Computers and Concrete, Vol. 11, No. 4 (2013) 337-350
DOI: http://dx.doi.org/10.12989/cac.2013.11.4.337

Support vector machine for prediction of the compressive strength of no-slump concrete

J. Sobhani*[1], M. Khanzadi[2] and A.H. Movahedian[2]

[1]Department of Concrete Technology; Road, Housing & Urban Development Research Center (BHRC), Pas Farhangian St, Sheikh Fazlollah Exp. Way, Tehran , P.O. Box 13145-1696, Iran
[2]Department of Civil Engineering, Iran University of Science and Technology, Tehran, P.O. Box 16846-13114, Iran

(Received September 7, 2011, Revised February 12, 2012, Accepted October 9, 2012)

Abstract. The sensitivity of compressive strength of no-slump concrete to its ingredient materials and proportions, necessitate the use of robust models to guarantee both estimation and generalization features. It was known that the problem of compressive strength prediction owes high degree of complexity and uncertainty due to the variable nature of materials, workmanship quality, etc. Moreover, using the chemical and mineral additives, superimposes the problem's complexity. Traditionally this property of concrete is predicted by conventional linear or nonlinear regression models. In general, these models comprise lower accuracy and in most cases they fail to meet the extrapolation accuracy and generalization requirements. Recently, artificial intelligence-based robust systems have been successfully implemented in this area. In this regard, this paper aims to investigate the use of optimized support vector machine (SVM) to predict the compressive strength of no-slump concrete and compare with optimized neural network (ANN). The results showed that after optimization process, both models are applicable for prediction purposes with similar high-qualities of estimation and generalization norms; however, it was indicated that optimization and modeling with SVM is very rapid than ANN models.

Keywords: no-slump concrete; compressive strength; prediction; support vector machine; neural networks; optimization

1. Introduction

The compressive strength of concrete is known as the most important mechanical property which is generally obtained by measuring concrete specimen's strength after a standard curing of 28 days (Hong-Guang 2000). The problem of compressive strength prediction owes a high degree of complexity and uncertainty due to the variable nature of constituent materials, workmanship quality, etc. It should be noted that the involved complexity superimposed by application of chemical and mineral additives which commonly used to modify the fresh and hardened properties of concrete. Linear or nonlinear regression models are usually utilized to predict the compressive strength of concrete (Hong-Guang 2000, Sobhani *et al*. 2010). In general, these models comprise lower accuracy and in most cases they fail to meet the extrapolation accuracy and generalization

*Corresponding author, Ph.D., E-mail: ja_sobhani@yahoo.com, sobhani@bhrc.ac.ir

requirements (Zain and Abd 2009). Recently, artificial intelligence-based robust systems have been successfully implemented in this area; like the neural networks (Sobhani *et al.* 2010, Hong-Guang 2000, Oztas *et al.* 2006, Bilim *et al.* 2009), fuzzy systems (Ozcan *et al.* 2009, SarIdemir 2009), adaptive network-based fuzzy inference systems (Sobhani *et al.* 2010, Ramezanianpour *et al.* 2004), neuro-fuzzy polynomials (Fazel Zarandi *et al.* 2008). The obtained results demonstrate the superior prediction performance of these models in comparison with the traditional ones. On the other hand, the need for better prediction norms motivated the researchers toward the utilization of more advanced models with higher accuracy and extrapolations features. Among these methods, support vector machines, based on the structural risk minimization principle (Vapnik 2000, Chou *et al.* 2011), seem to be a promising method for modeling and prediction of concrete's compressive strength as successfully applied in the area of predicting the fracture parameters of concrete (Samui and Kim 2012), early warning of hazard for pipelines (Wan and Mita 2010), data mining (Tinoco *et al.* 2010), knowledge discovery (Acevedo-Rodriguez *et al.* 2009), structural/non-structural components (Yan and Shi 2010, Chen *et al.* 2009, Tong *et al.* 2008, Mashford and Marlow 2010, Pal and Deswal 2011), properties of materials (Das 2011, Wang 2006), and construction management (An *et al.* 2007, Lam *et al.* 2009, Cheng and Wu 2009).

The focus of this paper is on the prediction of the 28-days compressive strength (28-CS) of a special type of concrete known as no-slump concrete with optimized support vector machines (SVMs) and compares it with optimized neural network (ANN) models. No-slump concrete which also known as dry cast concrete is commonly defined as concrete having slump in a range of 0-25 mm (ACI 211.3 2002, Shelestynsky 1972). The physical properties of no-slump concrete, specifically the compressive strength, are very sensitive to its ingredients and mix proportions, so, predicting the compressive strength of no-slump concrete is a highly complicated problem that requires more accurate and reliable methods for strength prediction.

The rest of the paper is organized as follows: In Section 2 the methods of modeling including SVM and ANN are briefly discussed. Section 2 is devoted to the experimental program conducted to prepare the required data for training and testing of proposed models. Moreover in this section the methodology of the optimization process is presented. In Section 5 the results are compared and discussed. Finally section 6 provides conclusions and highlight of this study.

