Computers and Concrete, *Vol. 21, No. 5 (2018) 505-511*
DOI: https://doi.org/10.12989/cac.2018.21.5.505

Evaluation of concrete compressive strength based on an improved PSO-LSSVM model

Xinhua Xue*

State Key Laboratory of Hydraulics and Mountain River Engineering, College of Water Resource and Hydropower, Sichuan University, No.24 South Section 1, Yihuan Road, Chengdu, Sichuan, 610065, P.R. China

(Received January 17, 2017, Revised February 17, 2018, Accepted February 20, 2018)

Abstract. This paper investigates the potential of a hybrid model which combines the least squares support vector machine (LSSVM) and an improved particle swarm optimization (IMPSO) techniques for prediction of concrete compressive strength. A modified PSO algorithm is employed in determining the optimal values of LSSVM parameters to improve the forecasting accuracy. Experimental data on concrete compressive strength in the literature were used to validate and evaluate the performance of the proposed IMPSO-LSSVM model. Further, predictions from five models (the IMPSO-LSSVM, PSO-LSSVM, genetic algorithm (GA) based LSSVM, back propagation (BP) neural network, and a statistical model) were compared with the experimental data. The results show that the proposed IMPSO-LSSVM model is a feasible and efficient tool for predicting the concrete compressive strength with high accuracy.

Keywords: concrete compressive strength; improved particle swarm optimization; genetic algorithm; statistical model

1. Introduction

It is well known that concrete is one of the most widely used construction materials in the world and the compressive strength is a commonly used criterion in evaluating concrete. Although testing of the compressive strength of concrete specimens is done routinely, it is performed on the 28-day after concrete placement. Nevertheless, it is sometimes desirable and necessary to predict the concrete strength based upon the early strength data, and the reason for that is that it could provide the time for concrete form removal, re-shoring to slab, project scheduling and quality control, etc. In addition, it also provides useful information for designers and engineers including the structural engineer, especially in the application of post-tensioning (Lee 2003, Wisniewski *et al*. 2012). In this context, rapid and reliable prediction for the compressive strength of concrete would be of great significance.

Traditionally, concrete was fabricated from a few well-defined components, such as cement, water, fine and coarse aggregates, etc. However, the rapid development of society and the need of higher performance concretes have led to more complex mixes than the traditional ones. Accordingly, the factors that affect the concrete compressive strength have increased in number and complexity. Although many researchers have proposed various traditional methods for predicting the concrete compressive strength, however, such traditional prediction models have been developed with a fixed equation form based on the limited number of data and parameters. If new data is quite different from original data, then the model should update not only its coefficients but also its equation form. With such limitations, an alternative method is required that provides better estimates of concrete compressive strength.

In recent years there are several attempts to use intelligent computational systems such as artificial neural network (ANN) in civil engineering, including the prediction of concrete compressive strength (e.g., Ni and Wang 2000, Hola and Schabowica 2005, Bilgehan 2011, Khan *et al*. 2013, Ramin *et al*. 2014, Ali 2015, Faruqi *et al*. 2015, Nikoo *et al*. 2015, Chopra *et al*. 2016, Mohammed *et al*. 2016, Gholamreza *et al*. 2016). Although this is successful in many regards, ANN has also several inherent drawbacks such as over fitting, slow convergence, poor generalizing performance (Park and Rilett 1999). Least squares support vector machine (LSSVM) is a supervised machine learning technique that can solve high-dimension and nonlinear pattern recognition problems (Cai *et al*. 2016, Zhang *et al*. 2016). LSSVM is able of performing a faster training process in huge scale problem compared to the standard SVM's. As a modified version of SVM, LSSVM applies equality constraint instead of inequality constrain that has been used in SVM to obtain a linear set of equations, which simplifies the complex calculation and easy to train. However, the disadvantages of the LSSVM method mainly lie in the choice of the kernel and regularization parameters. In this study, three optimization technologies, improved particle swarm optimization (IMPSO), particle swarm optimization (PSO) and genetic algorithm (GA) are employed in selecting the appropriate LSSVM parameters to improve the forecasting accuracy.

