Prediction of High-Performance Concrete Strength Using a Hybrid Artificial Intelligence Approach

Doddy Prayogo[1,*], Foek Tjong Wong[1], and Daniel Tjandra[1]

[1]Petra Christian University, Department of Civil Engineering, Jalan Siwalankerto 121-131 Surabaya 60236, Indonesia

Abstract. This study introduces an improved artificial intelligence (AI) approach called intelligence optimized support vector regression (IO-SVR) for estimating the compressive strength of high-performance concrete (HPC). The nonlinear functional mapping between the HPC materials and compressive strength is conducted using the AI approach. A dataset with 1,030 HPC experimental tests is used to train and validate the prediction model. Depending on the results of the experiments, the forecast outcomes of the IO-SVR model are of a much higher quality compared to the outcomes of other AI approaches. Additionally, because of the high-quality learning capabilities, the IO-SVR is highly recommended for calculating HPC strength.

1 Introduction

High-performance concrete (HPC) is widely used in the construction sector in a variety of projects because its superior strength, workability and durability surpass those of regular concrete. Specific materials are used to produce these concretes in order to meet the performance demands. The most critical property of HPC is its compressive strength. As HPC is being used more and more in the construction field, improving the forecasting capabilities of the compression strength of this concrete is extremely helpful in choosing the proper concrete mixtures [1,2].

The uniaxial compression test is commonly used to determine the compressive strength of concrete. The usual formula-based approaches restrict predictive functionality and have been proven to be unable to deliver acceptable performances, due to a variety of conditions and materials which could affect the compressive strength. Since the correlation between compressive strength and concrete materials are highly nonlinear, mathematical modeling of HPC is rather difficult, and in many cases inaccurate [3].

In past decades, artificial intelligence (AI) methods have proven to be feasible and powerful techniques for studying and evaluating the compressive strength of HPC. In 1998, Yeh [3] collected a large set of HPC database and proposed an artificial neural network (ANN) for establishing the compressive strength model of HPC. Following the dataset collected by Yeh [3], many investigations have been carried out to establish the accurate

---

* Corresponding author: prayogo@petra.ac.id

prediction model. Chou, et al. [4] proposed five different AI methods to optimize the prediction accuracy of compressive strength of HPC. Among AI strategies, the least squares support vector regression (LS-SVR) is a highly efficient procedure for handling learning and prediction issues in complex situations [5-7]. Even so, to the best of our understanding, the use of the LS-SVR in approximating the compressive strength of concrete is restricted when compared with other AI methods.

Additionally, the LS-SVR execution requires a suitable setting of its hyper-parameter [8]. This is because the LS-SVR hyper-parameter directly affects the complexity of the model. Given that the process of parameter estimation could be developed as an integrated optimization problem, applying metaheuristic optimization algorithms, together with AI-based learning techniques, may create a potential new compressive strength predictive method.

This study develops an improved AI method for estimation of the compressive strength of HPC. The method is called intelligence optimized support vector regression (IO-SVR), and it is based on the integration of the LS-SVR with a symbiotic organisms search (SOS) optimization algorithm for hyper-parameter selection. A dataset of HPC laboratory experiments was used to construct and verify the new prediction model. Subsequently, its prediction performance was compared with other prediction methods in terms of mean absolute percentage error (*MAPE*), mean absolute error (*MAE*), root mean squared error (*RMSE*), and linear correlation coefficient (*R*).

2 Methodology

2.1 Regression model: least squares support vector regression

LS-SVR was first introduced by Suykens and Vandewalle [9] as a modification of the conventional support vector regression (SVR). Where highly nonlinear spaces occur, an RBF kernel is chosen as the kernel function in LS-SVR, bringing more promising results than other kernels [9]. The following model of interest underlies the functional relationship between one or more independent variables along with a response variable:

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

where $x \in R^n$, $y \in R$, and $\phi(x): R^n \to R^{nh}$ is the mapping to the high dimensional feature space, respectively, w is SVM margin vector, and b represents SVM bias term. In LS-SVR, given a training dataset $\{x_k, y_k\}_{k=1}^{N}$, the optimization problem is formulated as follows:

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

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

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:

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

where $L(w,b,e,\alpha)$ is Lagrange multiplier of w, b, e, α variables, and α represents SVM alpha dot products. The conditions for optimality are given by:

$$\begin{cases} \dfrac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{k=1}^{N} \alpha_k \phi(x_k) \\[2mm] \dfrac{\partial L}{\partial b} = 0 \rightarrow \sum_{k=1}^{N} \alpha_k = 0 \\[2mm] \dfrac{\partial L}{\partial e_k} = 0 \rightarrow \alpha_k = \gamma e_k, \quad k = 1,\ldots,N \\[2mm] \dfrac{\partial L}{\partial \alpha_k} = 0 \rightarrow w^T \phi(x_k) + b + e_k - y_k = 0, \quad k = 1,..N \end{cases} \tag{5}$$

After elimination of e and w, the following linear system is obtained:

$$\begin{bmatrix} 0 & 1_v^T \\ 1_v & \Omega + I/\gamma \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \tag{6}$$

where $\quad y = y_1,\ldots,y_N$, $1_v = [1;\ldots;1]$, and $\alpha = [\alpha_1;\ldots;\alpha_N]$. And the kernel function, $K(x_k, x_l)$, is applied as follows:

$$\Omega = \phi(x_k)^T \phi(x_l) = K(x_k, x_l) \tag{7}$$

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

$$y(x) = \sum_{k=1}^{N} \alpha_k K(x_k, x_l) + b \tag{8}$$

α_k and b are the solution to the linear system (5). The kernel function that is often utilized is an RBF kernel; a description is given as:

$$K(x_k, x_l) = \exp\left(-\frac{\|x_k - x_l\|^2}{2\sigma^2}\right) \tag{9}$$

where σ^2 is the kernel function parameter.

With the γ parameter, the imposed penalty (to data points that move away from the regression function) can be controlled. For the σ^2 parameter, this will have a direct impact on the smoothness of the regression function. To ensure the best performance of the predictive model, it should be noted that proper setting of these tuning hyper-parameters is required.

2.2 Hyper-parameter tuning: symbiotic organisms search optimization algorithm

In numerous optimization applications, the quest for optimality represents a challenging task. In nature, perfect solutions to problems have occurred through evolution, as the mechanism of natural selection has eliminated all deficient solutions. According to the studies presented in the literature reviews, nature-inspired metaheuristic algorithms such as SOS are efficient in solving difficult optimization problems. Hence, this study uses the adjusted SOS to optimize LS-SVR hyper-parameters, and to ensure superior prediction accuracy.

Cheng and Prayogo developed one of the most widely-used metaheuristic algorithms, named the SOS algorithm [10]. It is based on the symbiosis–dependency-based interaction–typical among natural organisms. As in the cases of other metaheuristic algorithms, the

purpose of the SOS is to lead the search process through special operators using candidate solutions. More precisely, it searches for organisms with candidate solutions to identify the global solution in the search space. A maximum number of evaluations and additional typical control parameters are needed. A selection mechanism is used to preserve improved solutions.

However, several crucial distinctions exist between SOS and other metaheuristic algorithms. To illustrate, algorithm-specific parameters are not necessary for the SOS, while metaheuristic algorithms such as particle swarm optimization (PSO) needs a proper parameter setting of a cognitive factor, inertia weight, and a social factor. In other words, SOS does not require additional work to tune the parameters. Inadequate parameter tuning is likely to lead to a finding of the obtained solutions in local optima regions. The SOS has been applied to optimization problem solving in many research areas since its development in 2014 [11-17].

First, the SOS generates a random ecosystem matrix (population). All problems have particular viable candidate solutions. The ecosystem size refers to the number of organisms that can be entered into the ecosystem. In each matrix row, it represents organisms identical to individuals in numerous other solutions. With each virtual organism, the corresponding objective and a candidate solution are represented. The search starts following the creation of the ecosystem. There are three phases in the searching process, in which the organisms benefit from the interaction (mutualism, commensalism, and parasitism). The fitness of the updated organism must be enhanced in order to replace the current organisms. The best organism can be updated after finishing all phases. To conclude, the phases repeat in a continual cycle until the stopping criterion has been met.

The three rules of symbioses are applied by the SOS. The first one is mutualism symbiosis, which refers to the reciprocative benefits of two living organisms. The second one is commensalism symbiosis; in this relationship one organism takes all the benefits from the other, which is not substantially affected by this interaction. The third one is parasitism symbiosis. In this relationship, the benefits that an organism gains from the other are a detriment to this other organism. The mathematical equations used to model the SOS are described in the following subsections.