2. Methods of modeling

2.1 Support vector machine

Support vector machine (SVM) is a supervised learning method to analyze the data for classification and regression problems. With a given a set of training patterns, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other. More formally, a support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite- dimensional space, which can be used for classification, and regression in many tasks like image retrials, financial research, etc.

Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier (Cortes and Vapnik 1995).

Vapnik (Vapnik 2000) proposed ε-support vector regression (SVR) by introducing an alternative ε-insensitive loss function (Vapnik 1998). Generally speaking, the purpose of the SVR is to find a function having at most ε deviation from the actual target vectors for all given training data and have to be as flat as possible (Pal and Deswal 2011). A brief description of the SVR methodology is presented as follows:

Let the n array vector x_i have real value y_i, and let $F(x)$, be a set of real functions that contains the regression function $f_0(x)$. Considering the problem of approximating the set of data, $\{(x_1, y_1),...(x_n, y_n)\}$ with a linear function, $f(x) = (w.x) + b$ the optimal regression function is given by minimizing the empirical risk R

$$R = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - f(x_i)\right|_\varepsilon \tag{1}$$

With ε-insensitive loss function

$$\left|y - f(x)\right|_\varepsilon = \begin{cases} \text{if} & \left|y - f(x)\right| \le \varepsilon \quad 0 \\ \text{otherwise} & \left|y - f(x)\right| - \varepsilon \end{cases} \tag{2}$$

Now the objective is to find a function $f(x)$ with minimum deviation of ε from the actual observed targets y_i for all of the training data and at the same time it is as flat as possible.

This is equivalent to minimizing the following function

$$R(w) = \frac{\left\|w^2\right\|}{2} + C \times \frac{1}{n}\sum_{i=1}^{n}\left|y_i - f(x_i)\right|_\varepsilon \tag{3}$$

Where the first term $\left\|w^2\right\|/2$ considering the flatness of function and the second term $\frac{1}{n}\sum_{i=1}^{n}\left|y_i - f(x_i)\right|_\varepsilon$ calculate empirical risk the C is a penalty value that tune trade off between empirical risk and flatness of function with the larger C factor the training error was decreased but the generalization performance of the function was decreased as well. Eq. (3) could be represented as the dual optimization problem and this optimization problem can be solved with Lagrange method as follows

$$L_2 = \sum_{i=1}^{l} y_i(\alpha_i^* - \alpha_i) - \varepsilon \times \sum_{i=1}^{l} y_i(\alpha_i^* - \alpha_i) - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_i)(x_i \cdot x_j) \tag{4}$$

Subject to these constrains

$$\sum \alpha_i^* = \sum \alpha_i;\ 0 \le \alpha_i^* \le C;\ 0 \le \alpha_i \le C \tag{5}$$

The training data with nonzero Lagrangian multipliers (α_i^*, α_i) are called support vectors.

The final solution could be as follows

Table 1 Different types of kernel function

Type of classification	Kernel Function
Polynomial degree	$K(x_i \cdot x) = \left[(x^T \cdot x_i) + 1\right]^d$
Gaussian	$K(x \cdot x_i) = e^{-\frac{\lVert x - x_i \rVert^2}{\sigma^2}}$
Multi layer perceptron	$K(x \cdot x_i) = \tanh\left[(x^T \cdot x_i)\right] + b$

$$f(x) = \sum_{i=1}^{n_{sv}} (\alpha_i^* - \alpha_i) x_i + b \tag{6}$$

Where n_{sv} is the number of support vectors

In the real situation linear regression is uncommon, as in the case of most engineering applications, in this case SVM regression is mapping the input data x into a high-dimensional space that is called feature space by a non-linear mapping. In the feature space, linear regression can be done (Yinfeng *et al.* 2008). To this means, SVM uses a kernel function $K(x_i \cdot x_j)$.

So when the optimum values α_i and α_i^* are obtained, the regression function is given by

$$f(x) = \sum_{i=1}^{n_{sv}} (\alpha_i^* - \alpha_i) \cdot K(x_i \cdot x) + b \tag{7}$$

Any function which satisfies Mercer condition (Vapnik 2000) can be used as the kernel function. Some common kernel functions are listed in Table 1. The kernel parameters should be carefully chosen because they have an important role in accuracy of SVM solution and its complexity. The SVMs performance is largely governed by the type of kernel function being used kernel function, choosing an appropriate kernel function and kernel parameters for each application problem is very important in order to guarantee satisfactory results.

It should be noted that tuning of SVM parameters is still a heuristic process and the parameters specified by user are (1) Type of kernel function and the parameters, (2) The value of the penalty factor C and (3) The value of ε-insensitive.