*Corresponding author, Associate Professor
E-mail: scuxxh@163.com

http://www.techno-press.org/?journal=cac&subpage=8
ISSN: 1598-8198 (Print), 1598-818X (Online)

2. Materials and method

2.1 Least squares support vector machine (LSSVM)

The LSSVM method, which solves a set of linear equations instead of solving a quadratic programming problem, is an alternative method of SVM described by Suykens *et al*. (2001). It possesses the advantage of not only good generalization performance as SVM, but also simpler structure and shorter optimization time. A brief illustrate of LSSVM is described as follows (Suykens and Vandewalle 1999):

Given training dataset $\{x_i, y_i\}_{i=1}^{N}$, where x_i denotes the input data, y_i denotes the target data, and N is the total number of samples. The minimization of the cost function J of LSSVM can be given by

$$J(w,e)=\frac{1}{2}w^{\mathrm{T}}w+\frac{1}{2}\gamma\sum_{i=1}^{N}e_i^2 \tag{1}$$

where e_i^2 and γ are the quadratic loss term and regularization parameter, respectively.

The solution of the optimization problem of LSSVM can be obtained by introducing the Lagrangian as

$$\begin{aligned}&L(w,b,e,\alpha)\\&=J(w,e)-\sum_{i=1}^{N}\alpha_i\left[w^{\mathrm{T}}\varphi(x_i)+b+e_i-y_i\right]\end{aligned} \tag{2}$$

where α_i is the Lagrange multiplier. The conditions for optimality can be obtained by differentiating with respect to w, b, e_i and α_i, i.e.

$$\frac{\partial L}{\partial w}=0\rightarrow w=\sum_{i=1}^{N}\alpha_i\varphi(x_i) \tag{3a}$$

$$\frac{\partial L}{\partial b}=0\rightarrow w=\sum_{i=1}^{N}\alpha_i=0 \tag{3b}$$

$$\frac{\partial L}{\partial e_i}=0\rightarrow \alpha_i=\gamma e_i,\quad i=1,\cdots,N \tag{3c}$$

$$\frac{\partial L}{\partial \alpha_i}=0\rightarrow w^{\mathrm{T}}\varphi(x_i)+b+e_i-y_i\quad i=1,\cdots,N \tag{3d}$$

By elimination of w and e_i, the solution can be written as follows

$$\begin{bmatrix}0 & \vec{1}^{\mathrm{T}}\\ \vec{1} & \Omega+\gamma^{-1}I\end{bmatrix}\begin{bmatrix}b\\ \alpha\end{bmatrix}=\begin{bmatrix}0\\ y\end{bmatrix} \tag{4}$$

where y=[y_1;…;y_N], $\vec{1}=[1;\ldots;1]$, α=[α_1; …; α_N]. By applying Mercer's theorem (1909), the resulting LSSVM for function estimation can be obtained as follows

$$y(x)=\sum_{i=1}^{N}\alpha_i K(x,x_i)+b \tag{5}$$

where $K(x,x_i)$ is the kernel function.

There are several kernel functions, such as the linear kernel functions, polynomial kernel functions, radial basis function (RBF), sigmoid kernel functions that are used in LSSVM. Dibike *et al*. (2001) demonstrated that the RBF outperformed other kernel functions after using different kernels in SVM for rainfall runoff modeling. Therefore, the RBF is adopted in this study and expressed as

$$K(x,x_i)=\exp\left(-\frac{\|x-x_i\|^2}{2\sigma^2}\right) \tag{6}$$

where σ is the width of the radial basis function.

It can be seen that there are two parameters, namely the regularization parameter γ and kernel parameter σ, to be optimized in model building of LSSVM. The regularization parameter γ determines the training error and complexity of the LSSVM. Generally, the higher the values of γ, the lower the training error. However, if the value of γ is set too high, the model will be overly complex, and it will cause the over-training problem. The kernel parameter σ affects a nonlinear mapping relationship from input space to a high-dimension feature space. The value of σ defines the shape of a bell-type function in the feature space which determines the local properties of the RBF function. Therefore, the parameters of LSSVM (γ and σ) need to be optimized to improve the prediction performance.

2.2 LSSVM optimized by improved PSO (IMPSO-LSSVM)

PSO algorithm is a population-based heuristic search technique developed by Kennedy and Eberhart (1995), inspired by the social behavior of bird flocks and fish schools. For more details of PSO, readers are referred to Kennedy and Eberhart (1995).

