

# Metaheuristic-Based Machine Learning System for Prediction of Compressive Strength based on Concrete Mixture Properties and Early-Age Strength Test Results

Prayogo, D.<sup>1</sup>

---

**Abstract:** Estimating the accurate concrete strength has become a critical issue in civil engineering. The 28-day concrete cylinder test results depict the concrete's characteristic strength which was prepared and cast as part of the concrete work on the project. Waiting 28 days is important to guarantee the quality control of the procedure, even though it is a slow process. This research develops an advanced machine learning method to forecast the concrete compressive strength using the concrete mix proportion and early-age strength test results. Thirty-eight historical cases in total were used to create the intelligence prediction method. The results obtained indicate the effectiveness of the advanced hybrid machine learning strategy in forecasting the strength of the concrete with a comparatively high degree of accuracy calculated using 4 error indicators. As a result, the suggested study can provide a great advantage for construction project managers in decision-making procedures that depend on early strength results of the tests.

**Keywords:** Concrete compressive strength; early-age; machine learning; metaheuristic; prediction.

---

## Introduction

In the construction field, the strength of concrete is a significant criterion when selecting a type of concrete to be used for a specific purpose. Construction concrete will increase in strength for an extended time span once it has been poured. The nominal strength of concrete is defined by the compressive strength of the sample at 28 days. If there was a defective mix preparation or mix design on location, the test results might show the required strength was not reached, resulting in a mandatory repeating of the entire procedure, which could be a slow and expensive process.

Any failures would require waiting another 28 days for the new test results, therefore, there was a great need for the ability to determine the final strength of concrete at an early-age. Consequently, a suitable, fast procedure for predicting the strength of concrete would definitely be a major advancement in the construction industry [1]. The capability to forecast the early compressive concrete strength would enable contractors to quickly recognize the concrete's possible weakness, thereby planning to deal with a destruction procedure or to keep going with the construction.

Additionally, to the advantage of both the user and the manufacturer, reliably and quickly forecasting the outcomes of a 28-day test would be advantageous to all stakeholders involved rather than waiting the entire, traditional, 28 days.

Concrete behavior, including forecasting the strength of concrete has been an area of interest for researchers and is gaining even more interests today. There have been several recent scientific studies examining the behavior of concrete and the potential for improving the estimations of characteristic strength. The research has shown that numerous tests have focused on how concrete strength is impacted by the mix, however, just a handful of studies has concentrated on the connection between early testing and the total 28-day compressive strength test. Furthermore, the majority of the scientific studies has certain constraints; such as, a lack of advanced technique for calculating accuracy, no validating methods involved, or working with only a traditional approach.

Machine learning techniques have been shown to exceed traditional techniques due to their excellent learning capabilities [2-6]. The traditional methods, like linear regression and decision tree, are not good enough to develop a suitable model with regards to accuracy and computation time. The least squares support vector regression (LSSVR) technique has evolved into an excellent machine learning approach and has been extensively applied in numerous fields because of its advantages. For example, Cheng and

---

<sup>1</sup> Faculty of Civil Engineering and Planning, Civil Engineering Department, Petra Christian University, Jl. Siwalankerto 121-131, Surabaya 60236, INDONESIA

\* Corresponding author: prayogo@petra.ac.id

**Note:** Discussion is expected before June, 1<sup>st</sup> 2018, and will be published in the "Civil Engineering Dimension", volume 20, number 2, September 2018.

Received 18 February 2018; revised 15 March 2018; accepted 19 March 2018.

Prayogo applied an enhanced LSSVR technique to predict the permanent deformation behavior of asphalt mixtures [6]. Hoang et al. developed an optimized LSSVR model for groutability estimation of grouting process [7]. Notably, the hyperparameters of the LSSVR technique must be fine-tuned to enhance prediction accuracy.

This study develops a novel hybrid machine learning technique called symbiotic organisms search–least squares support vector regression (SOS–LSSVR). The main objective of this research is to develop an advanced, accurate method to improve early-age estimates of concrete strength. The SOS–LSSVR approach combines an accurate prediction method, least squares support vector regression (LSSVR), as well as a powerful and new metaheuristic, symbiotic organisms search (SOS). The proposed model will be examined together with various prediction techniques to establish the prediction model of concrete strength with the early-age strength test results.

## Least Squares Support Vector Regression

### The Basic Model

LS-SVR was first introduced by Suykens and Vandewalle [8] as a modification of the conventional support vector regression (SVR). The following model of interest underlies the functional relationship between one or more independent variables by using a regression function as states as follows:

$$y(x) = w^T \phi(x) + b \quad (1)$$

where  $x \in R^n$ ,  $y \in R$ , and  $\phi(x): R^n \rightarrow R^{nh}$  is the mapping to the high dimensional feature space. In LS-SVR, given a training data set  $\{x_k, y_k\}_{k=1}^N$ , the optimization problem is formulated as follows:

$$\text{Minimize } J_p(w, e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=1}^N e_k^2 \quad (2)$$

$$\text{Subjected to } y_k = w^T \phi(x_k) + b + e_k, \quad k = 1, \dots, N \quad (3)$$

where  $e_k \in R$  are error variables;  $\gamma > 0$  denotes a regularization constant.

