Scientia Iranica A (2017) 24(2), 487–496

SCIENTIA
IRANICA

Sharif University of Technology
Scientia Iranica
Transactions A: Civil Engineering
www.scientiairanica.com

Comparison of artificial neural network and coupled simulated annealing based least square support vector regression models for prediction of compressive strength of high-performance concrete

M.A. Ayubi Rad[a,*] **and M.S. Ayubi Rad**[b]

a. *Advanced Control Systems Lab (ACSL), Control & Intelligent Processing Center of Excellence, University of Tehran, North Kargar Ave, Tehran, P.O. Box 14395/515, Iran.*
b. *Department of Civil and Environmental Engineering, Shiraz University, Molla Sadra St., Shiraz, Fars, P.O. Box 71348-51154, Iran.*

Received 19 April 2015; received in revised form 30 November 2015; accepted 27 February 2016

KEYWORDS
High performance concrete;
Compressive strength;
Coupled simulated annealing;
ANN;
LSSVR.

Abstract. High-Performance Concrete (HPC) is a complex composite material with highly nonlinear mechanical behavior. Concrete compressive strength, as one of the most essential qualities of concrete, is also a highly nonlinear function of ingredients. In this paper, Least Square Support Vector Regression (LSSVR) model based on Coupled Simulated Annealing (CSA) has been successfully used to find the nonlinear relationship between the concrete compressive strength and eight input factors (the cement, the blast furnace slags, the fly ashes, the water, the superplasticizer, the coarse aggregates, the fine aggregates, age of testing). To evaluate the performance of the CSA-LSSVR model, the results of the hybrid model were compared with those obtained by Artificial Neural Network (ANN) model. A comparison study is made using the coefficient of determination R^2 and Root Mean Squared Error (RMSE) as evaluation criteria. The accuracy, the computational time, the advantages and shortcomings of these modeling methods are also discussed. The training and testing results have shown that ANNs and CSA-LSSVR models have strong potential for predicting the compressive strength of HPC.

1. Introduction

High-performance concrete is a construction material characterized by high workability, high strength, and high durability. The American Concrete Institute (ACI) defines HPC as concrete meeting special combination of performance and uniformity requirements that cannot always be achieved routinely using conventional constituents and normal mixing, placing, and curing practices [1]. The conventional concrete is a mixture of water, Portland cement, fine aggregate, and coarse aggregate, while HPC employs fly ash, blast furnace slag, and silica fume as mineral admixtures, and superplasticizer as chemical admixture [2,3]. The use of mineral admixtures as partial cement replacement improves the properties of concrete by acting as fine fillers and pozzolanic materials [2]. On the other hand, the chemical admixture improves the compressive strength of HPC by reducing the water content and the level of porosity within the hydrated cement paste [3,4].

The compressive strength is a major mechanical property and probably the most essential quality of

*. *Corresponding author. Tel.: +98 711 7137315833; Fax: +98 21 61119742*
Email addresses: m.ayubirad@ut.ac.ir (M.A. Ayubi Rad); m.s.ayubirad@shirazu.ac.ir (M.S. Ayubi Rad)

concrete which is generally obtained by measuring the concrete specimen after a standard curing of 28 days [5], but waiting 28 days to get the 28-day compressing strength is time-consuming and uncommon. Therefore, designing a prediction model to obtain an early determination of compressive strength has gained a lot of attention.

For conventional concrete, the linear and nonlinear regression models can be used to predict the values of compressive strength, but for HPC as the number of input factor increases, the relationship between the input factors and the compressive strength becomes highly nonlinear and complex. Hence, the regression models are not suitable for predicting the values of compressive strength of HPC [6]. Therefore, more attentions have been paid to models based on artificial intelligence. Yeh [7] proposed a novel neural network architecture and examined its efficiency and accuracy in modeling the compressive strength of concrete. Prasad et al. [8] used an ANN to predict the 28-day compressive strength of a normal and high strength Self-Compacting Concrete (SCC) and HPC with high volume fly ash. Sobhani et al. [9] proposed several regression models, Adaptive Neuro Fuzzy Inference System (ANFIS), and ANN model to predict the 28-day compressive strength of no-slump concrete. They showed that the neural network and ANFIS models can predict the compressive strength with satisfactory performance, but the regression models fail to be reliable. Alshihri et al.[10] investigated the use of Back-Propagation (BP) and Cascade Correlation (CC) neural networks for predicting the compressive strength of Light Weight Concrete (LWC). The findings of their study indicated that the neural network models are sufficient tools for predicting the compressive strength of LWC. Topcu and Sarıdemir [11] developed ANN and fuzzy logic models for predicting 7, 28, and 90 days compressive strength of concretes containing high-lime and low-lime fly ashes. Their conclusions have shown that ANN and fuzzy logic models are practical models for predicting the compressive strength of concrete. Cheng et al. [6] proposed an artificial intelligence hybrid system to predict the HPC compressive strength by fusing the fuzzy logic, weighted Support Vector Machines (SVMs), and fast messy genetic algorithm into Evolutionary Fuzzy Support Vector Machine Inference Model for Time Series Data (EFSIMT). Their validation results indicated that EFSIMT method achieves higher performance goal in comparison with SVMs. They also concluded that in comparison with ANN and EFSIMT, the SVMs have the least satisfactory result.