To improve the performance of PSO, an improved PSO algorithm (IMPSO) is adopted herein by modifying the inertia weight factor κ and the velocity of each particle as shown in the following equations:

$$\begin{aligned}&v_i^{t+1}\\&=\eta\left[\kappa v_i^t+c_1r_1\left(P_i-x_i^t\right)+c_2r_2\left(P_g-x_i^t\right)\right]\end{aligned} \tag{7}$$

$$\eta=\frac{2}{\left|2-\varphi-\sqrt{\varphi^2-4\varphi}\right|},\quad \varphi=c_1+c_2>4 \tag{8}$$

$$\kappa=\kappa_{\max}-n_i\times\frac{(\kappa_{\max}-\kappa_{\min})}{n_{\max}} \tag{9}$$

where c_1 and c_2 are the acceleration coefficients, usually c_1=c_2=2; r_1 and r_2 are two independent random numbers uniformly distributed in the range [0, 1]; P_i is the best previous position of the particle, while P_g is the best position among all the particles in the swarm. κ is the inertia weight factor. $\kappa_{\max}$ and $\kappa_{\min}$ are the maximum and minimum inertia weights, respectively, n_i is the current generation number, $n_{\max}$ is the maximum number of iterations. According to Sakthivel *et al*. (2010), the maximum and minimum weights are set as follows: $\kappa_{\max}$=0.9, $\kappa_{\min}$=0.1. η is the constriction factor.

The main steps of the proposed IMPSO-LSSVM approach are described as follows:

Step 1: Take the parameters (γ, σ) as swarms and

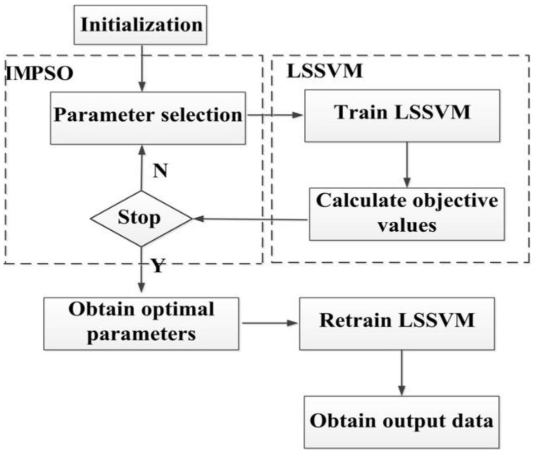


Fig. 1 Flowchart of the IMPSO-LSSVM algorithm

initialize a population of particles with random positions and velocities.

Step 2: Train the LSSVM model and evaluate the objective values of all particles. In this paper, the root mean squared error (*RMSE*) is used as the objective function to evaluate the objective values of the particles, as shown in the following equation

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y-\hat{y}\right)^2} \qquad (10)$$

where y and $\hat{y}$ represent the measured and simulated values, respectively. N is the number of samples. Clearly, the smaller the *RMSE* value, the better the prediction accuracy and vice versa.

Step 3: Update the velocity and position of each particle according to Eqs. (7)-(9) and (11), respectively.

$$x_i^{t+1} = x_i^t + v_i^{t+1} \qquad (11)$$

Step 4: If the maximum of the iteration is achieved or the optimum solution is acquired, then the algorithm is stopped; otherwise go back to Step 2.

Step 5: Obtain the LSSVM model at the optimal parameters and get the output data.

The implementation of the proposed IMPSO-LSSVM model was carried out using MATLAB R2012b program in this study. The main flowchart of IMPSO-LSSVM is shown in Fig. 1.

2.3 LSSVM optimized by GA (GA-LSSVM)

Genetic algorithm (GA) is a search algorithm based upon the mechanics of natural selection, derived from the theory of natural evolution. GA simulates mechanisms of population genetics and natural rules of survival in pursuit of the ideas of adaptation. In this section, the proposed GA-based LSSVM parameter optimization approach is described as follows.