In the previous optimization problem, a regularization term and a sum of squared fitting errors make for the objective function. The Lagrangian is given by:

$$L(w, b, e; \alpha) = J_p(w, e) - \sum_{k=1}^N \alpha_k \{w^T \phi(x_k) + b + e_k - y_k\} \quad (4)$$

where  $\alpha_k$  are Lagrange multipliers. The conditions for optimality are given by:

$$\begin{cases} \frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{k=1}^N \alpha_k \phi(x_k) \\ \frac{\partial L}{\partial b} = 0 \rightarrow \sum_{k=1}^N \alpha_k = 0 \\ \frac{\partial L}{\partial e_k} = 0 \rightarrow \alpha_k = \gamma e_k, \quad k = 1, \dots, N \\ \frac{\partial L}{\partial \alpha_k} = 0 \rightarrow w^T \phi(x_k) + b + e_k - y_k = 0, \quad k = 1, \dots, N \end{cases} \quad (5)$$

The resulting LS-SVR model for function estimation is expressed as:

$$y(x) = \sum_{k=1}^N \alpha_k K(x_k, x_l) + b \quad (6)$$

where  $\alpha_k$  and  $b$  are the solution to the linear system.

The kernel function that is often utilized is a radial based function (RBF) kernel; a description is given as:

$$K(x_k, x_l) = \exp\left(-\frac{\|x_k - x_l\|^2}{2\sigma^2}\right) \quad (7)$$

where  $\sigma$  is the kernel function parameter.

### Model Selection in the Training Phase

Initially, the prediction model of the LSSVR is trained on a training set, which is a set of samples adapted to fit the parameters of the model. The test data set is employed to give an impartial assessment of the trained model. It is a popular notion that the generalization performance of an LSSVR trained model relies on fine-tuning of the hyper-parameters ( $\gamma$  and  $\sigma$ ) referred to as “model selection”. In order to guarantee optimal performance of the prediction model, the two tuning hyper-parameters have to be set correctly.

The LSSVR parameter selection is commonly referred to as the model selection problem and could be put together as an optimization problem. One common approach to help with the model selection is to use an extensive grid search over the parameter domain. Clearly, the parameter domain for grid search needs to involve a large search space to cover the global optimum. This forces the grid search to have expensive computational cost, particularly on the large-scale training set. Consequently, a more advanced system is required to identify the best combination of LSSVR parameters.

Research has shown that there is an increase in prediction accuracy when metaheuristic algorithm is used as the optimizer for model selection and has improved prediction performance in various engineering problems [2,6,9,10]. Many studies have

shown that the SOS algorithm is better when compared to other metaheuristic algorithms in finding the optimal solutions to problems involving complex and nonlinear optimization [2,11]. Additionally, a past study has successfully utilized the SOS algorithm as a self-automatic tuning framework in optimizing hyper-parameters of machine learning techniques and has produced a better accuracy when compared to a variety of hybrid techniques [12]. Therefore, this study uses SOS to enhance the prediction accuracy of LSSVR.

### The Proposed Metaheuristic-based Machine Learning System

The search for optimality proves to be a demanding task in many optimization applications. Optimum solutions to problems have taken place in nature via evolution, where the process of natural selection has removed most inferior solutions. Past studies revealed that metaheuristic algorithms inspired by nature like SOS are effective in finding solutions to complicated optimization problems. Therefore, this study utilizes SOS to fine-tune the LSSVR hyper-parameters, as well as to guarantee excellent prediction accuracy.

### Symbiotic Organisms Search

The SOS algorithm was proposed by Cheng and Prayogo and has become one of the most popular metaheuristic algorithms in use today [2,13]. It is derived from natural organisms that usually have symbiosis–dependency-based relationships. Similar to other typical metaheuristic algorithms, the SOS algorithm leads to the optimization process of the candidate solutions through special search operators. The SOS algorithm has been used for solving optimization problems in different areas of research from the time it was developed in 2014 [11,14-20].

To begin with, an ecosystem matrix (population) is randomly generated by the SOS. The ecosystem size denotes the number of organisms that can be put into the ecosystem. Every matrix row corresponds to virtual organisms which represents the different candidate solutions. Each virtual organism has to be related to the objective value of the current problem. The search begins following the initial generation of the random ecosystem. The searching process contains three phases, during which the organisms take advantage of the interaction (mutualism, commensalism, and parasitism). The objective value of the updated virtual organism has to be improved so that the pre-interaction virtual organisms can be replaced. In every iteration, once all phases are done, the best virtual organisms can be updated. To conclude, the phase cycle continuously until the termination criterion has been satisfied.