2.2.1 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:

$$newO_i = currentO_i + rand(0,1) * (bestO - mutualO_{ij} * BF_1) \qquad (10)$$

$$newO_j = currentO_j + rand(0,1) * (bestO - mutualO_{ij} * BF_2) \qquad (11)$$

where $currentO_i$ and $currentO_j$ are two current organisms involved in mutualism; $bestO$ is the current best organism; $rand(0,1)$ represents the uniform random value between 0 to 1; $mutualO_{ij}$ models the mutualism interaction of current organisms; $newO_i$ and $newO_j$ are the updated organisms following the interaction; BF_1 and BF_2 represent two random values of either 0 or 1 illustrating the level of benefit each organism has. The following formulation is used to calculate $mutualO_{ij}$.

$$mutualO_{ij} = (currentO_i + currentO_j) / 2 \qquad (12)$$

2.2.2 Commensalism Phase

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

$$newO_i = currentO_i + rand(-1,1) * (bestO - currentO_j) \qquad (13)$$

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

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

$$parasiteO_i = F * currentO_i + (1 - F) * [rand(0,1) * (ub - lb) + lb] \qquad (14)$$

where *parasiteO_i* is the artificial parasite engaged with *currentO_i*, and it threatens the existence of *currentO_j*; *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, respectively.

2.3 Performance measurement methods

Table 1 shows the performance measures for evaluating the predictive methods. They are applied on the predicted output results of the training and test datasets. As *MAPE* is unaffected by the unit and size of predicted and actual values, it is efficient in determining the relative differences between models. *MAE* disregards the direction of errors while calculating the average magnitude of errors between actual and predicted errors. The average distance of a data point from the fitted line, which is measured along a vertical line, is *RMSE*. The linear association strength between two variables is measured by *R*. Here, the value $R = -1$ represents a perfect negative correlation, while the value $R = 1$ represents a perfect positive correlation. The best model outcome is indicated by the highest *R* value and the lowest *MAE*, *MAPE*, and *RMSE* values.

Table 1. Performance measurement methods.

Performance measurement	Mathematical formula
Correlation coefficient (*R*)	$R = \dfrac{n\sum y \times y' - (\sum y)(\sum y')}{\sqrt{n(\sum y^2) - (\sum y)^2}\ \sqrt{n(\sum y'^2) - (\sum y')^2}}$
Root mean squared error (*RMSE*)	$RMSE = \sqrt{\dfrac{1}{n}\sum_{i=1}^{n}(y - y')^2}$
Mean absolute percentage error (*MAPE*)	$MAPE = \dfrac{1}{n}\sum_{i=1}^{n}\left|\dfrac{y - y'}{y}\right|$
Mean absolute error (*MAE*)	$MAE = \dfrac{1}{n}\sum_{i=1}^{n}|y - y'|$

y represents the actual data point, *y'* represents the predicted data point

2.4 Cross-validation for partitioning the training dataset

Training and testing processes are critical in setting up the prediction model. First, a dataset is implemented in a training process to establish a model using the artificial intelligence method. The purpose of the model is to verify a new dataset. However, it is possible that an 'overfitting' phenomenon may occur when the whole dataset is used for training. In this phenomenon, the prediction model fits the dataset very well; however, it cannot be used for an unseen, new dataset. Thus, to avoid the overfitting problem, it is common to divide the training dataset into two subsets. The greater part of the training dataset is called a 'learning subset', whereas the smaller is referred to as a 'validation subset'. The smaller subset is employed to validate the model built.

The k-fold cross-validation technique is used for eliminating the randomness in partitioning the training dataset [18]. In the process, k-fold cross-validation generates k subsets from training dataset. They are always non-overlapping. As k is an unfixed parameter, it can be any adequate number. In this study, the value of k is 10. Accordingly, the data is divided into ten random groups of equal size. Nine subsets are employed as learning subsets. and one as a validation subset. To train the inference model (IO-SVR), the first (k-1) subsets are used, whereas the last k-th subset is employed for the validation of the training results. Being based on cross-validation, the process is repeated for k times so that all data subsets are employed once as the validation subset.

