

RESEARCH

Open Access



Estimation of mechanical properties of the modified high-performance concrete by novel regression models

Li Jingtao¹, Wang Jing¹ and Yin Suyuan^{1*}

*Correspondence:
AprilDou@163.com

¹ Hebei Vocational College
of Rail Transportation,
Shijiazhuang 050000, China

Abstract

Using support vector regression (SVR) analytics, a novel method for evaluating the high-performance concrete (HPC) compressive strength (CS) containing fly ash (FA) and blast furnace slag (BFS) has been developed. Both Salp swarm optimization (SSA) and Grasshoppers optimization algorithm (GOA) were used in this research to look for critical SVR method variables that may be tweaked for better performance. The suggested approaches were created using 1030 trials, eight inputs (the primary component of admixtures, mix designs, curing age, and aggregates), and the CS as the forecasting goal. After that, the findings were compared to those found elsewhere in the literature. Combined SSA-SVR and GOA-SVR analysis could work exceptionally well when it comes to estimating, according to the estimation findings. The root means square error (RMSE) value for the GOA-SVR faces a remarkable increment in comparison with the SSA-SVR. The comparison resulted that the GOA-SVR delivered a higher rate of accuracy than any previous published research. At the outset, the developed GOA-SVR model might be considered a practical predictive system for the CS prediction of HPC admixed with FA and BFS.

Keywords: Mechanical properties, High-performance concrete, Optimized SVR analysis, Fly ash, Simulation, Blast furnace slag

Introduction

High-performance concrete (HPC) is more efficient than conventional concrete (CC) in most cases, depending on the applications. The American Concrete Institute (ACI) states that *HPC* is a concrete kind that satisfies specified characteristics for homogeneity and performance. Neither standard mixing or curing methods nor standard components can produce these qualities [1, 2]. There is a possibility that the durability of this concrete will be improved. Maintenance costs are reduced as well as service life is extended. Consider the accessibility and cost of locally available resources while selecting the HPC mix characteristics. As a result, more trial mix quantities and analysis are necessary than in CC [2]. Additional cementitious components and additives are added to CC to make HPC [2–4]. The compressive strength of HPC mixtures is critical. Establishing an appropriate estimation system for this critical attribute saves money and time while

also allowing for more efficient combination creation. Statistical regression approaches have been used by researchers in this regard [3, 4]. An experimental approach based on regression could have major flaws. There must be a formula in place before a regression analysis can be performed. Another significant limitation of the use of regression models is the necessity for continued regularity [2]. Nevertheless, there is disagreement on the experimental formulae used in codes and standards to establish CS. This is due to these formulae being developed by testing on concrete, which did not involve any components of additional cementitious materials.

Over the last decade, there has been a surge of interest in applying machine learning methods to tackle civil engineering challenges in both commercial and academic settings [5, 6]. Using machine learning methods, in which empirical efforts can create learning, could be highly beneficial in developing computer models [7, 8]. The most extensively utilized method for machine learning is artificial neural networks (ANNs). ANNs are often used to analyze a wide range of concrete qualities [9–12]. They have been used to forecast the CS and slump flow of HPC mixtures [13, 14]. Employing sequential learning NNs, Rajasekaran and Amalraj [15] and Rajasekaran et al. [16] developed prediction approaches for the power of HPC mixtures. A wavelet NN approach was utilized to analyze the HPC's CS by Rajasekaran and Lavanya [17]. Since ANNs are unable to characterize the prediction fundamentals, they are commonly referred to as black box systems. Despite their efficiency, since ANNs usually fails to give a good forecasting model, they are still considered as a black box. Various authors have forecasted the mechanical features of HPC in previous years utilizing NNs and other AI approaches. Gradient-boosted ANN and Bagged ANN were employed by Erdal et al. [18] to simulate the HPC's CS. When compared to previous trials, Chou and Pham's ensemble approaches [19] fared well. So long as Erdal [20] built two connected ensemble decision trees, Cheng et al. [21] evaluated HPC CS using a tree. According to the previous study, AI techniques are more effective than traditional methods for precisely and quickly measuring the CS of HPCs [22, 23]. Rafiei et al. [24] propose the introduction of a unique deep machine as a replacement for back-propagation NN and SVR. This is done to determine the properties of concrete on the basis of the mixture percentages. Considering real testing results, they assess the 98% precise. Nguyen et al. [25] used a deep NN framework to determine the foamed concrete strength. Rafiei et al. [26] used an optimization algorithm to uniquely handle the optimal values in the concrete mix. Genetic programming is another machine learning approach (GP) [27]. The criteria of natural genetic advancement lead GP to generate computer simulations on its own. In recent years, classical GP and its modifications have been used to deliver easier answers to civil engineering problems [28]. Mousavi et al. [2] combine GP and orthogonal minimum squares approaches to estimate the CS of HPC combinations. When creating CS, the amount of fine/coarse aggregate, the percentage of superplasticizer to samples' age, and the binder were all considered. Gene expression programming is a more modern development of GP (GEP) [29]. The GEP approach may be used as a dependable and effective substitute for traditional GP. Some study has been conducted in an attempt to use GEP in civil engineering [30–33].