2.2 Artificial neural networks

Artificial neural networks (ANN) are computing systems made up of a number of simple, highly interconnected processing elements (PEs), which process information by their dynamic state response to external inputs (Fausett 1994, Sobhani *et al.* 2010). The architecture of an ANN is composed of an input layer of neurons or namely processing elements (PEs), one or several hidden layers (HLs) of neurons and output layer of neurons. The neighboring layers are fully interconnected by weight. The input layer neurons receive input data and transmit them to the neurons of the hidden layer without performing any calculation. Layers between the input and output layers are called HLs and may contain a large number of hidden processing units. Finally,

Table 2 The chemical and physical properties of cementitious materials

Chemical analysis (%) / Properties	Cement	Silica fume
Calcium oxide (CaO)	61.9	0.6
Silica (SiO_2)	21.2	90.9
Alumina (Al_2O_2)	4.2	0.6
Iron oxide (Fe_3O2)	4.6	0.7
Magnesia (MgO)	3.4	1.3
Sodium oxide (Na_3O)	0.6	0.4
Potassium oxide (K_2O)	0.5	1.1
Sulfur trioxide (SO_2)	1.7	–
Bogue potential compound composition, %		
Tri calcium silicate (C_3S)	52.74	–
Di calcium silicate (C_2S)	20.31	–
Tri calcium aluminate (C_3A)	3.35	–
3 days compressive strength, kg/cm^2	223	–
7 days compressive strength, kg/cm^2	306	–
28 days compressive strength, kg/cm^2	414	–
Initial setting time	150	–
min Final setting time	190	–
min Specific surface, cm^2/g	3296	–

Table 3 The physical and mechanical properties of the aggregates

Type	Specific gravity (g/cm^3)	Absorption (%)	Fineness modulus	Passing from 75 μm sieve
Fine aggregate	2.53	2.6	3.2	1.1
Coarse aggregate	2.56	1.46	-	0.4

the neurons of output layer produce the network predictions (Fausett 1994). Each neuron of a layer other than the input layer computes first a linear combination of the outputs of the neurons of the previous layer, plus a bias. The coefficients of the linear combinations plus the biases are called weights. Then, neurons in the HL apply a nonlinear function as activation function to their inputs (Fausett 1994, Adeloye *et al.* 2006). A systematic algorithm like backpropagation is utilized to tune the connecting weights on the base of the error observed between the real and estimated data till a satisfactory result is achieved.

3. Materials and methods

3.1 Materials

Standard Type II Portland cement (American Society for Testing and Materials 2009a) was used in this study, with silica fume as supplementary cementitious materials. Moreover, siliceous

Table 4 No-slump concrete mix designs

Mix	Cement (kg/m^3)	Silica fume (kg/m^3)	Water (kg/m^3)	Fine aggregation (kg/m^3)	Coarse aggregation (kg/m^3)	Filler (kg/m^3)	*w*/cm	Average compressive strength (MPa)
NSC-1	350	0	95.2	575.9	1273	0	0.27	61.1
NSC-2	350	0	98.5	558.2	1325.4	0	0.28	54
NSC-3	339.5	0	97.7	655.3	1273	10.5	0.28	65.7
NSC-4	339.5	0	97.6	535	1247	10.5	0.28	62.2
NSC-5	336	0	97.6	535	1247	14	0.28	54.5
NSC-6	332.5	0	97.7	655.3	1273	17.5	0.28	63.1
NSC-7	329	0	97.6	535	1247	21	0.28	52.2
NSC-8	325.5	0	97.7	655.3	1273	24.5	0.28	64.1
NSC-9	410	0	117.8	491.2	1273	0	0.29	59.9
NSC-10	350	0	100.9	460.3	1419.8	0	0.29	61.9
NSC-11	350	0	102.6	535	1247	0	0.29	64.2
NSC-12	332.5	17.5	105.6	535	1247	0	0.3	62.2
NSC-13	380	0	118.1	354.2	1440.6	0	0.31	60.5
NSC-14	350	0	107.6	535	1247	0	0.31	61.5
NSC-15	325.5	24.5	107.8	535	1247	0	0.31	65
NSC-16	343	0	107.6	535	1247	7	0.31	61.2
NSC-17	320	0	97.7	671.8	1247	38.5	0.31	63.2
NSC-18	346	27.3	115.6	484	1289	156.3	0.31	76.7
NSC-19	380	0	121.1	502.5	1325.4	0	0.32	67.4
NSC-20	320	0	102.2	679.1	1259.7	19	0.32	62.8
NSC-21	320	0	103.2	665.6	1234.2	57	0.32	60.3
NSC-22	350	0	120.4	526.2	1325.4	0	0.34	63.5
NSC-23	350	0	119	710.6	1121.5	0	0.34	59.6
NSC-24	350	0	120	623.3	1208.7	94	0.34	61.1
NSC-25	252.6	19.6	95	828	1206	0	0.35	66.7
NSC-26	345.2	27.1	129.9	482	1282	155.5	0.35	71.2
NSC-27	375	0	134	1300	600	0	0.36	64
NSC-28	332.5	17.5	129.9	509.8	1325.4	0	0.37	61.4
NSC-29	343	27	136.9	480	1278	154.9	0.37	71.2
NSC-30	252.6	19.6	103.4	836	1063	135	0.38	62.7
NSC-31	258.9	0	98.4	835	1083	135	0.38	55
NSC-32	350	0	139.7	591.3	1145.5	188	0.4	58.3

filler with more than 99.0% SiO_2 was used as neutral filler. The chemical and physical properties of cement and silica fume are shown in Table 2.