2.5 Intelligence optimized support vector regression (IO-SVR) framework

The IO-SVR procedure describes how the proposed method interacts with and uses training and test datasets in order to provide the best prediction results. Initially, the dataset obtained from the laboratory was divided randomly into a training dataset and a test dataset. The framework of the IO-SVR is presented in Fig. 1.
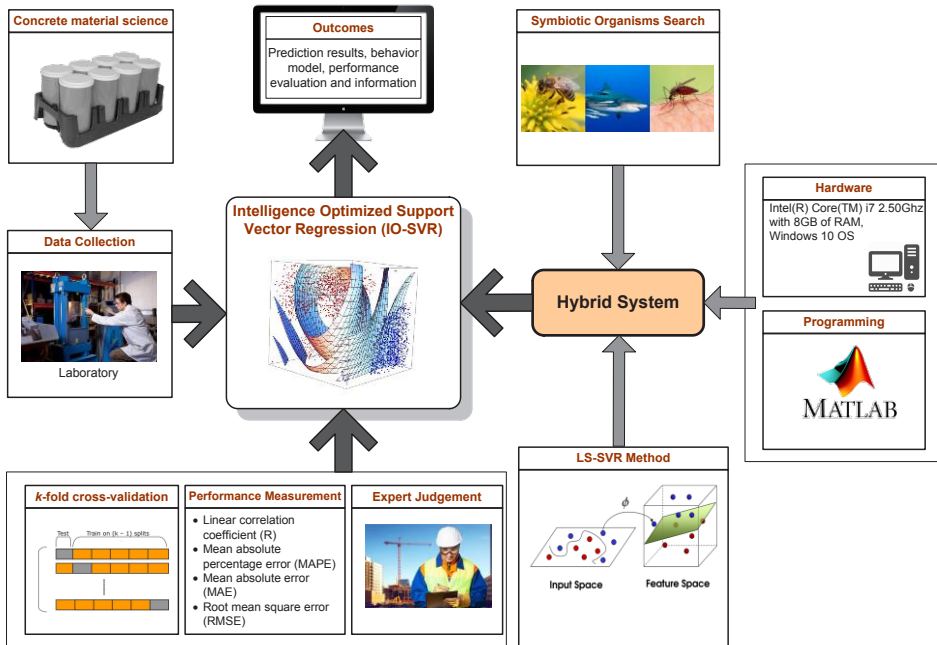

Fig. 1. IOS-SVR model architecture.

The SOS was allowed to identify the optimal LS-SVR parameters, and accordingly the predictive model sets were constructed in the training process. There are two parts of the training dataset, namely the 'learning subset' and the 'validation subset'. The purpose of this division is to eliminate the possibility of overfitting during the training process. To prevent bias during the sample partitioning process, the study selected the k-fold cross-validation procedure. First, the prediction model is made to fit a learning subset. The purpose is to fit the γ and σ^2 of the LS-SVR hyper-parameter of the model. A supervised learning method of the LS-SVR is used to train the model on the learning subset.

Subsequently, the fitted model is employed for predicting 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 learning subset, and it also tunes the hyper-parameters of the model. The prediction error from the validation dataset is observed by RMSE.

SOS simulates the optimization process of the parameter selection of LS-SVR. The searching process of SOS starts with a random initial population of hyper-parameters. For each iteration, 10 sets of learning and validation subsets are used to perform simulations of parameter searching, which was previously partitioned by the 10-fold cross-validation method. The objective value of the searching process is the average RMSE value of the 10 validation subsets. The parameter set which produces the minimum average RMSE on validation subsets, through ten rounds of training simulation, represents the best parameter.

After the best parameter set in the training process has been identified, the test dataset is employed to provide an assessment of the trained LS-SVR model. The IO-SVR framework combines the SOS's ability to optimize the two LS-SVR parameters (γ and σ^2) in order to reduce prediction errors, with the LS-SVR's ability to address curve fitting and learning.

3 Data collection and preparation

A published dataset [3] was utilized to evaluate the efficiency of the proposed AI-based regression model. Yeh [3] compiled a large database of uniaxial compression test records on HPC samples that was conducted by various university research labs. There are 1,030 samples in the uniaxial compression test database, each comprises eight input variables and one output variable. The input variables consist of many HPC properties, such as the amount of ordinary Portland cement, additives, and supplementary materials. The compressive strength of HPC is denoted as the only output variable. The performance of the proposed hybrid AI system to forecast the compressive strength of the HPC was assessed by those data.