Several research have proposed alternative models for predicting the CS of HPCs, including the hybrid adaptive neuro-fuzzy inference system (ANFIS) with arithmetic

optimization algorithm (AOA) and equilibrium optimizer (EO). The findings suggest that the integrated systems exhibited robust estimate skills, as shown by R^2 values of 0.9941 and 0.9975 for the training and testing phases, respectively [34]. Four deep learning algorithms were offered in different research, which was unusual in the literature. When using deep learning models, the aforesaid R^2 value was approximately 0.960 during training and almost reached 0.940 after testing [35]. In research, a unique hybrid model combining artificial bee colony (ABC) optimization and cascade forward neural network (CFNN) technology was created for the CS prediction of HPC. The created model (CFNN-ABC) was able to correctly estimate the compressive strength of HPC with an R^2 of 0.953, according to the results, and two layers is the ideal neural network architecture chosen by the ABC approach [36]. An incredibly accurate machine learning (ML) model was trained using the innovative eXtreme Gradient Boosting (XGB) approach. The baseline model tends to overfit, with values of 0.996 and 0.919 for the training and testing datasets, respectively [37].

The main objective of the current research is to supply a practical way to evaluate the effectiveness of intelligent machines in calculating HPC's CS via stringent testing. Using the SVR technique, we attempted to construct models that might predict HPC's qualities while it had fresh and hardened properties. It used the grasshoppers' optimization algorithm (GOA) and salp swarm optimization (SSA) algorithms in this work to pinpoint the most critical components of SVR that need to be changed. The created techniques were assessed using 1030 experiments, eight input parameters, such as concrete age, admixtures, the major constituent of mixes, and the compressive strength as the forecasting objective. The findings were then compared to other studies in the field [19, 28, 38–42].

Developed models offer several advantages compared to traditional mixed design methods in the field of materials science and civil engineering. These advantages stem from ML's ability to analyze large datasets, discover complex patterns, and make predictions based on learned patterns. Algorithms can analyze vast amounts of data to identify subtle relationships between various material properties and mix design outcomes. This can lead to mixed designs that are more accurate and precise, reducing the risk of defects and failures. Developed models can automate many aspects of mix design, such as optimizing the proportions of materials and predicting the properties of the resulting mixture. Models can adapt to different materials, conditions, and project requirements. Traditional mixed design methods often rely on simplified assumptions and linear models. Algorithms, on the other hand, can handle complex, nonlinear relationships between material properties and mix performance, leading to more accurate predictions. ML can help optimize mix designs to use materials more efficiently, minimizing waste and reducing costs associated with the overuse or underuse of materials. ML-based mix design can help reduce the cost of materials and construction by optimizing the use of resources, improving performance, and minimizing the need for expensive adjustments or revisions.

Methods

Dataset preparation and description

Over a thousand HPC sample datasets were investigated in this research [4, 43–46]. Regular Portland cement was used to build all specimens, which were then permitted

time to dry out. Several different types and sizes of samples have been used in the *HPC* data that has been released so far. In this study, the *HPCCS* is based on the following eight factors:

- (1) C Contents of cement
- (2) BFS Blast furnace slag
- (3) FA Fly ash
- (4) W Water
- (5) SP Superplasticizer
- (6) CA Coarse aggregate
- (7) FA Fine aggregate
- (8) AC *HPC* age

Figure 1 shows the training and evaluation datasets' charts with their Lognormal distribution, and Table 1 shows the collection range for these items. When it came time to divide up everything that was needed, 70% of the 1030 data was used for learning and the remainder for testing. Subgroups from the original dataset were selected at random using a uniform distribution for training and testing. The 70:30 train/test ratio was chosen for this investigation even though many other train/test ratios were able to be used, as suggested in previous studies [47, 48]. In order to show that the choice of these inputs was enough, a statistical study was carried out. As a consequence, no substantial cross in the eight-dimensional input space was found [45, 49], which is needed to train AI networks with good generalization abilities.