Fine river sand and crushed stone with properties shown in Table 3 were used as fine and coarse aggregates. Table 4 shows concrete mix design which was used for production of no-slump concrete.

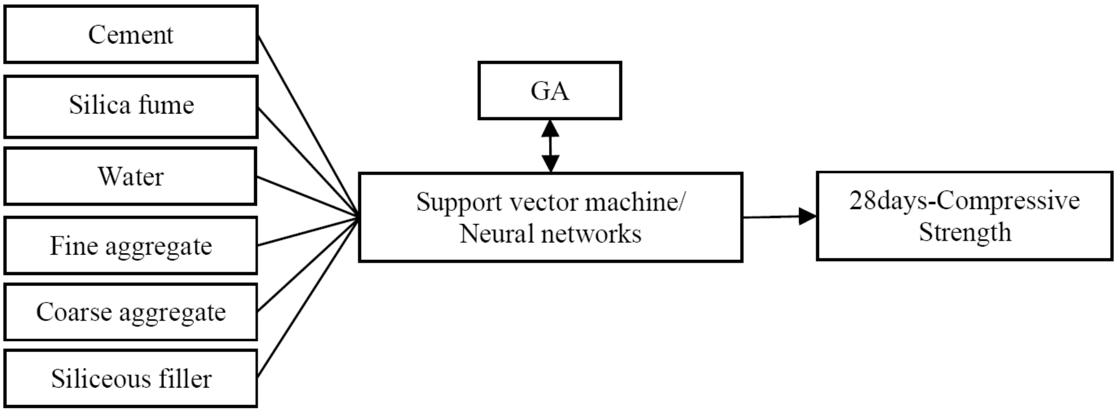

Fig. 1 The schematic structure of the system

Table 5 Range of inputs and output

Input variables	Range	
	Minimum	Maximum
Cement (kg/m^3)	252.6	410
Silica fume (kg/m^3)	0	27.3
Water (kg/m^3)	95	139.7
Fine aggregate (kg/m^3)	354.2	1300
Coarse aggregate (kg/m^3)	600	1440.6
Filler (kg/m^3)	0	188
Water to cementitious material	0.27	0.4
Output (Target value)		
28 days-compressive strength (MPa)	50	78

3.2 Database

To collect the required data for training and testing the models, three samples of each mix design shown in Table 4 were provided, and then compressive strength of these samples was determined in 28 days according to the ASTM C39 (American Society for Testing and Materials 2009b). Totally 96 records of no-slump concrete were gathered by the above mentioned procedure. For all models, 79 samples were randomly assigned to training phase and the remaining 17 samples allocated to testing phase.

3.3 Framework for modeler systems

The schematic structure of the modeler systems is shown in Fig. 1. As suggested in this figure, the inputs are the amounts of cement, silica fume, water, fine aggregates, coarse aggregates and filler of each mix design per unit volume of concrete. Range of inputs and output is provided in Table 5.

Matlab software and its corresponding neural network toolbox [Math Works] and an open code Matlab-based toolbox named as SVM-KM (Rakotomamonjy 2005) utilized to construct and training the ANN and SVM models respectively. In both cases Matlab GA toolbox [MathWorks] was used for optimization purposes.

4. Results and discussion

To evaluate the performances of both models, three indexes; mean squared error (MSE), mean absolute error (MAE) and correlation factor (CF) were used. MSE and MAE values are calculated as follows

$$MSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i^{\text{model}} - y_i^{\text{real}})^2}{n}} \tag{8}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i^{\text{model}} - y_i^{\text{real}}\right| \tag{9}$$

Where y_i^{model} is real output and y_i^{real} is predictive output and n is the number of samples.

The correlation factor (CF) is the value that represent the amount of dependency between two datasets; CF value of 0 is represent the complete independency between datasets and the value of 1 represent complete dependency and it means two data sets have same direction. The CF calculated as follows

$$CF(Y^{\text{model}}, Y^{\text{real}}) = \frac{COV(Y^{\text{model}}, Y^{\text{real}})}{\sqrt{COV(Y^{\text{model}}, Y^{\text{model}}) \times COV(Y^{\text{real}}, Y^{\text{real}})}} \tag{10}$$

Where $Y^{\text{model}} = (y_1^{\text{model}}, y_2^{\text{model}}, ... y_n^{\text{model}})$ and $Y^{\text{model}} = (y_1^{\text{real}}, y_2^{\text{real}}, ... y_n^{\text{real}})$, and $COV(Y^{\text{model}}, Y^{\text{real}})$ is the covariance of Y^{model} and Y^{reral}.