The statistical descriptions of the variables are given in Table 2. As mentioned earlier, the HPC dataset was randomly partitioned into a training dataset and a test dataset containing 90% and 10% of dataset respectively. Based on the machine learning paradigm, the amount of training dataset must be significantly greater than the amount of test dataset. The language of technical computing called MATLAB was used to develop the IO-SVR code for modeling the HPC test database.

Table 2. HPC compression test properties and statistical descriptions.

Variables	Unit	Min	Mean	Max	Standard deviation
X_1: Cement	kg/m^3	102.0	281.17	540.0	104.51
X_2: Blast-furnace slag	kg/m^3	11.0	107.28	359.4	61.88
X_3: Fly ash	kg/m^3	24.5	83.86	200.1	39.99
X_4: Water	kg/m^3	121.8	181.57	247.0	24.35
X_5: Superplasticizer	kg/m^3	1.7	8.49	32.2	4.04
X_6: Coarse aggregate	kg/m^3	801.0	972.92	1,145.0	77.75
X_7: Fine aggregate	kg/m^3	594.0	773.58	992.6	80.18
X_8: Age of testing	Day	1.0	45.66	365.0	63.17
Y: HPC compressive strength	MPa	2.3	35.82	82.6	16.71

Following the original dataset partitioning, a data transformation process (e.g., normalization) is carried out to enhance the efficiency and accuracy of the support vector regression. The data input for analysis should be scaled to particular ranges, for instance, [0,1], to obtain adequate results. The dominance of a variable can be reduced by data transformation.

The following is a mathematical formulation for the attribute scaling method.

$$X_i^{norm} = (X_i - X_i^{min}) / (X_i^{max} - X_i^{min}) \tag{15}$$

where X_i is the original data, which are transformed into a normalized value X_i^{norm}, with $X_i^{norm} \in [0, 1]$, and X_i^{max} denotes the maximum value for X_i, and X_i^{min} denotes the minimum value for X_i.

4 Experimental results

The efficiency and applicability of the given hybrid model in predicting HPC strength on the basis of laboratory test records were assessed by benchmarking its performance relative to the performance levels of other AI models (i.e., LS-SVR and SVR). Training and test datasets were utilized to establish the HPC strength model. The following are determined suitable parameters: (1) maximum number of iterations for SOS is set to be 20, (2) population size for SOS is set to be 20, (3) search range for the γ and σ^2 are varied from 10^{-10} to 10^{10}.

A random initial population of hyper-parameter is used to start the training procedure and to build the initial strength prediction model. Moreover, 10-fold cross-validation of learning and validating subsets are simulated by IO-SVR for each iteration. The average RMSE value of the validation subset of each fold is kept as the objective value. This cross-validation technique is also utilized to prevent the sampling bias, and to ensure the optimum accuracy of the strength prediction model. A summary of the cross-validation training performance for the validation subset in the training period is given in Fig. 2.
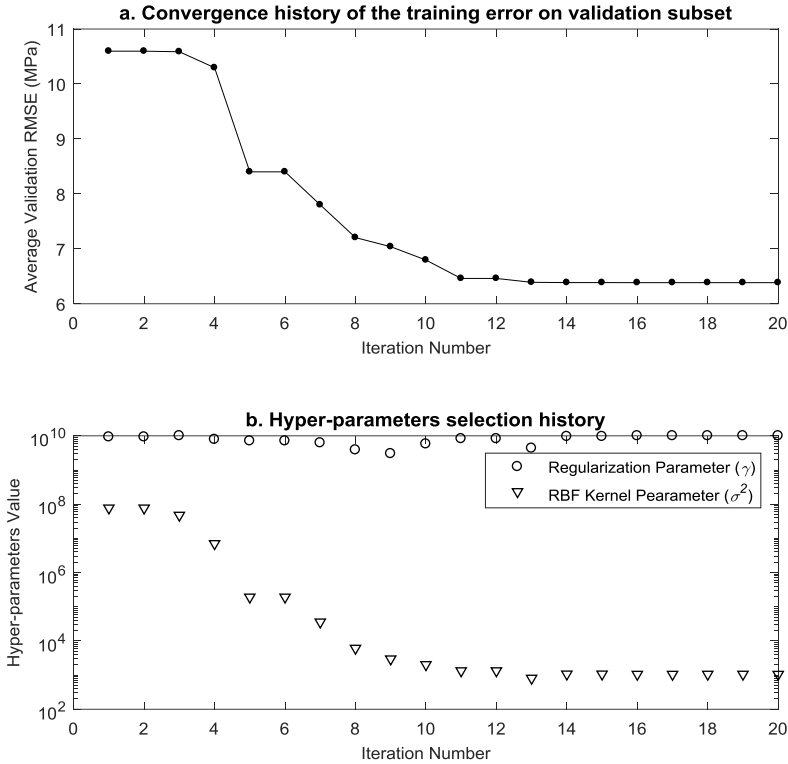
Fig. 2. Training performance of the proposed IO-SVR method.