SOS uses the three rules of symbioses: (1) mutualism symbiosis, which refers to the reciprocated advantages associated with two living organisms; (2) commensalism symbiosis, which one organism takes all the benefits from the other, while this other organism not substantially affected by this interaction; (3) parasitism symbiosis, which the advantage that an organism obtains from the other are a disadvantage to this other organism. The mathematical model adaptation for these symbioses is explained in the following subsections.

### Mutualism Phase

The relationship in the mutualism phase is characterized by the benefits of both sides. One such case is the relationship between bees and flowers. The following is the mathematical formulation of this phase:

$$newSol_i = currentSol_i + rand(0,1) * (bestSol - mutualSol_{ij} * BF_1) \quad (8)$$

$$newSol_j = currentSol_j + rand(0,1) * (bestSol - mutualSol_{ij} * BF_2) \quad (9)$$

Here  $currentSol_i$  and  $currentSol_j$  are two current virtual organisms involved in mutualism;  $bestSol$  is the current best virtual organism;  $rand(0,1)$  represents the uniform random value between 0 to 1;  $mutualSol_{ij}$  models the mutualism interaction of current virtual organisms;  $newSol_i$  and  $newSol_j$  are the updated virtual organisms following the interaction;  $BF_1$  and  $BF_2$  represent two random values of either 0 or 1 illustrating the level of benefit each virtual organism has. The following formulation is used to calculate  $mutualSol_{ij}$ .

$$mutualSol_{ij} = (currentSol_i + currentSol_j) / 2 \quad (10)$$

### Commensalism Phase

In the commensalism phase, one virtual organism establishes a relationship in which it is the sole beneficiary, such as, a relationship between sharks and remora fish. The following is a mathematical formulation for this phase:

$$newSol_i = currentSol_i + rand(-1,1) * (bestSol - currentSol_j) \quad (11)$$

where  $rand(-1,1)$  represents the uniform random value between -1 to 1.

### Parasitism Phase

The relationship in the parasitism phase is denoted by being harmful to one side and beneficial for the other. To illustrate, the plasmodium parasite uses the anopheles mosquito to transfer itself from one human to another. The harmed side of this relation-

ship will probably perish, whereas the beneficiary will become fitter. The following is a mathematical formulation for this phase.

$$parasiteSol_i = F * currentSol_i + (1 - F) * [rand(0,1) * (ub - lb)] \quad (12)$$

Here  $parasiteSol_i$  is the artificial parasite engaged with  $currentSol_i$ , and it threatens the existence of  $currentSol_i$ ;  $F$  and  $(1-F)$  are the binary random matrix and its inverse, respectively;  $ub$  and  $lb$  are the upper and lower bound of the searching area.

### Cross-validation for Partitioning the Training Data Set

For machine learning technique, establishing a good prediction model is critical and requires a proper training and test process. To start off with, a data set is utilized to build a prediction model in the training phase. The trained model is further used to validate a new and unseen data set. Nevertheless, if the entire data set is used for training, there is a possibility of an ‘overfitting’ phenomenon to actually occur. In this circumstance, the trained model fits the data set very well, but, it performs poorly for a new and unseen data set. To prevent the overfitting problem, dividing the training set into two subsets has become a common practice nowadays. The large portion of the training set is categorized as the ‘training subset’ and the small portion of the training set is labelled as the ‘validation subset’. The larger subset is used for training the model while the smaller subset is employed to validate the model built.

To eliminate the randomness in partitioning the training set, a  $k$ -fold cross-validation technique is proposed in this study [21]. During this procedure,  $k$ -fold cross-validation creates non-overlapping  $k$  subsets from training set. Since  $k$  is a variable parameter, any adequate number will work. This study sets the value of  $k$  to be 5. Hence, the data is split up into five random equal size groups, with 4 subsets are employed as training subsets and one as a validation subset. A total of  $(k-1)$  subsets are employed for training the model, while the remaining  $k$ -th subset is employed for the validating the training process. Since the procedure is based on cross-validation, it is repeated  $k$  times to ensure that every subset is used at least once as the validation subset.

### Integrating the Metaheuristic in Model Selection for LSSVR

In this step, a hybrid system is proposed so called the symbiotic organisms search-least squares support vector regression (SOS–LSSVR) that combines the

two different techniques of SOS and LSSVR. As mentioned earlier, the LSSVR acts as a predictor to build the accurate input-output relationship of the data set; and the SOS works to optimize the LSSVR parameters, the  $\gamma$  and  $\sigma$  parameters. The whole procedure of SOS–LSSVR is shown in Figure 1.