4.1 Modeling with SVM

As mentioned, the tuning of SVM parameters is a heuristic process, so to achieve the optimized SVM a GA (Table 6) was utilized. Gaussian kernel was used in proposed model so the optimizing parameters were as C (Penalty factor), ε (ε-insensitive zone radius) and σ (standard deviation of Gaussian function). The fitness function of GA was chosen as Root Mean Squared Error (RMS) between real 28 days compressive strength and model prediction. Fig. 2 shows a fitness plot for 50 populations. CPU time for GA tuning process is 65 seconds which demonstrates a good optimal point in the optimization process. The optimized SVM was then utilized for training and testing purposes with respect to the collected data sets. Table 7 summarizes the performance indexes for training and testing data sets. Correlation factor achieved with both training and testing data sets were as 0.982 and 0.9473 respectively which demonstrate a higher correlation of experimental observations with SVM-predicted results. The results for MSE and

Table 6 Configuration of GA for optimization process

Item	Parameter value/method
Population size	50
Max generation	100
Selection method	Stochastic Uniform
Crossover method	Scattered
Crossover rate	0.8
Mutation rate	0.1

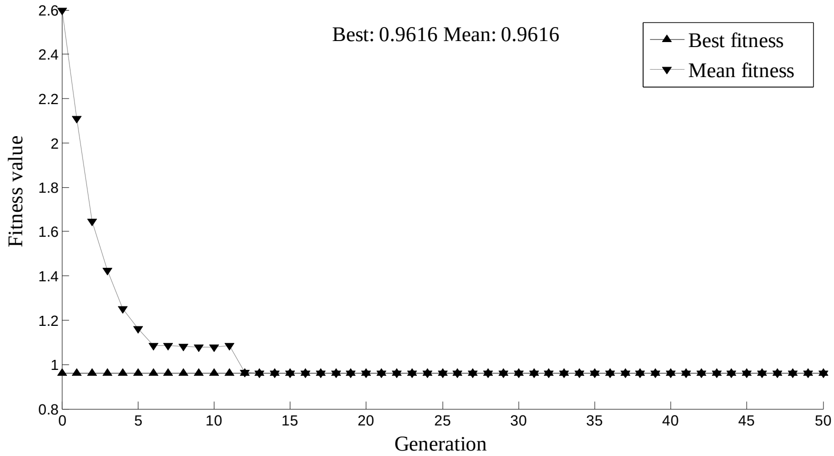

Fig. 2 GA tuning of SVM parameters

Table 7 SVM model results for training and testing data sets

Training set			Testing set			Run Time
CF	MSE	MAE	CF	MSE	MAE	Parameter optimization: 60 seconds*
0.982	0.9616	0.7536	0.9473	2.479	1.0941	

* For all of optimizations PC with 2.53 GHz Core2Dou CPU and 4 GB Ram was used

MAE for both training and testing data sets demonstrate a higher interpolation and extrapolation accuracy and generalization properties respectively.

4.2 Modeling with neural networks

Similar to SVM, an optimization procedure was applied to determine the optimal neural network considering the same database.

For the case of neural network model, a two-staged optimization procedure was adopted. At first stage, a topology optimization was carried out to construct the optimal architecture and then neuron's weight was optimized. In the topology optimization phase (Sobhani *et al.* 2010, Uysal and Harun Tanyildizi 2011, Oztas *et al.* 2006), an initial neural network architecture having two hidden layers was adopted with variable number of PEs which designated to be trained by Levenberg–Marquardt back propagation algorithm. Log-sigmoid and linear transfer functions were

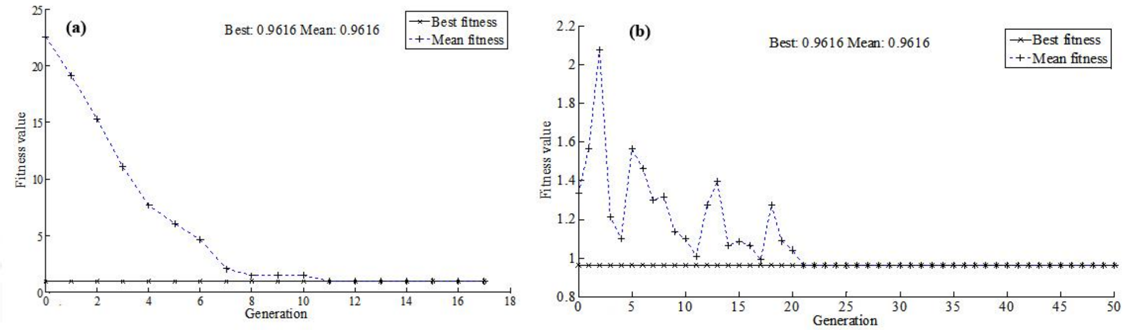

Fig. 3 (a) ANN topology optimization and (b) ANN initial weight optimization

Table 8 ANN model results for training and testing data sets

Training set			Testing set			Run time
CF	MSE	MAE	CF	MSE	MAE	Topology optimization: 100 min*
0.982	0.9616	0.7536	0.9473	2.48	1.0941	Weight optimization: 150 min*