2.3.1 Chromosome design

In this study, the RBF kernel function is adopted due to its promising performances and thus only two parameters, γ and σ, need to be optimized by using the proposed GA-based method. Therefore, the chromosome comprises two parameters, γ and σ. Fig. 2 shows the chromosome model formed by two binary blocks: (1) the first block, which includes γ_i with $1 \le i \le n_\gamma$, is the γ parameter binary representation in n_γ bits; (2) the second block, which includes σ_j with $1 \le j \le n_\sigma$, is the σ parameter binary representation in n_σ bits.

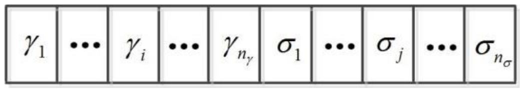


Fig. 2 The chromosome comprises two parameters, γ and σ

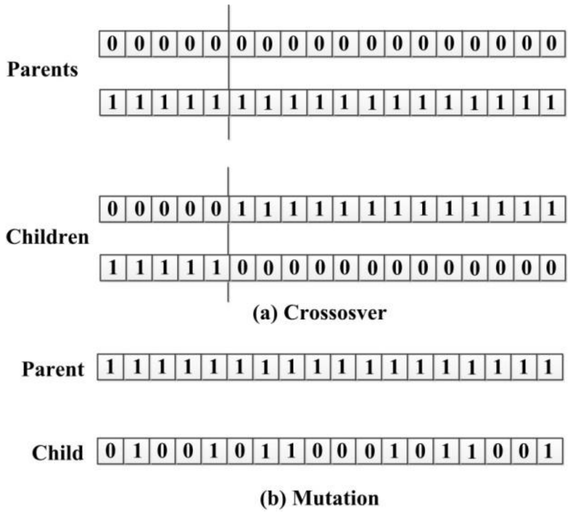


Fig. 3 Genetic crossover and mutation operation

2.3.2 Genetic operators

Fig.3 illustrates the genetic operators of crossover and mutation. Crossover is the critical genetic operator that allows new solution regions in the search space to be explored, and it is performed by selecting a random gene along the length of the chromosomes and swapping all genes after that point. In mutation, the genes may occasionally be altered, i.e. binary code genes can change from 1 to 0 or vice versa.

2.3.3 GA-LSSVM approach

The main steps of the proposed GA-LSSVM approach are described as follows:

Step 1: Initialization. Generate initial population which individually is comprised of γ and σ. In this study, the searching ranges of γ and σ are [0, 1000] and [0,100], respectively.

Step 2: Fitness definition. Fitness function is an objective function that estimates the quality of each chromosome. The root mean squared error (*RMSE*) is used herein as the fitness function as shown in Eq. (10).

Step 3: Genetic manipulations, including selection, crossover and mutation, are performed to attain a new population.

Step 4: Termination criteria. When the termination criteria are satisfied, the process ends; otherwise, go back to Step 3.

Step 5: Obtain the LSSVM model at the optimal

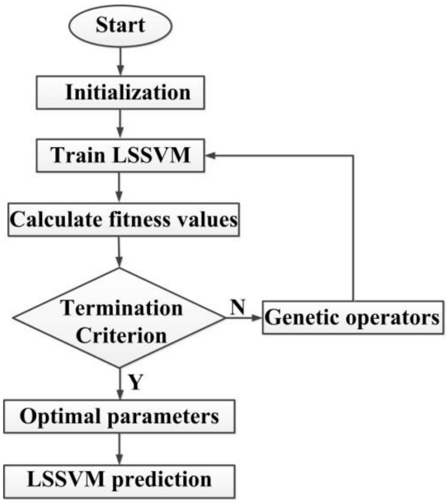


Fig. 4 Flowchart of the GA-LSSVM algorithm

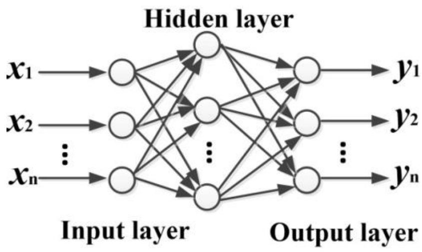


Fig. 5 BP neural network architecture

parameters and get the output data.

The implementation of the proposed GA-LSSVM approach was carried out using MATLAB R2012b program in this study. The main flowchart of GA-LSSVM is shown in Fig. 4.

