, New York, 1998.
25. Gunn SR. Support vector machines for classification and regression, ISIS technical report, 1998, 14.
26. Jang JS. ANFIS: adaptive-network-based fuzzy inference system, *IEEE Transact Syst, Man, Cybernet* 1993; **23**(3): 665-85.
27. Topçu İB, Sarıdemir M. Prediction of mechanical properties of recycled aggregate concretes containing silica fume using artificial neural networks and fuzzy logic, *Computat Mater Sci* 2008; **42**(1): 74-82.
28. Eberhart RC, Kennedy J. A new optimizer using particle swarm theory, *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, New York, USA, 1995, pp. 39-43.
29. Kennedy J. Bare bones particle swarms, *Proceedings of the Swarm Intelligence Symposium*, 2003.
30. Mendes R, Kennedy J, Neves J. The fully informed particle swarm: simpler, maybe better, *IEEE Transact Evolut Comput* 2004; **8**(3): 204-10.
31. Yeh IC. Modeling of strength of high-performance concrete using artificial neural networks, *Cement Concr Res* 1998; **28**(12): 1797-1808.
32. Kaveh A, Servati H. Design of double layer grids using backpropagation neural networks, *Comput Struct* 2001; **79**(17): 1561-8.
33. Kaveh A, Elmieh R, Servati H. Prediction of moment-rotation characteristic for semi-rigid connections using BP neural networks, *Asian J Civil Eng* 2001; **2**(2): 131-42.
34. Armaghani DJ, Raja RSNSB, Faizi K, Rashid ASA. Developing a hybrid PSO–ANN model for estimating the ultimate bearing capacity of rock-socketed piles, *Neural Comput Applicat* 2017; **28**(2): 391-405.
35. Chandwani V, Agrawal V, Nagar R. Modeling slump of ready mix concrete using genetic algorithms assisted training of artificial neural networks, *Expert Syst Applicat* 2015; **42**(2): 885-93.