*For all of optimizations PC with 2.53 GHz Core2Dou CPU and 4 GB Ram was used

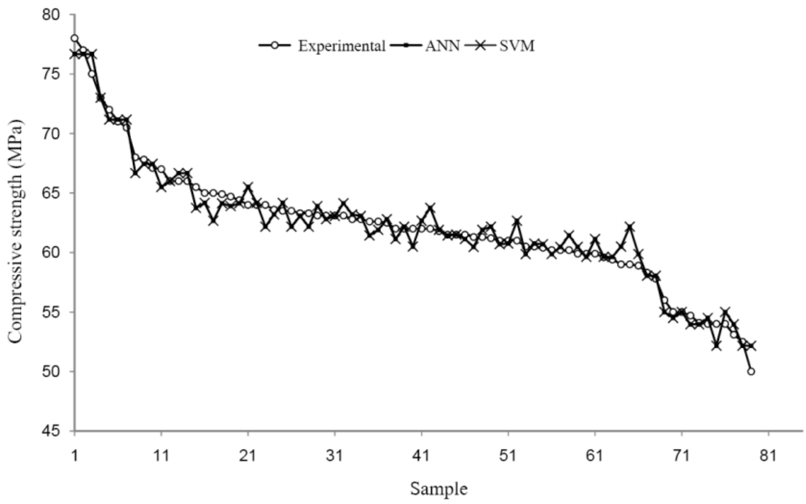

Fig. 4 Comparison of SVM and ANM predictions with experimental observations for training data set

also utilized in hidden and output layers respectively. The mean square error of neural network model was considered as a genetic algorithm's fitness function. On this basis, number of neurons in each hidden layer as parallel processing elements and then their initial weight were optimized using GA setting (Table 6) similar to the one used for SVM optimization process (Adeloye and Munari 2006, Hagan and Menhaj 1994). Figs. 3(a) and (b) shows the progress of GA optimization for topology optimization phase and initial weight optimization respectively. After completion of optimization process which takes 100 min for topology phase and 150 min for weight tuning phase, the optimized architecture was found to be comprising of 3 and 7 neurons in the first and second

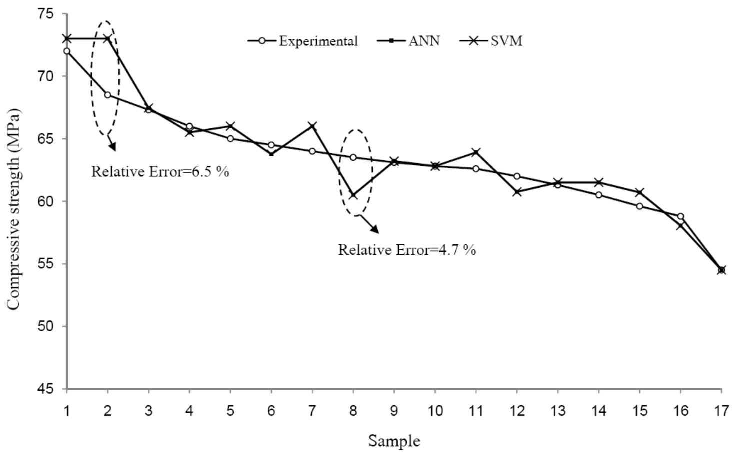

Fig. 5 Comparison of SVM and ANM predictions with experimental observations for testing data set

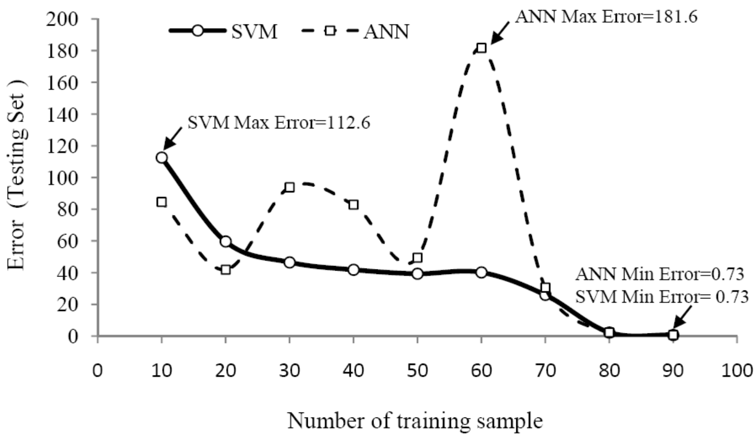

Fig. 6 Comparison between SVM and ANN in facing with different size of training data set

hidden layer respectively. The results of predictions by optimized ANN are presented in Table 8 which demonstrates a quite similar result comparing the optimized SVM model in Table 7.

4.3 Comparing SVM and ANN

Tables 7 and 8 indicate similar results for optimized SVM and optimized ANN model respectively. The only difference is very large run-time of ANN model in comparison to SVM. The optimization procedure of SVM takes only about 60 seconds; however, it needs about 100 min and 150 min for topology optimization and weight optimization process respectively with ANN model. The observed difference could be related to the fact that the learning process of SVM is a direct solution with no need of trial and error procedure as needed for ANN. Another reason could be related to the fact that SVM requires few parameters in comparison to ANN.