2.4 BP neural network (BP)

In artificial neural networks, BP neural network is one of the powerful tools for prediction of nonlinearities. It mainly consists of three layers: input layer, hidden layer, and output layer. The neighboring layers are fully interconnected by weights, as shown in Fig. 5. That is, each neuron in the input layer is connected to all of the neurons in the first hidden layer. Each of the neurons in the first hidden layer is connected to each output neuron. Further, each of the neurons in the input layer is connected to each output neuron.

2.5 Statistical analysis

In the conventional material modeling process, multiple regression method is often used to determine the relationships between different variables. In this study, concrete compressive strength is considered to be the outcome of seven parameters i.e., cement (*C*), blast furnace slag (*BFS*), fly ash (*F*), water (*W*), superplasticizer (*S*), coarse aggregate (*CA*), and fine aggregate (*FA*). To generate multivariate relation based on the main data (425 datasets), the MS Excel was used and the obtained regression equation is

$$f_c = -95.66 + 0.1688 \cdot C + 0.1441 \cdot BFS + 0.1059 \cdot F - 0.065 \cdot W + 0.1107 \cdot S + 0.0403 \cdot CA + 0.0537 \cdot FA \tag{12}$$

2.6 Performance evaluation

To validate and compare the acquired results from the IMPSO-LSSVM, PSO-LSSVM, GA-LSSVM, BP, and that of the statistical method (Eq. (12)), three statistical indexes are used, that is, root mean squared error (*RMSE*), mean absolute error (*MAE*) and coefficient of determination (R^2).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{n}} \tag{13}$$

$$MAE = \frac{\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|}{n} \tag{14}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \tilde{y}_i\right)^2} \tag{15}$$

where y_i, $\hat{y}_i$ and $\tilde{y}_i$ are the predicted, actual and averaged actual output of the network, respectively, and *n* is the total number of patterns.

3. Case study

3.1 Datasets

In this study, the following seven factors including the cement, blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, and fine aggregate were taken into account as the input parameters of the models of IMPSO-LSSVM, PSO-LSSVM, GA-LSSVM, BP, and that of the statistical method (Eq. (12)). Most often the 28-day concrete compressive strength (CCS) is the most used mechanical property in the design of concrete structures. Therefore, the 28-day CCS is the output of IMPSO-LSSVM, PSO-LSSVM, GA-LSSVM, BP, and that of the statistical method (Eq. (12)). The database used in this study was selected from Yeh (1998) and the total number of datasets is 425. It is noted that those experimental results (Yeh 1998) were separated in terms of 28-day concrete compressive strength and used in this study. The first 340 of the total data were used to train the proposed LSSVM model, whereas the remaining 85 of the data were used to verify the accuracy and the effectiveness of the trained LSSVM model. The detailed training and testing datasets are summarized in Table 1.

3.2 Results and discussion

The important parameters of IMPSO used in this study are given as follows: the acceleration coefficients c_1=c_2=2.0, N_p=30 (warm size), the maximum iterations

Table 1 Statistics analysis of dataset (data from Yeh 1998)

Variable	Maximum	Minimum	Average
C (kg/m^3)	540	102	265.44
BFS (kg/m^3)	359.4	0	86.29
F (kg/m^3)	200.1	0	62.79
W (kg/m^3)	247	121.8	183.06
S (kg/m^3)	32.2	0	6.99
CA (kg/m^3)	1145	801	956.06
FA (kg/m^3)	992.6	594	764.38
CCS (MPa)	81.75	8.54	36.75