The aim of this study is to improve the accuracy of SVM model in predicting the compressive strength of HPC. In spite of SVM's many advantages, the theory of SVM only covers the determination of the parameters for a given value of the regularization and kernel parameters and choice of kernel. The existing SVM methods also have high algorithmic complexity and extensive memory requirements. In this paper, the LSSVR model, a variation of Support Vector Regression (SVR) model with lower computational cost, has been proposed. However, similar to SVR, the effectiveness of the LSSVR model depends on the appropriate regularization and kernel parameter settings which can be identified as an optimization problem. In this work, the CSA [12], as a global optimization method, has been used for determining the tuning parameters of LSSVR model. The proposed model is constructed, trained, tested, and validated by applying the HPC experimental data originally generated by Yeh [13]. It is shown that in comparison with ANN, in terms of accuracy, the proposed CSA-LSSVR method achieves comparable results.

2. Data collection

The experimental datasets were obtained from University of California, Irvine (UCI) database, provided by Professor Yeh [13]. The dataset is consisted of 1030 samples, each containing 8 components for input vector and one output value (compressive strength). To build and evaluate the model, we used random sampling and generated 4 datasets and named them Experiment 1, Experiment 2, Experiment 3, and Experiment 4. Each generated dataset was divided into two categories. 800 samples were used for training and 230 samples were used to evaluate the ANN and CSA-LSSVR models. Table 1 shows the experimental data used in this study. Table 2 presents the range of inputs and outputs.

3. Artificial neural network model

ANNs are computational models which were developed to mimic the biological neural networks. ANNs have been used by many researchers for a variety of different applications [14,15]. In civil engineering, the ANNs have been applied to damage detection of bridge structures [16], modeling the mechanical behavior of materials [17], active control of structures [18], optimal monitoring network of ground water [19], and concrete mix proportion design [20]. An artificial neuron is consisted of 5 major units: inputs, weights, sum function, activation function, and outputs. The inputs to the network are represented by $\mathbf{x} = (x_0, x_1, ..., x_n)$ where x_0 is a constant. Each input is multiplied by a connection weight. The weights are represented by $\mathbf{w} = (w_0, w_1, ..., w_n)$. The weighted sum of inputs is calculated as follows:

$$(\text{net})_j = \sum_{i=0}^{n} w_{ij} x_i, \tag{1}$$

Table 1. Experimental dataset.

Number	Input								Output
	Cement (kg/m^3)	BFS (kg/m^3)	Fly ash (kg/m^3)	Water (kg/m^3)	SP (kg/m^3)	CA (kg/m^3)	FA (kg/m^3)	Age (day)	CSC (MPa)
1	540	0	0	162	2.5	1040	676	28	79.99
2	540	0	0	162	2.5	1055	676	28	61.89
3	332.5	142.5	0	228	0	932	594	270	40.27
4	332.5	142.5	0	228	0	932	594	365	41.05
5	198.6	132.4	0	192	0	978.4	825.5	360	44.30
:	:	:	:	:	:	:	:	:	:
1030	260.9	100.5	78.3	200.6	8.6	864.5	761.5	28	32.4

Note: BFS: Blast Furnace Slag; CA: Coarse Aggregate; SP: superplasticizer; FA: Fine Aggregate;
CSC: Compressive Strength of concrete.

Table 2. Ranges of experimental data.

Range	Input								Output
	Cement (kg/m^3)	BFS (kg/m^3)	Fly ash (kg/m^3)	Water (kg/m^3)	SP (kg/m^3)	CA (kg/m^3)	FA (kg/m^3)	Age (day)	CSC (MPa)
Min	102	0	0	121.8	0	801	594	1	2.33
Max	540	359.4	200.1	247	32.2	1145	992.6	365	82.6

where $(\text{net})_j$ is the weighted sum of jth node, n is the number of neurons in the preceding layer, w_{ij} is the weight between this neuron and the ith neuron in the preceding layer, and x_i is the output of ith neuron in the preceding layer. Afterward, the weighted summation is fed through an activation function to generate the output. The most common activation functions are hyperbolic tangent sigmoid, sigmoid and linear functions. When the activation function of hidden layer neurons is nonlinear, then the artificial neural network can be proven to be a universal function approximator [21]. By selecting the hyperbolic tangent sigmoid as the activation function of hidden layer neurons, the output of jth neuron can be written as:

$$(\text{out})_j = \frac{2}{1+e^{-(\text{net}_j)}} - 1. \qquad (2)$$

In this work, ANN is used for non-parametric nonlinear regression. Therefore, the linear function is selected as the activation function of the output layer neurons.

3.1. Training of artificial neural networks

An ANN is composed of many artificial neurons which are linked together via network of weights and biases $(\mathbf{w}, b)$, carrying the output of one neuron as input to another neuron. The training procedure of ANN is consisted of finding the optimum values of these weights and biases. One of the most successful algorithms used for training a multilayer perceptron (MLP) neural network is back-propagation algorithm. This method calculates the error between the network outputs and desired targets and propagates back to the network through a learning mechanism. As a result, the weights and biases (thresholds) are updated until the network reaches a predefined performance goal. In this work, Levenberg-Marquardt Back-Propagation (LMBP) algorithm is selected as the training function instead of commonly used standard BP methods for its robustness in computing process to update weights and biases [22]. LMBP is often the fastest available BP algorithm, and is highly recommended as the first-choice supervised algorithm, although it requires more memory than other algorithms. The main problem of neural network in training stage is the existence of many local minimums. Back-propagation based methods frequently find suboptimal solutions being trapped in local minimums. On the other hand, the back-propagation algorithm can lead to a model which overfits the training data. To overcome these problems, the training data are divided into training and validation subsets. The training subset is used to train the neural network with different initial weights, while validation subset is used to stop the network from getting overtrained. To stop the ANN from getting overtrained, the model with the lowest RMSE on validation dataset is selected as the appropriate one. The number of hidden layer neurons is estimated by the trial and error procedure, and number of neurons in input layer is equal to the number of input variables.