Moreover comparisons of SVM and ANN prediction results with experimental observations are depicted in Fig. 4 and 5 respectively for training and testing data set. As seen, SVM and ANN models were satisfactory in the prediction of compressive strength of no-slump concrete both for training and testing data set.

Some errors could be observed in Fig. 5 with maximum values corresponding to the samples 2 and 8 with relative error of 6.5% and 4.7% respectively. It should be noted that these amounts of error values are very low and might be negligible in practical situations.

4.4 Effects of varied number of training-testing sets on performance of predictions

To study the effects of the size of training data, several training sets were created to test both SVM and ANN models. Nine dataset containing (10,86), (20,76), (30,66), (40,56), (50,46), (60,36), (70,26) , (80,16), (90,6) samples were used for training purposes for both SVM and ANN models. In each training run the SVM and ANN models were optimized for corresponding training data set size. For example when the data set is composed of (50,46) data group set, then 50 data used for training and optimization task.

The results suggest that the error norms of both optimized SVM and ANN are similar, but the state of testing for SVM and ANN are quite different. Fig. 6 depicts the testing error for SVM and ANN. It was observed that the maximum errors for testing set were 112.6 and 181.6 respectively for SVM and ANN. This suggests that ANN is more sensitive to the variation in training and testing set in comparison to SVM. However SVM is more robust with a continuous decreasing trend of error amount. The robustness of SVM might be related to its design that aids in optimization process with less parameters and structures.

5. Conclusions

In this paper, optimized support vector machines and optimized neural network models were developed for prediction of the compressive strength of no-slump concrete. It was found that both proposed ANN and SVM models are able to predict the compressive strength of no-slump concrete with a high degree of accuracy and generalization quality; however, SVM is preferred due to its unique and robust features summarized as follows:

(1) Unit architecture of SVM simplifies its application comparing with the ANN.

(2) SVM models include lower parameters for optimization purposes than ANN models which facilitate the optimization procedure.

(3) In general, SVM learn samples faster than ANN. In fact SVM tries to find a direct solution for estimated regression function; however, ANN deals with a time-consuming procedure of multiple connecting-weight adjustments.

(4) The results of this study suggest that the SVM models are more robust and less sensitive than ANN models when encountering the problems with a deficient training data.

References

ACI 211.3 (2002), *Guide for selecting proportions for no-slump concrete*, Farmington Hills (MI).

Adeloye, A. and Munari, A. (2006), “Artificial neural network based generalized storage–yield–reliability models using the Levenberg–Marquardt algorithm”, *J. Hydrol.*, **326**(1-4), 215-230.

Acevedo-Rodriguez, J., Maldonado-Bascon, S., Lafuente-Arroyo, S., Siegmann, P. and Lopez-Ferreras, F. (2009), "Computational load reduction in decision functions using support vector machines", *Signal Process*, **89**(10), 2066-2071.

American Society for Testing and Materials, (2009a), *ASTM C150/C150M–09 Standard specification for Portland cement*, *Annual Book of ASTM Standard Vol. 4.01*, Philadelphia.

American Society for Testing and Materials, (2009b), *ASTM C39/C39M-09a Standard test method for compressive strength of cylindrical concrete specimens*, *Concrete Specimens*, *Annual Book of ASTM Standard Vol. 4.01*, Philadelphia.

An, S.H., Park, U.Y., Kang, K., Cho, M.Y. and Cho, H.H. (2007), "Application of support vector machines in assessing conceptual cost estimates", *ASCE J. Comput. Civil Eng*., **21**(4), 259-264.

Bilim, C., Atiş, C.D., Tanyildiz, H. and Karahan, O. (2009), "Predicting the compressive strength of ground granulated blast furnace slag concrete using artificial neural network", *Adv. Eng. Softw.*, **40**(5), 334-340.

Chen, B.T., Chang, T.P., Shih, J.Y. and Wang, J.J. (2009), "Estimation of exposed temperature for fire-damaged concrete using support vector machine", *Comput. Mater. Sci*., **44**(3), 913-920.

Cheng, M.Y. and Wu, Y.W. (2009), "Evolutionary support vector machine inference system for construction management", *Automat. Constr.*, **18**(5), 597-604.

Chou, J.S., Chiu, C.K., Farfoura, M. and Al-Taharwa, I. (2011), "Optimizing the prediction accuracy of concrete compressive strength based on a comparison of data-mining techniques", *J. Comput Civil Eng.*, **25**(3), 242-254.

Cortes, C. and Vapnik, V. (1995), "Support-vector networks", *Mac. Learn.*, **20**(3), 273-297.

Das, S.K., Samui, P. and Sabat, A.K. (2011), "Prediction of field hydraulic conductivity of clay liners using artificial neural network and support vector machine", *ASCE Int. J. Geomech.*, 12(5), 606-611.

Fausett, L.V. (1994), *Fundamentals of neural networks: architectures, algorithms, and applications*, Prentice Hall.

Fazel Zarandi, M., Türksen, I.B., Sobhani, J. and Ramezanianpour, A.A. (2008), "Fuzzy polynomial neural networks for approximation of the compressive strength of concrete", *Appl. Soft Comput.*, **8**(1), 488-498.