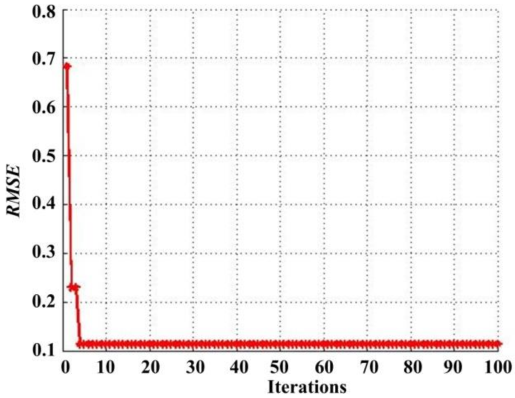


Fig. 6 Convergence procedure of IMPSO

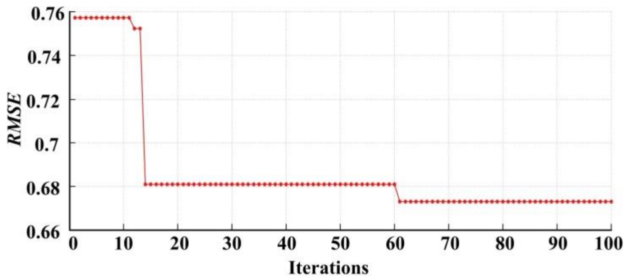


Fig. 7 Convergence procedure of PSO

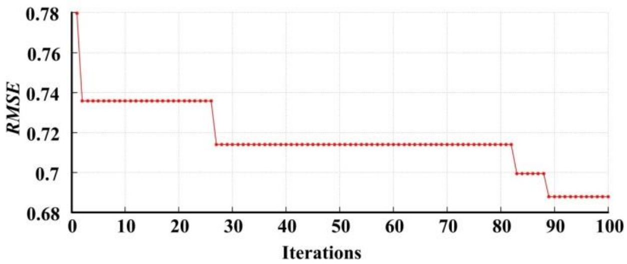


Fig. 8 Convergence procedure of GA

n_{max}=100. The parameters of the GA are set as a generation number of 100, initial population size of 20, crossover probability of 0.8, and mutation probability of 0.06. After the procedures of IMPSO-LSSVM, PSO-LSSVM, GA-LSSVM, respectively, the optimal parameters of LSSVM were selected, ie.

σ=0.203, γ=451.44 (IMPSO-LSSVM); σ=0.294, γ=282.97 (PSO-LSSVM);

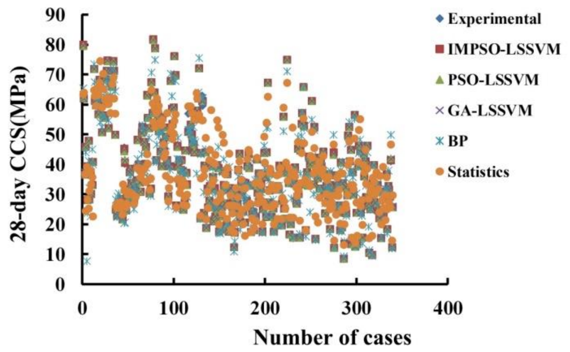
and σ=0.4918, γ=131.42 (GA-LSSVM). Figs. 6-8 show the convergence curves of IMPSO, PSO and GA, respectively.

Table 2 Performance comparison among different models

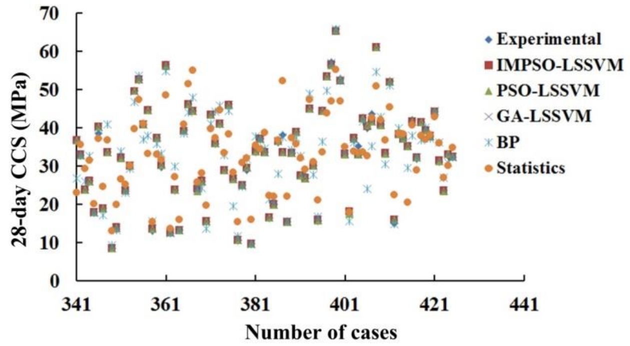
Models	*MAE*		R^2		*RMSE*	
	Training	Testing	Training	Testing	Training	Testing
IMPSO-LSSVM	0.0881	0.2526	0.9993	0.9986	0.1132	0.1459
PSO-LSSVM	0.1182	0.2865	0.9991	0.9985	0.6731	0.6892
GA-LSSVM	0.1492	0.2968	0.9890	0.9978	0.6879	0.7357
BP	3.4499	2.6992	0.9341	0.9505	5.4396	3.9084
Eq.(12)	5.4433	4.4901	0.8741	0.8989	7.3333	5.6335