In this work, to model the concrete compressive strength, an MLP neural network based on back-propagation algorithm is used (Figure 1). The MLP

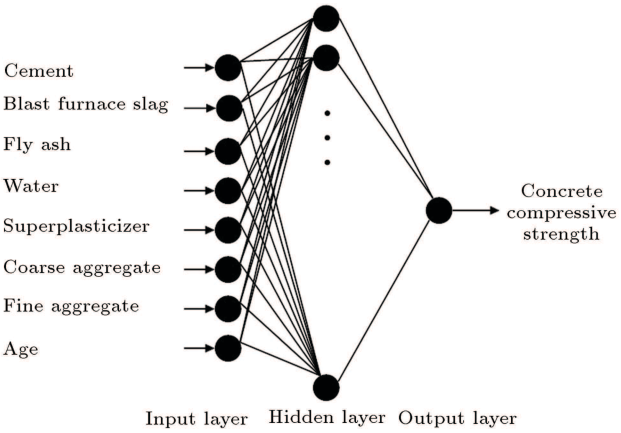

Figure 1. A three-layer ANN schematic.

neural network was created, trained, and implemented using Matlab neural network toolbox. The hyperbolic tangent sigmoid and linear function were selected as the activation functions of hidden layer and output layer, respectively. The structure of the network can be summarized as follows:

- Number of input layer neurons = 8;
- Number of hidden layer = 1;
- Number of hidden layer neurons = 20;
- Number of output neurons = 1;
- Performance goal = 0;
- Minimum gradient = 10^{-6};
- Maximum iterations = 10000.

By increasing the number of weights and biases of ANN, the overfitting occurrence is very much probable, and the training time would increase, while R^2 changes would be negligible.

4. Coupled simulated annealing based least square support vector regression model

Support vector machine proposed by Vapnik [23] is a powerful supervised learning method used for nonlinear classification, function approximation, and density estimation. SVM for classification (SVC) is a linear classifier which separates the classes by using hyperplanes. There are many hyperplanes which can separate the data; however, to have high accuracy and generalization on test data, SVC learns the optimal one with the maximized margin and minimized squared error. To separate the non-linearly separable data, SVC projects the data to a higher dimensional space and makes them linearly separable in the new representation. In other words, SVC learns the optimal hyperplane in the new feature space. However, mapping the data usually increases the number of features and computational cost. To overcome this problem, kernel tricks have been suggested. Using the kernel tricks, the mapping does not have to be explicitly computed and the computation with the mapped features remains efficient [24]. Linear kernel, quadratic kernel, polynomial kernel, and Radial Basic Function (RBF) are the most popular kernels for real valued vector inputs.

The goal of SVM for regression (SVR) is to find the linear regression function in higher dimensional space in a way that the estimated function deviates the least from training data, and it is as flat as possible at the same time [25]. Assume that $\{x_i, y_i\}_{i=1,2,\ldots,l}$ are given training data and they are in a linear relation, then the regression function can be defined as:

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b \quad \text{with} \quad \mathbf{w}, \mathbf{x} \in R^n \ \& \ b \in R. \tag{3}$$

In most cases, the relation between the input and output variables is nonlinear. Hence, the SVR algorithm maps the input data to a higher dimensional space in order to conduct the linear regression. Assume that the mapping ϕ takes $\mathbf{x}$ of input space and maps it to feature space. Now, the linear regression function in feature space can be written as:

$$f(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) + b. \tag{4}$$

In epsilon Support Vector Regression (ε-SVR), the goal is to find the function $f(\mathbf{x})$ in a way that the difference between the function value and the actual target, y_i, becomes less than ε for all training data. The problem of finding the regression function can be written as a convex optimization problem with the following constrains:

$$\min \frac{1}{2}||\mathbf{w}||^2 \begin{cases} y_i - \mathbf{w}^T\phi(\mathbf{x}_i) - b \le \varepsilon \\ \mathbf{w}^T\phi(\mathbf{x}_i) + b - y_i \le \varepsilon \end{cases} \tag{5}$$

However, the convex optimization may not always be feasible, and f function with ε precision may not always exist. To solve this problem, we can introduce two slack variables ξ and ξ_i* similar to soft margin function. Figure 2 shows the situation graphically. Hence, the

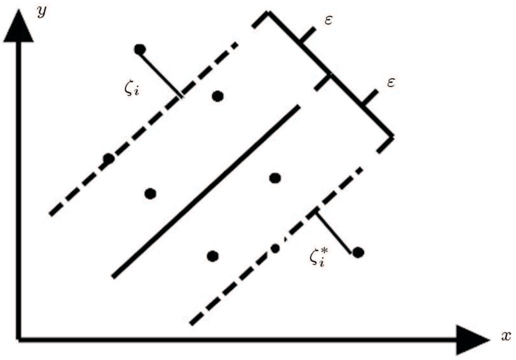

Figure 2. Epsilon-intensive SVR with slack variables.