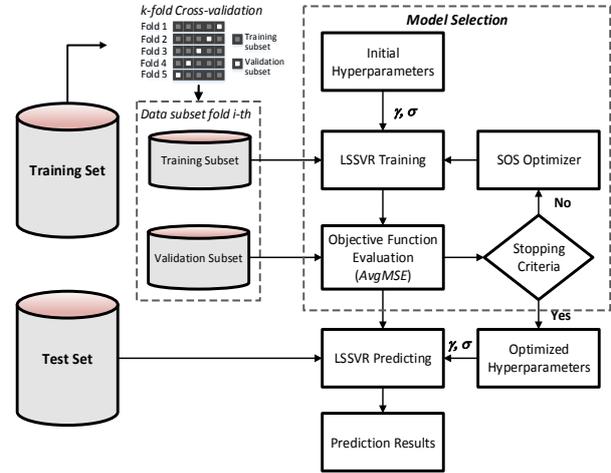


Figure 1. SOS–LSSVR Framework

The SOS was allowed to identify the optimal LSSVR parameters, and accordingly the predictive model sets were constructed in the training process. As mentioned previously, the training data set consists of two parts, the ‘training subset’ and the ‘validation subset’. The objective of this division is to avoid the chance of overfitting throughout the training procedure. The study chose the  $k$ -fold cross-validation process to avoid bias throughout the sample partitioning procedure. A supervised learning procedure of the LSSVR is utilized to train the model on the training subset to find the best fit of the  $\gamma$  and  $\sigma$  of the LSSVR hyper-parameter of the model.

Afterward, the fitted model is used to predict the target output from the validation subset. It is worth noting that the validation subset offers an unbiased assessment of a model fit on the training subset, and it also tunes the hyper-parameters of the model. Mean squared error ( $AvgMSE$ ) is employed to observe the average prediction error from the validation subset.

$$AvgMSE = \frac{1}{k} \frac{1}{n} \sum_{i=1}^k \sum_{j=1}^n (y_j - y_j')^2 \quad (12)$$

where  $y_j$  and  $y_j'$  are the  $j$ -th actual and predicted values, respectively;  $n$  represents the total number of validation samples; and  $k$  represents the total number of training simulations through  $k$ -fold cross-validation.

The optimization process of the parameter selection of LSSVR is simulated by SOS. The SOS search

process begins with a random initial population of hyper-parameters. Five sets of learning and validation subsets are employed for every iteration to carry our parameter search simulations. It follows the partitioning through the 5-fold cross-validation method. The average MSE value of the five validation subsets represents the objective value of the search process. The best parameter is the parameter set generating the minimum average MSE on validation subsets during the course of ten rounds of training simulation.

Following the identification of the best parameter set in the training process, the test set is used to assess the trained LSSVR model. In this proposed framework, the capacity of LSSVR in dealing with curve fitting and learning is combined with the capacity of SOS for optimization the two LSSVR parameters ( $\gamma$  and  $\sigma$ ), resulting on reduction of prediction errors.

## System Applications

### Description of Data Set

The historical data set for experiment was obtained from previous literature [22]. The data set has a total of 38 records of concrete mix proportion and was used to examine the behavior of concrete mixture with strength at seven days. Every data set has five input variables and one output variable, including cement, fine aggregate (FA), coarse aggregate (CA), water to cement ratio (W/C), 7-day strength test result ( $f_{c7}$ ), and the 28-day strength test result ( $f_{c28}$ ), respectively. The characteristics of the variables of each data set can be seen in Table 1.

**Table 1.** Statistical Description of Concrete Mix Proportion

Variables	Unit	Min	Max	Mean	Standard deviation
$X_1$ : Cement	kg/m <sup>3</sup>	270.00	517.00	378.58	61.13
$X_2$ : FA	kg/m <sup>3</sup>	468.00	900.00	688.34	131.04
$X_3$ : CA	kg/m <sup>3</sup>	900.00	1300.00	1050.50	115.27
$X_4$ : W/C	-	0.42	0.70	0.55	0.08
$X_5$ : $f_{c7}$	MPa	13.90	41.90	21.92	6.14
$Y$ : $f_{c28}$	MPa	21.83	50.50	31.47	6.49

### Machine Learning Models for Comparison

In order to compare the SOS-LSSVR model, this research applies three widely used machine learning models as follows:

#### Standard SVR Model

This SVR model used in this study is a traditional SVR model named  $\epsilon$ -SVR [23]. One can simply determine the parameters' value according to some

experimental rules, and the standard SVR model does the learning process by using all the training set. The standard SVR model performs the learning process by employing all the training sets. The values of parameters can be determined by some experimental rules. As suggested by Chang and Lin [23], the following are the values of parameters used in this study:  $C = 1$  and  $\gamma = 0.2$ ,

#### Standard LSSVR Model

LSSVR are recognized as least squares versions of the standard SVR model. Rather than solving convex quadratic programming problem for classical SVR, LSSVR uses a set of linear equation to come to a solution. As suggested by Suykens and Vandewalle [8], the following are the values of parameters used in this study:  $\gamma = 1$  and  $\sigma = 1$ ,

#### Levenberg-Marquardt Backpropagation Neural Network (LM-BPNN) Model

BPNN is introduced as a variant of neural network (NN). It is characterized by the fact that computation of the gradient of error function proceed backwards through the network. To minimize the error, the procedure is repeated during learning, and the weights are adjusted by the back propagation of error. Levenberg-Marquardt optimization is followed to update bias and weight values in LM-BPNN. The following are the values of parameters used in this study: minimum performance gradient =  $1E-07$ , initial  $\mu = 0.001$ ,  $\mu$  decrease factor = 0.1,  $\mu$  increase factor = 0.1, maximum  $\mu = 1E+10$ .