The test results of IO-SVR are shown in Fig. 3. The predicted and actual output of testing phases demonstrated a good fit to a straight line. The *R* for the training and test phase obtained in this study demonstrate the superior performance and high accuracy of the trained IO-SVR model. As aforementioned, *MAE*, *MAPE*, and *RMSE* were used in addition to *R* to provide a more precise assessment of the performing methods. The prediction results of each method are given in Table 3 for the future analysis.
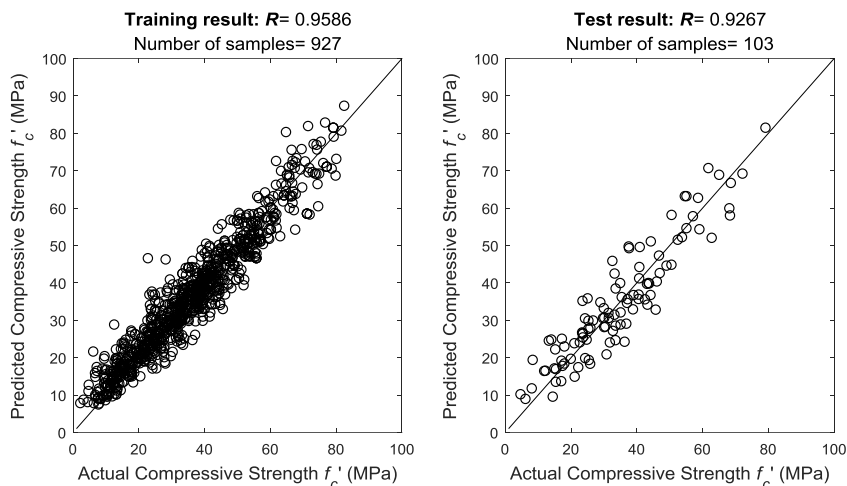
Fig. 3. Obtained results of IO-SVR strength prediction model for both training and test dataset.

Table 3. Comparative prediction results on test dataset.

Methods	R	RMSE (MPa)	MAE (MPa)	MAPE (%)	Parameters
SVR	0.7394	12.8244	10.2790	45.82	C=1, γ= 0.125
LS-SVR	0.7874	9.8838	7.7592	33.11	γ= 1, σ^2= 1
IO-SVR	0.9267	6.0288	4.9473	18.65	γ= 9976023522, σ^2= 1050.8

According to the results, the IO-SVR model facilitated efficiently the construction of an optimized predictive model over the default LS-SVR method. The performance measures of the *MAE*, *MAPE*, *RMSE*, and *R* of IO-SVR test results were enhanced by 2.8119 MPa, 14.46%, 3.8550 MPa, and 0.1873, respectively, by implementing the self-optimized framework. The capacity of LS-SVR and SOS for modeling the accurate compressive strength on the basis of laboratory test records was confirmed by this comprehensive assessment.

5 Conclusions

This study established an innovative method for predicting HPC concrete strength on the basis of an HPC compressive test. This research explored the capacity of the prediction of LS-SVR, and accordingly has contributed to the current body of knowledge. This model utilized real laboratory test records to obtain accurate prediction results. The primary purpose of this study was to explore the efficacy of a hybrid AI system in optimizing the LS-SVR parameters in order to enhance the accuracy of strength forecasting of HPC compressive strength.

Following the further analysis of the results, it is suggested that the combination of SOS and LS-SVR can significantly facilitate the creation of an optimized predictive model for forecasting the compressive strength. The accuracy errors (*R*, *MAE*, *MAPE*, and *RMSE*) obtained are quite remarkable, considering that modeling of nonlinear HPC material

behavior is challenging. This proposed hybrid AI system represents a robust and reliable tool for estimating HPC compressive strength, and it can greatly facilitate the work of concrete mix designers.