convex optimization problem is reformulated as follows:

$$\min \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*)$$

$$\begin{cases} y_i - \mathbf{w}^T\phi(\mathbf{x}_i) - b \le \varepsilon + \xi_i \\ \mathbf{w}^T\phi(\mathbf{x}_i) + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases} \tag{6}$$

where $C > 0$ is the trade-off between smoothness of the function f and the amount up to which deviations larger than ε are tolerated. In spite of SVR's many advantages, it still remains computationally difficult as the complexity of quadratic (or convex) programming increases with the size of dataset. LSSVR [26] simplifies the formulation by solving the following optimization problem:

$$\min \frac{1}{2}||\mathbf{w}||^2 + \frac{\gamma}{2}\sum_{i=1}^{l} e_i^2$$

$$\text{such that } y_i = \mathbf{w}^T\phi(\mathbf{x}_i) + b + e_i, \tag{7}$$

where γ is the regularization parameter which determines the trade-off between the minimization of training error and smoothness (or flatness) of the function. By using a set of dual variables, the lagrangian function is defined as follows:

$$L = \frac{1}{2}||\mathbf{w}||^2 + \frac{\gamma}{2}\sum_{i=1}^{l} e_i^2 - \sum_{i=1}^{l}\alpha_i\left(y_i - \mathbf{w}^T\phi(\mathbf{x}_i) - b - e_i\right), \tag{8}$$

where L is the Lagrangian and α_i is the Lagrange multiplier. To solve the optimization problem, the derivative of the function with respect to primal and dual variables is set as zero. As a result, we have:

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \rightarrow \mathbf{w} = \sum_{i=1}^{l}\alpha_i\phi(\mathbf{x}_i), \tag{9}$$

$$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^{l}\alpha_i = 0, \tag{10}$$

$$\frac{\partial L}{\partial e_i} = 0 \rightarrow \alpha_i = \gamma e_i, \tag{11}$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \rightarrow \mathbf{w}^T\phi(\mathbf{x}_i) + b + e_i - y_i = 0. \tag{12}$$

By substituting $\mathbf{w}$ and $\mathbf{e}$ into lagrangian, the following linear system is obtained [27]:

$$\begin{bmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & \mathbf{\Omega} + \gamma^{-1}\mathbf{I} \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix}, \tag{13}$$

where:

$$\mathbf{e} = [e_1, e_2, ..., e_l]^T,$$

$$\boldsymbol{\alpha} = [\alpha_1 + \alpha_2, ..., \alpha_l]^T,$$

$$\mathbf{\Omega} = \mathbf{Z}\mathbf{Z}^T,$$

$$\mathbf{Z} = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), ..., \phi(\mathbf{x}_l)]^T,$$

$$\mathbf{y} = [y_1, y_2, ..., y_l]^T,$$

$$1 = [1, 1, ..., 1]^T.$$

The kernel trick can now be applied to forming the matrix $\mathbf{\Omega}$ as follows:

$$\mathbf{\Omega}_{ij} = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j). \tag{14}$$

The solution of Eq. (13) can be formulated as:

$$b = \frac{\mathbf{1}^T(\mathbf{\Omega} + \gamma^{-1}\mathbf{I})^{-1}y}{\mathbf{1}^T(\mathbf{\Omega} + \gamma^{-1}\mathbf{I})^{-1}\mathbf{1}}, \tag{15}$$

$$\boldsymbol{\alpha} = (\mathbf{\Omega} + \gamma^{-1}\mathbf{I})^{-1}(y - b\mathbf{1}). \tag{16}$$

Using b and $\boldsymbol{\alpha}$ obtained from Eqs. (15) and (16) and selecting RBF as the kernel function, the optimal regression function can be written as:

$$f(\mathbf{x}) = \sum_{i=1}^{l}\alpha_i k(\mathbf{x}_i, \mathbf{x}) + b, \tag{17}$$

$$k(\mathbf{x}_i, \mathbf{x}) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}||^2}{2\sigma^2}\right), \tag{18}$$

where σ^2 is called the kernel squared bandwidth. The RBF kernel corresponds to an infinite dimensional function space. In other words, the RBF kernel defines a function space that is a lot larger than that of the linear kernel or the polynomial kernel and is generally more flexible. Model selection in LSSVR is the major problem, because choosing the wrong kernel and regularization parameters can lead to an overfitted model. As a result, the model will have low error rate on training data and high error rate on test data. Hence, to improve the accuracy of LSSVR model, the parameters of the model should be optimized. The parameter optimization in LSSVR model includes the regularization parameter (γ) and kernel squared bandwidth of RBF (σ^2). Previous researchers suggested different parameter settings methods for SMVs. Hsu et al. [28] suggested to set the parameters of SVM, herein γ and σ^2 to 1 and 1/k, respectively, where k represents the number of input patterns. However, Cheng et al. [6] illustrated that this parameter setting method would result in a poor performance. Aiyer et al. [29] determined the parameters of LSSVM by trial and error procedure. Since γ and σ^2 can have any positive real values, the trial and error procedure would be very time-consuming and often would not result in the best parameters. In the present work, the CSA optimization method is used to optimize the parameters γ and σ^2 of LSSVR model.