From Figs. 6-8, it can be seen that the performance of the proposed IMPSO-LSSVM is superior to those of PSO-LSSVM and GA-LSSVM. For example, the *RMSE* of IMPSO-LSSVM reaches the minimum (0.1132) at the 4th iteration; however, the *RMSE* of PSO-LSSVM and GA-LSSVM reach the minimum, 0.6731 and 0.6879, at the 61th and 89th iteration, respectively. The results show that the proposed IMPSO-LSSVM model has a good performance. In addition, to evaluate the performance of the proposed IMPSO-LSSVM method, a comparison between the experimental results and the predictions by the IMPSO-LSSVM, PSO-LSSVM, GA-LSSVM, BP, and that of the statistical method (Eq. (12)) are made and shown in Fig.9 and Table 2, respectively. It should be noted that it is very important to select the number of hidden layers and the number of neurons in various layers before using the BP neural network. The number of neurons in input and output layers is usually dictated by the nature of the problem. In this study, there are 7 parameters including the cement, blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, and fine aggregate were taken into account as the input parameters, therefore, the number of neurons in input layers is 7. As mentioned, the main objective of this paper is to predict the 28-day CCS, so the number of neurons in output layer is 1 and one hidden layer BP neural network is adopted herein. For the number of neurons in hidden layer, the main strategy is to use as few hidden layer neurons as possible, because each unit adds to the loads on the CPU during simulations. If the network fails to converge to a solution, it means that more hidden neurons are required. If it does converge we might try for fewer hidden neurons. Based on this idea the number of hidden neurons were determined by trial and found suitable network with 10 neurons in hidden layer. Thus, the structure of BP neural network is designed as 7-10-1. Many kinds of transfer functions have been proposed in literature and one of the most popular hidden layer transfer functions is the tangent sigmoid function, therefore, the tangent sigmoid transfer function is employed in the hidden layer herein. Because the pureline transfer function is sufficient for BP neural network to approximate almost any complex function, therefore, it is employed in the output layer in this study.

As shown in Table 2, the *RMSE* of IMPSO-LSSVM, PSO-LSSVM, GA-LSSVM, BP, and that of the statistical method (Eq. (12)) for training and testing are 0.1132 and 0.1459, 0.6731 and 0.6892, 0.6879 and 0.7357, 5.4396 and 3.9084, 7.3333 and 5.6335, respectively. The *MAE* of



(a) Training samples



(b) Testing samples

Fig. 9 Comparison between the forecasted and experimental results

IMPSO-LSSVM, PSO-LSSVM, GA-LSSVM, BP, and that of the statistical method (Eq. (12)) for training and testing are 0.0881 and 0.2526, 0.1182 and 0.2865, 0.1492 and 0.2968, 3.4499 and 2.6992, 5.4433 and 4.4901, respectively. Whereas the R^2 of IMPSO-LSSVM, PSO-LSSVM, GA-LSSVM, BP, and that of the statistical method (Eq. (12)) for training and testing are 0.9993 and 0.9986, 0.9991 and 0.9985, 0.9890 and 0.9978, 0.9341 and 0.9505, 0.8741 and 0.8989, respectively. Clearly, the smaller the *RMSE* and *MAE* values and the higher the R^2 values, the better the prediction accuracy and vice versa. These results indicate that the proposed IMPSO-LSSVM is a valid tool to predict the concrete compressive strength.

4. Conclusions

Prediction of concrete compressive strength is an engineering problem that involves several parameters. To address these problems, a hybrid model which combines the LSSVM with a modified PSO algorithm is developed for the prediction of 28-day concrete compressive strength in this study. Further, predictions from five models (the IMPSO-LSSVM, PSO-LSSVM, GA-LSSVM, BP and ta statistical model (Eq. (12)) were compared with the experimental data. The results confirmed that the developed IMPSO-LSSVM model can provide a precise evaluation of concrete compressive strength. Therefore, the proposed IMPSO-LSSVM model may be one of the most competent artificial intelligence subsystems to evaluate the concrete compressive strength.

References

Ali, D. (2015), “Prediction of hybrid fibre-added concrete strength using artificial neural networks”, *Comput. Conrete*, **15**(4), 503-514.

Bilgehan, M. (2011), “A comparative study for the concrete compressive strength estimation using neural network and neuro-fuzzy modeling approaches”, *Nondestr. Test. Eval.*, **26**(1), 35-55.

Cai, Z.L., Xu, W.Y., Meng, Y.D., Shi, C. and Wang, R.B. (2016), “Prediction of landslide displacement based GA-LSSVM with multiple factors”, *B. Eng.Geol. Environ.*, **75**(2), 637-646.