Ghaboussi, J., Garrett, J.H. and Wu, X. (1991), "Knowledge-based modeling of material behavior with neural networks", *J. Eng. Mech.-ASCE*, **117**(1), 117-139.

Hagan, M.T. and Menhaj, M.B. (1994), "Training feedforward networks with the Marquardt algorithm", *IEEE T. Neural Networ.*, **5**(6), 989-993.

Lam, K.C., Palaneeswaran, E. and Yu, C.Y. (2009), "A support vector machine model for contractor prequalification", *Automat. Constr*., **18**, 321-329.

Mashford, J. and Marlow, D. (2010), "Prediction of sewer condition grade using support vector machines", *J. Comput. Civil Eng.*, **25**(4), 283-290.

Neural Network Toolbox (MathWorks-a), *www.mathworks.com/help/toolbox/nnet/*

Global Optimization Toolbox (MathWorks-b), *www.mathworks.com/products/gads*/

Ni Hong-Guang, W.J.Z. (2000), "Prediction of compressive strength of concrete by neural networks", *Cement Concrete Res.*, **30**(8), 1245-1250.

Ozcan, F., Atiş, C.D., Karahan, O., Uncuoğlu, E. and Tanyildizi, H. (2009), "Comparison of artificial neural network and fuzzy logic models for prediction of long-term compressive strength of silica fume concrete", *Adv. Eng. Softw.*, **40**(9), 856-863.

Oztas, A., Pala, M. and Ozbay, E., (2006), "Predicting the compressive strength and slump of high strength concrete using neural network", *Constr. Build. Mater.*, **20**(9), 769-775.

Pal, M. and Deswal, S. (2011), "Support vector regression based shear strength modelling of deep beams", *Comput. Struct.*, **89**(13-14), 1430-1439.

Rakotomamonjy, A. (2005), *SVM and Kernel Methods Matlab Toolbox* http://asi.insa-rouen.fr/enseignants/~arakotom/toolbox/index.html

Ramezanianpour, A.A., Sobhani, M. and Sobhani, J. (2004), "Application of network based neuro-fuzzy system for prediction of the strength of high strength concrete", *AKU J. Sci. Technol*., **15** (59-C), 78-93.

Samui, P. and Kim, D. (2012), "Utilization of support vector machine for prediction of fracture parameters of concrete", *Comput. Concrete*, **9**(3), 215-226.

Sarıdemir, M. (2009), "Predicting the compressive strength of mortars containing metakaolin by artificial neural networks and fuzzy logic", *Adv. Eng. Softw.*, **40**(9), 920-927.

Schölkopf, B. (1997), *Support vector learning*, Munich: R. Oldenbourg.

Shelestynsky, E. (1972), *The workability of no-slump concrete*, University of Western Ontario.

Sobhani, J., Najimi, M., Pourkhorshidi, A.R. and Parhizkar, T. (2010), "Prediction of the compressive strength of no-slump concrete: a comparative study of regression, neural network and ANFIS models", *Constr. Build. Mater*, **24**(5), 709-718.

Tinoco, J., Gomes Correia, A. and Cortez, P. (2011), "Application of data mining techniques in the estimation of the uniaxial compressive strength of jet grouting columns over time", *Constr. Build. Mater*., **25**(3), 1257-1262.

Tong, F., Xu, X.M., Luk, B.L. and Liu, K.P. (2008), "Evaluation of tile-wall bonding integrity based on impact acoustics and support vector machine", *Sensor. Actuat. A-Phys.*, **144**(1), 97-104.

Uysal, M. and Tanyildizi, H. (2011), "Predicting the core compressive strength of self-compacting concrete (SCC) mixtures with mineral additives using artificial neural network", *Constr. Build. Mater*., **25**(11), 4105-4111.

Vapnik, V.N. (2000), *The nature of statistical learning theory*, Springer Verlag.

Vapnik, V.N. (1998), *Statistical learning theory*, Wiley-Interscience.

Wan, C. and Mita, A. (2010), "Early warning of hazard for pipelines by acoustic recognition using principal component analysis and one-class Support vector machines", *Smart Struct. Syst.*, **6**(4), 405-421.

Wang, L., Mu, Z. and Guo, H. (2006), "Application of support vector machine in the prediction of mechanical property of steel materials", *J. Univ. Sci. Tech. Beijing, Min. Met. Mater.*, **13**(6), 512-515.

Yan, K. and Shi, C. (2010), "Prediction of elastic modulus of normal and high strength concrete by support vector machine", *Constr. Build. Mater*, **24**(8), 1479-1485.

Yinfeng, D.m Yingmin, L., Ming, L. and Mingkui, X. (2008), "Nonlinear structural response prediction based on support vector machines", *J. Sound Vib.*, **311**(3-5), 886-897.

Zain, M. and Abd, S. (2009), "Multiple regression model for compressive strength prediction of high performance concrete", *J. Appl. Sci*., **9**(1), 155-160.

CC