### Model Evaluation Methods

Four different performance metrics are employed for evaluating the predictive methods, namely coefficient correlation ( $R$ ), root mean square error ( $RMSE$ ), mean absolute percentage error ( $MAPE$ ), and mean average error ( $MAE$ ). They are applied on the predicted output results of the test data set. The following equations express the performance metrics:

$$R = \frac{n \sum y_i y'_i - (\sum y_i)(\sum y'_i)}{\sqrt{n(\sum y_i^2) - (\sum y_i)^2} \sqrt{n(\sum y'^2) - (\sum y'_i)^2}} \quad (13)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - y'_i)^2} \quad (14)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - y'_i| \quad (15)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - y'_i}{y_i} \right| \quad (16)$$

where  $y_i$  and  $y_i'$  are the  $i$ -th forecasted and actual values respectively, and  $n$  is the total number of prediction samples.

The best model outcome is indicated by the highest  $R$  value and the lowest  $MAE$ ,  $MAPE$ , and  $RMSE$  values.

### The Process of Training and Model Selection

For construction and validation of the prediction models, the entire data set of 38 samples is divided into training set (80%) and test set (20%) randomly. Thus, the number of samples in the training and test sets are 30 and 8, respectively. Furthermore, the training set is additionally divided into training and validation sets to improve the performance of the trained model. It is essential to prevent imbalance and non-uniformity of either of subsets to avoid overfitting of the trained LSSVR. Hence, the integration of integrating the 5-fold cross validation method as a sampling technique contributes to the more balanced partitioning process between the training and validation subset.

The search space of the two tuning hyper-parameters,  $\gamma$  and  $\sigma$ , varies from  $1E-10$  to  $1E+10$ . The initial numbers of organisms and SOS iterations are both set to 50 for searching the global optimum region. SOS carries out a search strategy by governing iteratively a large number of organisms through symbioses-inspired relationship operators to a promising candidate region in the search space. SOS can effectively automate the model selection process in comparison to the trial-and-error based on user's prior knowledge.

SOS-LSSVR simulates five rounds of training and validation subsets for each iteration. The average MSE value of the validation subset of each fold is stored as the objective value. The historical records of the model selection process are given in Figure 2a. The beginning of the objective value of this training process is an initial validation  $MSE$  value and, subsequently, it decreases iteratively to converge with a lower validation  $MSE$  value as shown in Figure 2b. The final set of parameters that produces the lowest  $MSE$  value on validation subsets are  $5.2531E+09$  for  $\gamma$  and  $9.9283E+04$  for  $\sigma$ .

### Prediction Results and Comparison

The training process is carried out for every machine learning model in this study. A test set is applied to each trained model as it is necessary to evaluate the training of all developed models. In Figure 3, the training and test results of the SOS-LSSVR prediction model are shown when the final set of hyper-parameters is used. Table 2 shows the full prediction results for machine learning models.

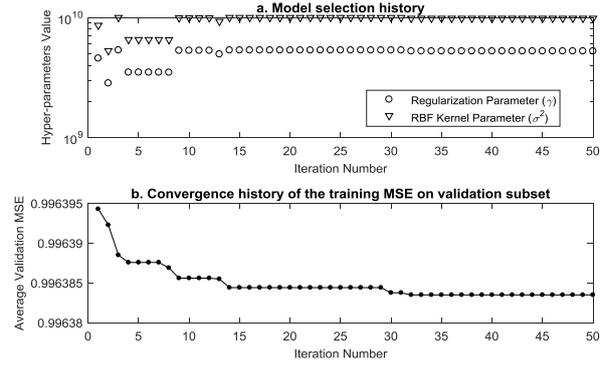


Figure 2. The Process of Model Selection and Training Process of SOS-LSSVR

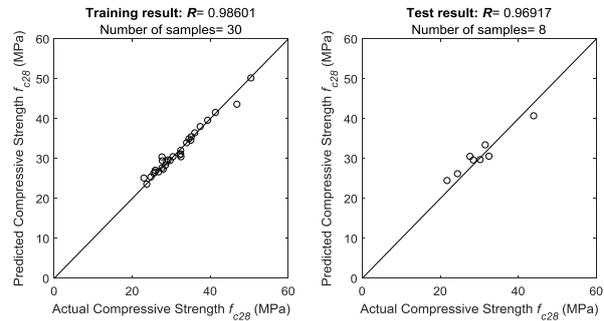


Figure 3. Obtained Training and Test Results of the Proposed SOS-LSSVR Prediction Model

The study applied four model performance metrics on the prediction results of every machine learning model. The linear association strength between actual and predicted output values is demonstrated by the  $R$ . The difference between actual and predicted output values is quantified by the  $RMSE$ . The average magnitude of errors between actual and predicted output values is calculated by the  $MAE$ , while the direction of errors is disregarded. The absolute errors are determined by the  $MAPE$ . Nevertheless, unlike in the case of the  $MAE$ , the size and unit of actual and predicted output values do not affect the  $MAPE$ . The error performance of every machine learning model on the test set is shown in Table 3.