4.1. Coupled simulated annealing

CSA is a global optimization method based on Simulated Annealing (SA) which can be used to solve a non-convex optimization problem with continuous variables. The SA algorithm was originally inspired from the annealing process which consists of heating the metal, holding its temperature, and then cooling it. In SA algorithm as analogous to thermodynamic annealing process, the temperature is considered as a variable. While the temperature is high, the algorithm has the ability to jump out of any local optimum and accept solutions which are worst than the current solution. By reducing the temperature, the chance of accepting worst solutions decreases. This allows the algorithm to focus on an area of search space in which near-optimum solution can be found. However, the SA algorithm is highly sensitive to the initial temperature and the initial parameters. This prompted the researchers to develop a new global optimization method called CSA, which is consisted of several SA processes with coupled acceptance probabilities. The acceptance probability of a traditional SA algorithm is often given as:

$$A(x \to y) = \frac{1}{1 + \exp\left(\frac{E(y)-E(x)}{T_t^{ac}}\right)}, \tag{19}$$

where x and y denote the current and probing solutions, respectively. $E(.)$ is the cost function and T_t^{ac} is the acceptance temperature at time instant t. While the acceptance probability of SA algorithm depends only on the current and probing solutions, the CSA algorithm considers other current solutions coupled together by their cost functions, featuring a new form of acceptance probability function as follows:

$$A_\Theta(\eta, x_i \to y_i) = \frac{\exp\left(\frac{E(x_i)-\max\limits_{x_i\in\Theta}(E(x_i))}{T_t^{ac}}\right)}{\eta}, \tag{20}$$

and:

$$\eta = \sum_{\forall x \in \Theta} \exp\left(\frac{E(x)-\max\limits_{x_i\in\Theta}(E(x_i))}{T_t^{ac}}\right), \tag{21}$$

where Θ is the set of current states, η is the coupling term, x_i and y_i, $i = 1, 2, ..., m$, with m being the number of elements in Θ, are the current state and the corresponding probing state, respectively. The CSA is able to easily escape from local optima; it improves the quality of solution without slowing down the speed of convergence; above all, it shows an excellent reduction in dependency on initial parameters. More details about the SA and CSA can be found in the paper of Xavier-de-Souza et al. [12]. The flowchart of the CSA-LSSVR model is shown in Figure 3.

In this work, to optimize the parameters of LSSVM based on CSA algorithm, we used a freely available LS-SVMlab v1.8 package [30] which was run in Matlab commercial software. The LS-SVMlab toolbox, first, uses CSA to determine the suitable parameters, then these parameters are given to a second optimization technique (simplex or grid search) to perform a fine-tuning step. Grid search is an exhaustive searching through a manually specified subset of the hyperparameter space of a learning algorithm [31]. To avoid overfitting, a 10-fold cross-validation algorithm is performed in the training process [32]. Note that compared to ANN, the LSSVR model reaches a global minimum due to linear programming (solving a set of linear equations), but the optimization problem to be solved in tuning the kernel and regularization parameters is generally non-convex and may have a local minima.

5. Results and discussion

This section discusses the application of ANN and CSA-LSSVR in modeling the Compressive Strength of Concrete (CSC) with the following eight factors:

1. Cement (kg/m^3);
2. Blast furnace slag (kg/m^3);
3. Fly ash (kg/m^3);
4. Water (kg/m^3);
5. Superplasticizer (kg/m^3);
6. Coarse aggregate (kg/m^3);
7. Fine aggregate (kg/m^3);
8. Age (day).

RMSE and R-squared coefficient are selected as criteria to evaluate and compare the performance of ANN and CSA-LSSVR models. RMSE is used as a measure of differences between the values predicted by the model and the values observed in the lab. R-squared is a measure of how well the independent variables considered account for the measured dependent variable [13]. The higher the R^2 value, the better

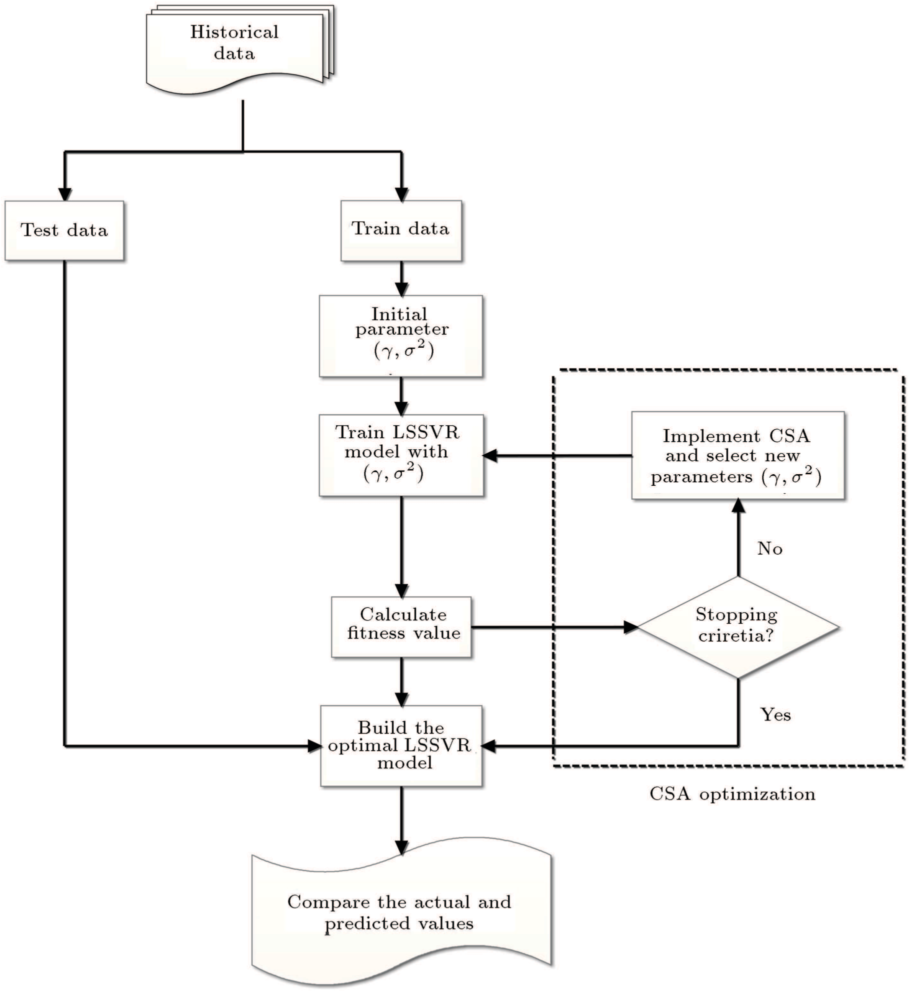
Historical data
Test data
Train data
Initial parameter (γ,σ^2)
Train LSSVR model with (γ,σ^2)
Implement CSA and select new parameters (γ,σ^2)
No
Calculate fitness value
Stopping criretia?
Yes
Build the optimal LSSVR model
CSA optimization
Compare the actual and predicted values