Chopra, P., Sharma, R.K. and Kumar, M. (2016), “Prediction of compressive strength of concrete using artificial neural network and genetic programming”, *Adv. Mater. Sci. Eng.*, 1-10.

Dibike, Y., Velickov, S., Solomatine, D. and Abbott, M. (2001), “Model induction with support vector machines: introduction and applications”, *J. Comput. Civil Eng.*, **15**(3), 208-216.

Faruqi, M.A., Agarwala, R., Sai, J. and Francisco, A. (2015), “Application of artificial intelligence to predict compressive strength of concrete from mix design parameters: a structural engineering application”, *J. Civil Eng. Res.*, **5**(6), 158-161.

Gholamreza, A., Ehsan, J. and Zahra, K. (2016), “Predicting of compressive strength of recycled aggregate concrete by genetic programming”, *Comput. Conrete*, **18**(2), 155-163.

Hola, J. and Schabowicka, K. (2005), “Application of artificial neural network to determine concrete compressive strength based on non-destructive tests”, *J. Civil Eng. Manage.*, **11**(1), 23-32.

Kennedy, J. and Eberhart, R.C. (1995), “Particle swarm optimization”, *Proc. IEEE Int. Conf. on Neural Networks*, **4**, IEEE Service Center, Piscataway, Perth, Australia.

Khan, S.U., Ayub, T. and Rafeeqi, S.F.A. (2013), “Prediction of compressive strength of plain concrete confined with ferrocement using artificial neural network and comparison with existing mathematical models”, *Am. J. Civil Eng. Arch.*, **1**(1), 7-14.

Lee, S.C. (2003), “Prediction of concrete strength using artificial neural networks”, *Eng. Struct.*, **25**, 849-857.

Mercer, J. (1909), “Functions of positive and negative type and their connection with the theory of integral equations”, *Philos. T. R. Soc.*, **209**, 415-446.

Mohammed, S., Steffen, G., Abdulkadir, C. and Joost, W. (2016), “Modelling fresh properties of self-compacting concrete using neural network technique”, *Comput. Conrete*, **18**(4), 903-921.

Ni, H.G. and Wang, J.Z. (2000), “Prediction of compressive strength of concrete by neural networks”, *Cement Concrete Res.*, **30**(8), 1245-1250.

Nikoo, M., Moghadam, F.T. and Sadowski, L. (2015), “Prediction of concrete compressive strength by evolutionary artificial neural networks”, *Adv. Mater. Sci. Eng.*, 1-8.

Park, D. and Rilett, L.R. (1999), “Forecasting freeway link ravel times with a multi-layer feed forward neural network”, *Comput Aid. Civil Infr.*, **14**, 358-367.

Ramin, T., Hamid, R.S. and Mohsen, S. (2014), “The use of artificial neural networks in predicting ASR of concrete containing nano-silica”, *Comput. Conrete*, **13**(6), 739-748.

Sakthivel, V.P., Bhuvaneswari, R. and Subramanian, S. (2010), “An improved particle swarm optimization for induction motor parameter determination”, *Int. J. Comput. Appl.*, **1**(2), 62-67.

Suykens, J.A.K. and Vandewalle, J. (1999), “Least squares support vector machine classifiers”, *Neur. Proc. Lett.*, **9**, 293-300.

Suykens, J.A.K., Vandewalle, J. and De, M. B. (2001), “Optimal control by least squares support vector machines”, *Neur. Network.*, **14**, 23-35.

Wisniewski, D.F., Cruz, P.J.S., Henriques, A. Abel R. and Simoes, R.A.D. (2012), “Probabilistic models for mechanical

properties of concrete, reinforcing steel and pre-stressing steel", *Struct. Inf. Eng.*, **8**(2), 111-123.
Yeh, I.C. (1998), "Modeling of strength of high performance concrete using artificial neural networks", *Cement Concrete Res.*, **28**(12), 1797-1808.
Zhang, C.S., Ji, J., Gui, Y.l., Kodikara, J., Yang, S.Q. and He, L. (2016), "Evaluation of soil-concrete interface shear strength based on LS-SVM", *Geomech. Eng.*, **11**(3), 361-372.

CC