The results showed that the construction of an optimized predictive model is facilitated more efficiently with the SOS-LSSVR model than with the default LSSVR model. SOS-LSSVR improves the performance metrics of  $MAPE$ ,  $MAE$ ,  $RMSE$ , and  $R$  by 3.41%, 1.08 MPa, 1.79 MPa, and 0.0575 by implementing the self-optimized framework, in comparison to the original LSSVR test results. Moreover, SOS-LSSVR also outperformed in every performance metrics SVR and LM-BPNN, other machine learning models. This detailed evaluation confirmed on the basis of laboratory test records the capacity of SOS and LSSVR to model the accurate compressive strength.

**Table 2.** Prediction Results for Training and Test Set

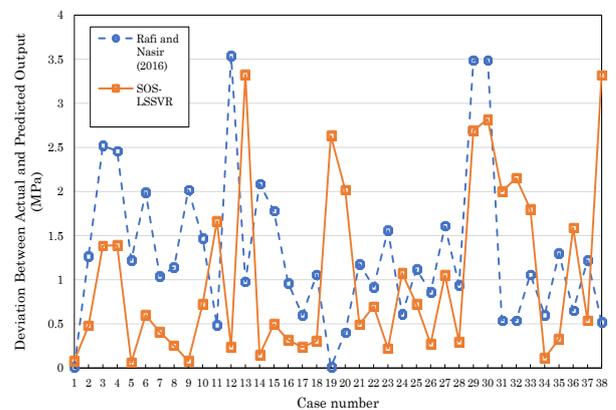
Data division	Case number	$X_1$ : Cement (kg/m <sup>3</sup> )	$X_2$ : FA (kg/m <sup>3</sup> )	$X_3$ : CA (kg/m <sup>3</sup> )	$X_4$ : W/C	$X_5$ : $f_{cr}$ (MPa)	Y: Actual $f_{c28}$ (MPa)	Y': Predicted $f_{c28}$ (MPa)					
								Rafi and Nasir [22]	Present Study				
								SVR	LSSVR	LM-BPNN	SOS-LSSVR		
Training	1	400	600	1200	0.45	24.5	34.01	34.00	30.23	33.29	34.04	33.93	
	2	390	588	1170	0.5	22.5	32.50	31.23	30.24	32.55	32.72	32.02	
	3	380	570	1140	0.55	21.6	32.50	29.98	30.34	32.15	32.35	31.11	
	4	370	555	1110	0.6	21.5	32.30	29.84	30.48	31.88	32.73	30.91	
	5	360	540	1080	0.65	21.1	30.50	29.28	30.60	31.57	32.70	30.44	
	7	325	650	1300	0.45	20.3	29.21	28.17	29.31	30.65	30.69	29.62	
	8	320	640	1280	0.5	20	28.90	27.76	29.37	29.65	29.24	29.15	
	9	315	630	1260	0.55	18.5	27.70	25.68	29.39	28.77	27.76	27.62	
	10	310	620	1240	0.6	17.6	25.90	24.43	29.46	28.15	27.83	26.62	
	12	300	600	1200	0.7	14.6	23.80	20.26	29.57	28.06	24.89	23.56	
	14	504	504	1008	0.5	29.9	39.41	41.50	31.84	36.80	37.78	39.56	
	15	491	491	982	0.55	28.3	37.50	39.28	31.88	36.66	36.94	38.00	
	16	479	479	958	0.6	26.7	36.10	37.06	31.90	35.66	36.06	36.42	
	17	468	468	936	0.65	25.8	35.21	35.81	31.93	34.80	35.66	35.45	
	18	457	547	914	0.7	25.7	34.61	35.67	31.80	34.22	33.72	34.92	
	20	385	767	1056	0.48	14.44	23.07	22.67	29.57	26.42	24.14	25.09	
	21	356	797	1057	0.52	15.08	24.85	23.67	29.46	27.12	24.85	25.34	
	22	396	744	1056	0.48	15.72	25.58	24.66	29.74	26.90	25.48	26.28	
	23	396	744	1057	0.48	16.12	26.85	25.29	29.77	26.99	25.95	26.63	
	24	365	775	1056	0.52	16.94	25.98	26.59	29.66	27.32	27.03	27.05	
	25	356	825	1021	0.52	17.13	28.01	26.89	29.62	28.47	27.85	27.29	
	26	396	776	985	0.48	17.66	28.58	27.72	29.99	28.40	29.26	28.31	
	28	365	808	1023	0.52	19.57	29.78	30.72	29.84	29.53	30.19	29.48	
	29	402	780	985	0.46	19.86	27.69	31.18	30.11	29.77	31.64	30.38	
	32	380	790	1021	0.5	20.41	32.56	32.02	29.98	29.68	31.25	30.41	
	34	320	860	900	0.6	33	41.40	40.80	30.83	38.20	41.27	41.51	
	35	380	810	900	0.5	41.9	50.50	51.80	31.55	43.24	50.36	50.17	
	36	270	900	900	0.7	20.2	27.80	27.15	29.95	30.65	27.91	29.39	
	37	320	860	900	0.6	25.2	35.10	33.88	30.36	34.88	34.93	34.56	
	38	380	810	900	0.5	34.5	46.90	46.38	31.15	42.39	46.85	43.58	
	Test	6	350	525	1050	0.7	20.4	30.30	28.31	30.70	31.79	32.79	29.70
		11	305	610	1220	0.65	17.3	24.50	24.01	29.57	27.78	28.17	26.16
		13	517	517	1034	0.45	31.01	44.01	43.03	31.75	35.65	38.23	40.69
		19	370	781	1055	0.5	13.9	21.83	21.82	29.46	26.42	23.61	24.46
		27	402	780	1023	0.46	19.23	28.56	30.17	29.97	28.93	30.00	29.61
		30	440	740	1021	0.42	19.86	27.69	31.18	30.23	31.13	30.21	30.51
		31	396	776	1021	0.48	20.41	32.56	32.02	30.06	29.56	31.17	30.56
	33	270	900	900	0.7	24.7	31.60	30.54	30.20	31.57	30.77	33.40	