Figure 3. Flowchart representing the CSA-LSSVR algorithm.

the prediction relationship. These two criteria are calculated as follows:

$$\text{RMSE}=\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i-\hat{y}_i)^2}, \tag{22}$$

$$R^2=\frac{\left(N\sum\limits_{i=1}^{N}y_i\hat{y}_i-\sum\limits_{i=1}^{N}y_i\sum\limits_{i=1}^{N}\hat{y}_i\right)^2}{\left[N\sum\limits_{i=1}^{N}(y_i^2)-\left(\sum\limits_{i=1}^{N}y_i\right)^2\right]\left[N\sum\limits_{i=1}^{N}(\hat{y}_i^2)-\left(\sum\limits_{i=1}^{N}\hat{y}_i\right)^2\right]}, \tag{23}$$

where y_i is the observed output, $\hat{y}_i$ is the model output, and N is the number of data points. To model the concrete strength, first, an MLP neural network has been trained using the train and validation datasets. During the training of ANN, the validation error begins to rise as the training error continues to fall. In this stage, the training stops and the weights and biases at the minimum validation error are selected as the optimum parameters. The primary drawback of ANN is the considerable time needed to determine the number of hidden neurons, which requires repetitive trial and error-tuning processes. The first disadvantage of ANN is the considerable time needed to determine the number of hidden neurons which requires repetitive trial and error-tuning processes and is the primary drawback of ANN. The second disadvantage of an MLP neural network based on back-propagation algorithm is that with different initial weights, it gets trapped in different local minimums. To solve this issue, we trained the network with 10 different initial weights and biases and selected the model with least RMSE on validation dataset.

In comparison to ANN, the advantage of the presented hybrid model is the absence of local minimums in optimizing the parameters **w** and b of regression function. Also, the presented hybrid model features a CSA-based approach which optimizes the combination of γ and σ^2 by the use of CSA and grid search optimization method. Solving the non-convex optimization problem of tuning the regularization and kernel parameters with CSA algorithm gives the presented model

Table 3. Root mean squared error of ANN and CSA-LSSVR models.

Experiment	MLP		CSA-LSSVR	
	RMSE of training dataset	RMSE of test dataset	RMSE of training dataset	RMSE of test dataset
1	2.95	5.32	3.16	5.57
2	3.46	6.17	2.97	6.23
3	3.46	4.73	3.64	4.89
4	3.26	5.69	3.06	6.27

Table 4. The coefficient of determination R^2 of ANN and CSA-LSSVR models.

Experiment	MLP		CSA-LSSVR	
	R^2 of training dataset	R^2 of test dataset	R^2 of training dataset	R^2 of test dataset
1	0.9686	0.8766	0.9644	0.8897
2	0.9572	0.8681	0.9688	0.8608
3	0.9570	0.9209	0.9541	0.9167
4	0.9621	0.8832	0.9668	0.8634

enormous computational advantage over other existing SVM methods. A 10-fold cross-validation method was used to select the parameters with best cross-validation accuracy. One fold was retained as the validation data, while the remaining 9 folds were used as training data. For dataset of Experiment 3, the parameters are selected as $\gamma = 23.88$ and $\sigma^2 = 6.8$. The performance of ANN and CSA-LSSVR in modeling the concrete compressive strength is compared in Tables 3 and 4.

Figures 4 and 5 illustrate the scatter diagram of ANN and CSA-LSSVR models in predicting the values of Experiment 3. The results presented in Tables 3 and 4 show that neural network models are supported better by the experimental data than the regression analysis based on LSSVM which is similar to results of Chou et al. [33]. Hence, the ANN of MLP type can model the nonlinear behavior of concrete compressive strength with slightly better accuracy. Computational time on a 2.4 GHz core i5 processor with 8 GB RAM using Matlab, for training and implementation of CSA-

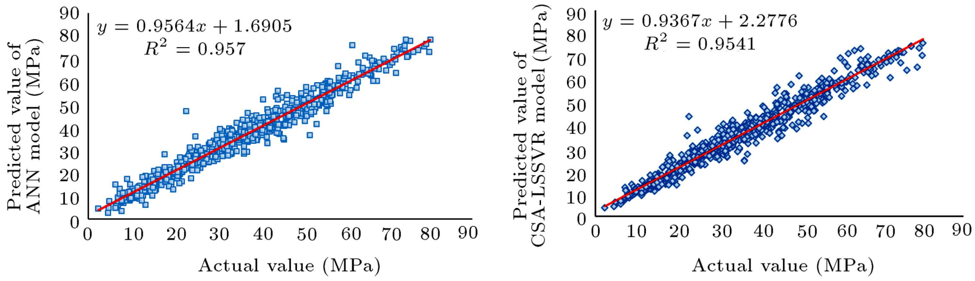

Figure 4. Scatter diagram of the measured versus the predicted values of compressive strength with ANN and CSA-LSSVR models for training set.