References

1. M.-Y. Cheng, P.M. Firdausi, D. Prayogo, High-performance concrete compressive strength prediction using Genetic Weighted Pyramid Operation Tree (GWPOT), Eng. Appl. Artif. Intell. **29**, 104-113 (2014)

2. M.-Y. Cheng, D. Prayogo, Y.-W. Wu, Novel Genetic Algorithm-Based Evolutionary Support Vector Machine for Optimizing High-Performance Concrete Mixture, J. Comput. Civ. Eng. **28** ,4, 06014003 (2014)

3. I.C. Yeh, Modeling of strength of high-performance concrete using artificial neural networks, Cem. Concr. Res. **28**,12, 1797-1808 (1998)

4. J.-S. Chou, C.-K. Chiu, M. Farfoura, I. Al-Taharwa, Optimizing the Prediction Accuracy of Concrete Compressive Strength Based on a Comparison of Data-Mining Techniques, J. Comput. Civ. Eng. **25**,3, 242-253 (2011)

5. N.-D. Hoang, D.T. Bui, Predicting earthquake-induced soil liquefaction based on a hybridization of kernel Fisher discriminant analysis and a least squares support vector machine: a multi-dataset study, B. Eng. Geol. Environ. **77**,1, 191-204 (2016)

6. D. Prayogo, Metaheuristic-Based Machine Learning System for Prediction of Compressive Strength based on Concrete Mixture Properties and Early-Age Strength Test Results, Civ. Eng. Dimens. **20**,1, 21-29 (2018)

7. D. Prayogo, Y.T.T. Susanto, Optimizing the Prediction Accuracy of Friction Capacity of Driven Piles in Cohesive Soil Using a Novel Self-Tuning Least Squares Support Vector Machine, Adv. Civ. Eng. **2018** (2018)

8. D. Tien Bui, B.T. Pham, Q.P. Nguyen, N.-D. Hoang, Spatial prediction of rainfall-induced shallow landslides using hybrid integration approach of Least-Squares Support Vector Machines and differential evolution optimization: a case study in Central Vietnam, Int. J. Digit. Earth **9**,11, 1077-1097 (2016)

9. J.A.K. Suykens, J. Vandewalle, Least Squares Support Vector Machine Classifiers, Neural Process. Lett. **9**,3, 293-300 (1999)

10. M.-Y. Cheng, D. Prayogo, Symbiotic Organisms Search: A new metaheuristic optimization algorithm, Comput. Struct. **139**, 98-112 (2014)

11. M.-Y. Cheng, D. Prayogo, D.-H. Tran, Optimizing Multiple-Resources Leveling in Multiple Projects Using Discrete Symbiotic Organisms Search, J. Comput. Civ. Eng. **30**,3, 04015036 (2016)

12. D.-H. Tran, M.-Y. Cheng, D. Prayogo, A novel Multiple Objective Symbiotic Organisms Search (MOSOS) for time–cost–labor utilization tradeoff problem, Knowl.-Based Syst. **94**, 132-145 (2016)

13. D. Prayogo, M.-Y. Cheng, H. Prayogo, A Novel Implementation of Nature-inspired Optimization for Civil Engineering: A Comparative Study of Symbiotic Organisms Search, Civ. Eng. Dimens. **19**,1, 36-43 (2017)

14. A. Panda, S. Pani, A Symbiotic Organisms Search algorithm with adaptive penalty function to solve multi-objective constrained optimization problems, Appl. Soft Comput. **46**, 344-360 (2016)

15. V.F. Yu, A.A.N.P. Redi, C.-L. Yang, E. Ruskartina, B. Santosa, Symbiotic organisms search and two solution representations for solving the capacitated vehicle routing problem, Appl. Soft Comput. **52**, 657-672 (2017)

16. G.G. Tejani, V.J. Savsani, S. Bureerat, V.K. Patel, Topology and Size Optimization of Trusses with Static and Dynamic Bounds by Modified Symbiotic Organisms Search, J. Comput. Civ. Eng. **32**,2, 04017085 (2018)

17. G.G. Tejani, V.J. Savsani, V.K. Patel, Adaptive symbiotic organisms search (SOS) algorithm for structural design optimization, Journal of Computational Design and Engineering **3**,3, 226-249 (2016)

18. R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection (International Joint Conference on Artificial Intelligence, **14**, Stanford, CA, 1995)