**Table 3.** Comparative Prediction Results on Test Set.

Methods	Test set			
	$R$	$RMSE$ (MPa)	$MAE$ (MPa)	$MAPE$ (%)
SVR	0.9281	5.60	4.15	13.88
LSSVR	0.9117	3.95	3.07	10.17
LM-BPNN	0.9468	2.90	2.49	8.19
SOS-LSSVR	0.9692	2.16	1.99	6.76

The testing results were compared with those reported in the relevant literature to additionally validate the modeling performance of the proposed system. The deviation between actual and predicted output values between Rafi and Nasir’s work [22] and the proposed system are shown in Figure 4. The comparison revealed that out of total 38 samples, SOS-LSSVR achieved a lower prediction deviation in 26 samples. Furthermore, the average prediction

deviation of SOS-LSSVR is much lower (1.03 MPa) in comparison to those of Rafi and Nasir (1.30 MPa).



**Figure 4.** Deviation between Actual and Predicted Output Values from Rafi and Nasir [22] and SOS-LSSVR

## Conclusion

In this study, a new metaheuristic-based machine learning system called SOS-LSSVR was proposed to accurately estimate the compressive strength considering the concrete properties and the early age-test result. To investigate the performance of the proposed SOS-LSSVR system, three machine learning techniques were employed for benchmarking purpose. The laboratory test of 38 samples obtained from the past literature served as the experimental data set. The data set was divided into training and test sets and was used for building and validating the prediction model. Subsequently, the training set was divided into training subset and validation subset generated by 5-fold cross-validation to avoid the overfitting problem. This study used four performance metrics ( $R$ ,  $RMSE$ ,  $MAE$ , and  $MAE$ ) to further compare the proposed SOS-LSSVR for the overall performance outcome of the applied predictive techniques. According to the results, the proposed SOS-LSSVR has the highest accuracy for all mentioned performance metrics.

This study presents a significant contribution to address the modelling of concrete behavior considering the mixture properties and early-age test result. The SOS-SVR helps the concrete designers and the users in decision-making processes on the basis of early strength test results as it predicts the concrete behavior accurately. As demonstrated by the analytical results, SOS-LSSVR is the most reliable model for predicting accurately behavior of concrete mix proportion at an early-age of strength.

## Acknowledgment

The authors gratefully acknowledge that the present research is financially supported by The Ministry of Research, Technology and Higher Education of the Republic of Indonesia under the PDUPT Research Grant Scheme.