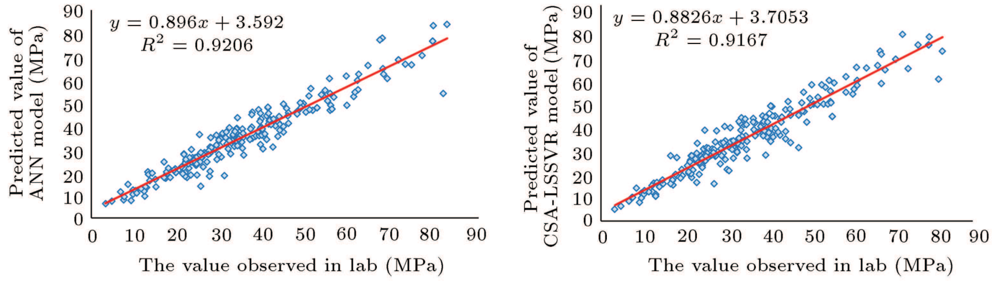

Figure 5. Scatter diagram of the measured versus the predicted values of compressive strength with ANN and CSA-LSSVR models for testing set.

LSSVR model takes about 45.1 s and for ANN model takes about 495.2 s. It is because the ANN often converges to a local minimum, hence it has to be trained with different initial weights, which is time-consuming. While the LSSVR models always reach the global optimum due to linear programming. The LSSVR models reduce the risk of overfitting by using the structural risk minimization strategy, while the ANN models exclude or reduce this effect by using other developed techniques such as early stopping method.

6. Conclusions

In this study, a hybrid model was developed using a LSSVR based on CSA optimization method for predicting complex behavior of concrete compressive strength. The results from the training and testing stages of the proposed hybrid model were compared with those obtained by ANN model using RMSE and R^2 as evaluation criteria. The R-squared coefficient for testing set was in the range of 0.8681-0.9209 for ANN and 0.8634-0.9167 for CSA-LSSVR model. The values of RMSE are in range of 4.73-6.17 for ANN and 4.89-6.27 for CSA-LSSVR model in test set. The results have shown that in comparison with CSA-LSSVR model, the ANN of multilayer perceptron type can model the nonlinear behavior of concrete compressive strength with slightly better accuracy. However, the major disadvantage of ANN model was the existence of many local minimums. As a result, different initial weights would generate different models with different accuracy. On the other hand, determining the number of neurons in the hidden layer of ANN model by trial and error procedure is a time-consuming task. While the CSA-LSSVR model will always converge to a global minimum in optimizing the parameters $\mathbf{w}$ and $\mathbf{b}$ of regression function. Also, the LSSVR model needs to optimize only two parameters of γ and σ^2 to improve its accuracy, which gives it an enormous computational advantage over other methods. The weakness includes the need for a good kernel function. The computational time of these two models were compared, and it was shown that the overall training and implementation time was less for CSA-LSSVR models than ANNs. The conclusions have shown that both ANN and CSA-LSSVR models have high potential for predicting the compressive strength of high performance concrete.

Acknowledgment

The authors would like to thank Professor I-Cheng Yeh for providing the HPC database.

References

1. Russell, H.G. "ACI defines high-performance concrete", *Concrete International*, **21**, pp. 56-57 (1999).
2. Bharatkumar, B., Narayanan, R., Raghuprasad, B. and Ramachandramurthy, D. "Mix proportioning of high performance concrete", *Cement and Concrete Composites*, **23**, pp. 71-80 (2001).
3. Lim, C.-H., Yoon, Y.-S. and Kim, J.-H. "Genetic algorithm in mix proportioning of high-performance concrete", *Cement and Concrete Research*, **34**, pp. 409-420 (2004).
4. Aïtcin, P.-C., *High Performance Concrete*, CRC Press (2011).
5. Ni, H.-G. and Wang, J.-Z. "Prediction of compressive strength of concrete by neural networks", *Cement and Concrete Research*, **30**, pp. 1245-1250 (2000).
6. Cheng, M.-Y., Chou, J.-S., Roy, A.F. and Wu, Y.-W. "High-performance concrete compressive strength prediction using time-weighted evolutionary fuzzy support vector machines inference model", *Automation in Construction*, **28**, pp. 106-115 (2012).
7. Yeh, I.-C. "Modeling concrete strength with augment-neuron networks", *Journal of Materials in Civil Engineering*, **10**, pp. 263-268 (1998).
8. Prasad, B.R., Eskandari, H. and Reddy, B.V. "Prediction of compressive strength of SCC and HPC with high volume fly ash using ANN", *Construction and Building Materials*, **23**, pp. 117-128 (2009).
9. Sobhani, J., Najimi, M., Pourkhorshidi, A.R. and Parhizkar, T. "Prediction of the compressive strength of no-slump concrete: a comparative study of regression, neural network and ANFIS models", *Construction and Building Materials*, **24**, pp. 709-718 (2010).
10. Alshihri, M.M., Azmy, A.M. and El-Bisy, M.S. "Neural networks for predicting compressive strength of structural light weight concrete", *Construction and Building Materials*, **23**, pp. 2214-2219 (2009).
11. Topcu, I.B. and Sarıdemir, M. "Prediction of compressive strength of concrete containing fly ash using artificial neural networks and fuzzy logic", *Computational Materials Science*, **41**, pp. 305-311 (2008).
12. Xavier-de-Souza, S., Suykens, J.A., Vandewalle, J. and Bollé, D. "Coupled simulated annealing", *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, **40**, pp. 320-335 (2010).
13. Yeh, I.-C. "Modeling of strength of high-performance concrete using artificial neural networks", *Cement and Concrete Research*, **28**, pp. 1797-1808 (1998).
14. Rad, M.A.A. and Yazdanpanah, M.J., *Designing Supervised Local Neural Network Classifiers Based on EM Clustering for Fault Diagnosis of Tennessee Eastman Process*, Chemometrics and Intelligent Laboratory Systems (2015).
15. Rehman, A. and Saba, T. "Neural networks for document image preprocessing: state of the art", *Artificial Intelligence Review*, **42**, pp. 253-273 (2014).