## References

1. Kheder, G.F., Gabban, A.M.A., and Abid, S.M., Mathematical Model for the Prediction of Cement Compressive Strength at the Ages of 7 and 28 Days within 24 Hours, *Materials and Structures*, 36(10), 2003, pp. 693.
2. Cheng, M.-Y. and Prayogo, D., Symbiotic Organisms Search: A New Metaheuristic Optimization Algorithm, *Computers & Structures*, 139, 2014, pp. 98-112.
3. Cheng, M.-Y., Prayogo, D., and Wu, Y.-W., Novel Genetic Algorithm-Based Evolutionary Support Vector Machine for Optimizing High-Performance Concrete Mixture, *Journal of Computing in Civil Engineering*, 28(4), 2014.
4. Cheng, M.-Y., Wibowo, D.K., Prayogo, D., and Roy, A.F.V., Predicting Productivity Loss Caused by Change Orders using the Evolutionary Fuzzy Support Vector Machine Inference Model, *Journal of Civil Engineering and Management*, 21(7), 2015, pp. 881-892.
5. Hoang, N.-D. and Pham, A.-D., Hybrid Artificial Intelligence Approach based on Metaheuristic and Machine Learning for Slope Stability Assessment: A Multinational Data Analysis, *Expert Systems with Applications*, 46, 2016, pp. 60-68.
6. Cheng, M.-Y., Prayogo, D., Ju, Y.-H., Wu, Y.-W., and Sutanto, S., Optimizing Mixture Properties of Biodiesel Production using Genetic Algorithm-based Evolutionary Support Vector Machine, *International Journal of Green Energy*, 13(15), 2016, pp. 1599-1607.
7. Hoang, N.-D., Tien Bui, D., and Liao, K.-W., Groutability Estimation of Grouting Processes with Cement Grouts using Differential Flower Pollination Optimized Support Vector Machine, *Applied Soft Computing*, 45, 2016, pp. 173-186.
8. Suykens, J.A.K. and Vandewalle, J., Least Squares Support Vector Machine Classifiers, *Neural Processing Letter*, 9(3), 1999, pp. 293-300.
9. Cheng, M.-Y. and Hoang, N.-D., Estimating Construction Duration of Diaphragm Wall using Firefly-Tuned Least Squares Support Vector Machine, *Neural Computing and Applications*, 2017, pp. 1-9.
10. Prayogo, D., Cheng, M.Y., Widjaja, J., Ongkowiwoyo, H., and Prayogo, H. Prediction of Concrete Compressive Strength from Early Age Test Result Using an Advanced Metaheuristic-Based Machine Learning Technique. in *ISARC 2017 - Proceedings of the 34<sup>th</sup> International Symposium on Automation and Robotics in Construction*, 2017.
11. Panda, A. and Pani, S., A Symbiotic Organisms Search Algorithm with Adaptive Penalty Function to Solve Multi-Objective Constrained Optimization Problems, *Applied Soft Computing*, 46, 2016, pp. 344-360.
12. Verma, M., Thirumalaiselvi, A., and Rajasankar, J., Kernel-based Models for Prediction of Cement Compressive Strength, *Neural Computing and Applications*, 28(1), 2017, pp. 1083-1100.
13. Prayogo, D., *An Innovative Parameter-Free Symbiotic Organisms Search (SOS) for Solving Construction-Engineering Problems*, 2015, PhD thesis, Department of Construction Engineering, National Taiwan University of Science and Technology.
14. Cheng, M.-Y., Prayogo, D., and Tran, D.-H., Optimizing Multiple-Resources Leveling in Multiple Projects using Discrete Symbiotic Organisms Search, *Journal of Computing in Civil Engineering*, 30(3), 2016, pp. 04015036.

15. Tran, D.-H., Cheng, M.-Y., and Prayogo, D., A Novel Multiple Objective Symbiotic Organisms Search (MOSOS) for Time–Cost–Labor Utilization Tradeoff Problem, *Knowledge-Based Systems*, 94, 2016, pp. 132-145.
16. Prayogo, D., Cheng, M.-Y., and Prayogo, H., A Novel Implementation of Nature-Inspired Optimization for Civil Engineering: A Comparative Study of Symbiotic Organisms Search, *Civil Engineering Dimension*, 19(1), 2017, pp. 36-43.
17. Yu, V.F., Redi, A.A.N.P., Yang, C.-L., Ruskartina, E., and Santosa, B., Symbiotic Organisms Search and Two Solution Representations for Solving the Capacitated Vehicle Routing Problem, *Applied Soft Computing*, 52, 2017, pp. 657-672.
18. Tejani, G.G., Savsani, V.J., Bureerat, S., and Patel, V.K., Topology and Size Optimization of Trusses with Static and Dynamic Bounds by Modified Symbiotic Organisms Search, *Journal of Computing in Civil Engineering*, 32(2), 2018, pp. 04017085.
19. Prayogo, D., Gosno, R.A., Evander, R., and Limanto, S., Implementasi Metode Metaheuristik Symbiotic Organisms Search Dalam Penentuan Tata Letak Fasilitas Proyek Konstruksi Berdasarkan Jarak Tempuh Pekerja, *Jurnal Teknik Industri*, 19(2), 2018, pp. 103-114.
20. Cheng, M.-Y., Chiu, C.-K., Chiu, Y.-F., Wu, Y.-W., Syu, Z.-L., Prayogo, D., and Lin, C.-H., SOS Optimization Model for Bridge Life Cycle Risk Evaluation and Maintenance Strategies, *Journal of the Chinese Institute of Civil and Hydraulic Engineering*, 26(4), 2014.
21. Kohavi, R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, in *International Joint Conference on Artificial Intelligence*, 1995, Stanford, CA.
22. Rafi, M. and Nasir, M., Models for Prediction of 28-Day Concrete Compressive Strength, *Journal of Testing and Evaluation*, 44(3), 2016, pp. 1217-1228.
23. Chang, C.-C. and Lin, C.-J., LIBSVM: A Library for Support Vector Machines, *ACM Transactions on Intelligent Systems and Technology*, 2(3), 2011, pp. 1-27.