16. Shu, J., Zhang, Z., Gonzalez, I. and Karoumi, R. "The application of a damage detection method using artificial neural network and train-induced vibrations on a simplified railway bridge model", *Engineering Structures*, **52**, pp. 408-421 (2013).

17. El Kadi, H. "Modeling the mechanical behavior of fiber-reinforced polymeric composite materials using artificial neural networks - A review", *Composite Structures*, **73**, pp. 1-23 (2006).

18. Ghaboussi, J. and Joghataie, A., "Active control of structures using neural networks", *Journal of Engineering Mechanics*, **121**(4), pp. 555-567 (1995).

19. Taormina, R., Chau, K.-W. and Sethi, R. "Artificial neural network simulation of hourly groundwater levels in a coastal aquifer system of the Venice lagoon", *Engineering Applications of Artificial Intelligence*, **25**, pp. 1670-1676 (2012).

20. Ji, T., Lin, T. and Lin, X. "A concrete mix proportion design algorithm based on artificial neural networks", *Cement and Concrete Research*, **36**, pp. 1399-1408 (2006).

21. Hornik, K. "Approximation capabilities of multilayer feedforward networks", *Neural Networks*, **4**, pp. 251-257 (1991).

22. Suratgar, A.A., Tavakoli, M.B. and Hoseinabadi, A. "Modified Levenberg-Marquardt method for neural networks training", *World Acad. Sci. Eng. Technol*, **6**, pp. 46-48 (2005).

23. Cortes, C. and Vapnik, V. "Support-vector networks", *Machine Learning*, **20**, pp. 273-297 (1995).

24. Press, W.H., Numerical Recipes 3rd Edition: *The Art of Scientific Computing:* Cambridge University Press (2007).

25. Smola, A.J. and Schölkopf, B. "A tutorial on support vector regression", *Statistics and Computing*, **14**, pp. 199-222 (2004).

26. Suykens, J.A., Van Gestel, T., De Moor, B. and Vandewalle, J., *Least Squares Support Vector Machines*, **4**, World Scientific (2002).

27. Guo, Z. and Bai, G. "Application of least squares support vector machine for regression to reliability analysis", *Chinese Journal of Aeronautics*, **22**, pp. 160-166 (2009).

28. Hsu, C.W., Chang, C.C. and Lin, C.J., *A Practical Guide to Support Vector Classification*, Department of Computer Science, Taipei 106, Taiwan (2003).

29. Aiyer, B.G., Kim, D., Karingattikkal, N., Samui, P. and Rao, P.R. "Prediction of compressive strength of self-compacting concrete using least square support vector machine and relevance vector machine", *KSCE Journal of Civil Engineering*, **18**, pp. 1753-1758 (2014).

30. http://www.esat.kuleuven.be/sista/lssvmlab/.

31. Bergstra, J. and Bengio, Y. "Random search for hyper-parameter optimization", *The Journal of Machine Learning Research*, **13**, pp. 281-305 (2012).

32. Chou, J.-S. and Pham, A.-D. "Hybrid computational model for predicting bridge scour depth near piers and abutments", *Automation in Construction*, **48**, pp. 88-96 (2014).

33. Chou, J.-S., Chiu, C.-K., Farfoura, M. and Al-Taharwa, I. "Optimizing the prediction accuracy of concrete compressive strength based on a comparison of data-mining techniques", *Journal of Computing in Civil Engineering*, **25**, pp. 242-253 (2010).

Biographies

Mostafa Ali Ayubi Rad received his BS degree in Electrical Engineering from Shiraz University of Technology, Shiraz, Iran, in 2012, and his MS degree from University of Tehran, Iran, in 2015. His field of research includes engineering applications of artificial intelligence and application of various nonlinear controllers on power converters.

Mohammad Sadegh Ayubi Rad is a PhD student in Department of Civil and Environmental Engineering at Shiraz University, Shiraz, Iran. He holds BS degree in Civil Engineering from Shahid Bahonar University of Kerman, Kerman, Iran, and MS degree from Malayer University, Malayer, Iran. His research interests include structural optimization, reliability-based design, and seismic design of structures.