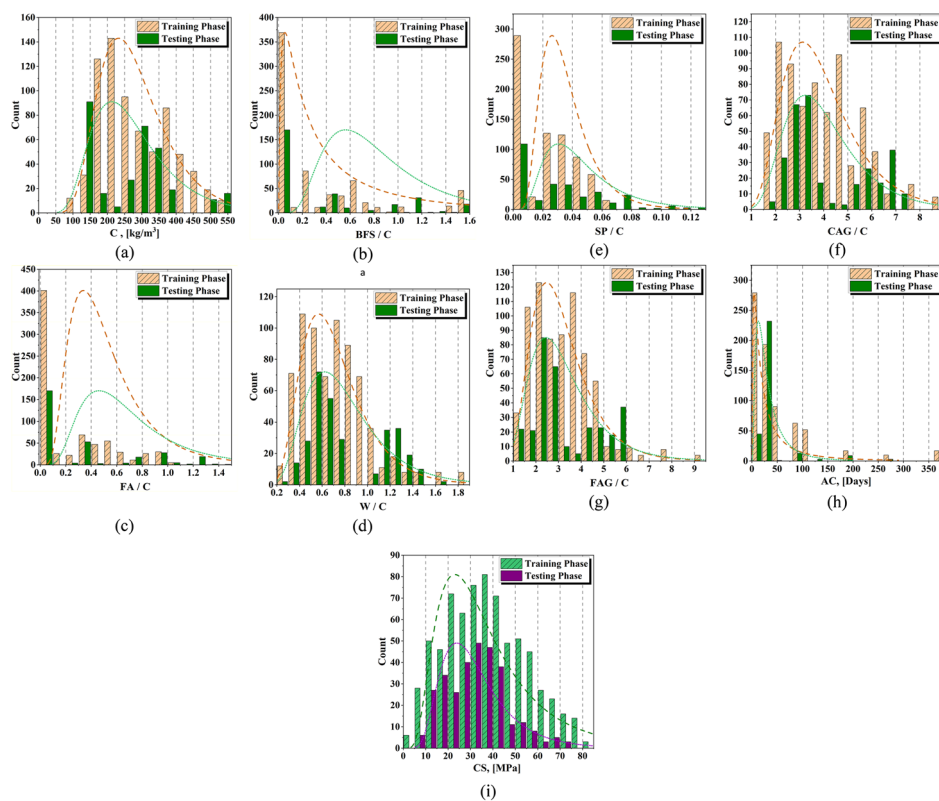


Fig. 1 Distribution of dataset and their Lognormal distribution

Table 1 The dataset attributes

Category	Parameter	Input variable							Output variable		
		C	BFS/C	FA/C	W/C	SP/C	CAG/C	FAG/C	AC	CS	MPa
Train	Min.	102	0.0	0.0	0.267	0.0	1.552	1.226	3.0	2.332	
	Max.	540	1.504	1	1.882	0.069	8.696	9.235	365.000	82.599	
	St.D.	101.84	0.446	0.306	0.298	0.02	1.549	1.323	67.717	17.718	
	Var.	10,371.318	0.198	0.094	0.089	0.0	2.399	1.751	4585.577	313.916	
	Range	438	1.504	1.000	1.615	0.069	7.144	8.009	362.000	80.267	
	Skew.	0.511	1.467	0.989	1.163	0.245	0.596	1.194	2.993	0.304	
	Kurt.	-0.692	1.284	-0.269	2.104	-1.243	-0.127	2.572	10.189	-0.549	
	Min.	132	0.0	0.0	0.2	0.0	1.716	1.135	1.0	6.267	
	Max.	540	1.584	1.430	1.694	0.125	7.148	5.993	360	74.987	
	St.D.	110.075	0.513	0.418	0.341	0.031	1.632	1.405	50.155	13.44	
Test	Var.	12,116.471	0.264	0.175	0.116	0.001	2.664	1.973	2515.551	180.634	
	Range	408	1.584	1.43	1.415	0.125	5.432	4.858	359	68.72	
	Skew.	0.533	1.036	1.043	0.554	0.873	0.580	0.556	4.229	0.499	
	Kurt.	-0.228	-0.314	-0.307	-1.068	0.190	-1.237	-0.956	20.129	0.214	

Considered techniques

Grasshoppers optimization algorithm (GOA)

Meta-heuristic methods are built on simulating nature (Fig. 2). Universal optimization issues are typically solved using these methods. Different categories of meta-heuristic methods include those based on evolution, physical and chemical, swarm intelligence, and human-based methods. We optimized using GOA in this research. Saremi et al. [50] introduce GOA to the research. The efficient meta-heuristic method GOA uses optimization based on a swarm, which was motivated by natural processes. Based on this study, GOA is utilized to determine the input variables' ideal values (ideal factor rates). The optimization process is then applied to GOA using these models. As a consequence, we employed GOA to look for these regression models' ideal factor rates.

GOA imitates the organic manners of grasshopper swarms. The two steps of nature-inspired optimization methods are exploitation and exploration. Sudden movements throughout the exploration are made by the search agents of the optimization method. Although they migrate more locally throughout exploitation. The following formulae represent the manners of the grasshopper and the theory of optimization search [50]:

$$X_i = r_1S_i + r_2G_i + r_3A_i \tag{1}$$

In this equation, i stands for every grasshopper and X_i indicates where the i_{th} grasshopper is located. S_i is a symbol of how grasshoppers connect. Similarly, to this, G_i and A_i stand in for gravity power and wind advection. The random numbers in the range [0, 1] are the r variables. Equation (2) [50] provides information on the grasshoppers' social manner (attraction–repulsion):

$$S_i = \sum_{\substack{j=1 \\ j \neq i}}^N s(d_{ij}) \widehat{d}_{ij} \tag{2}$$

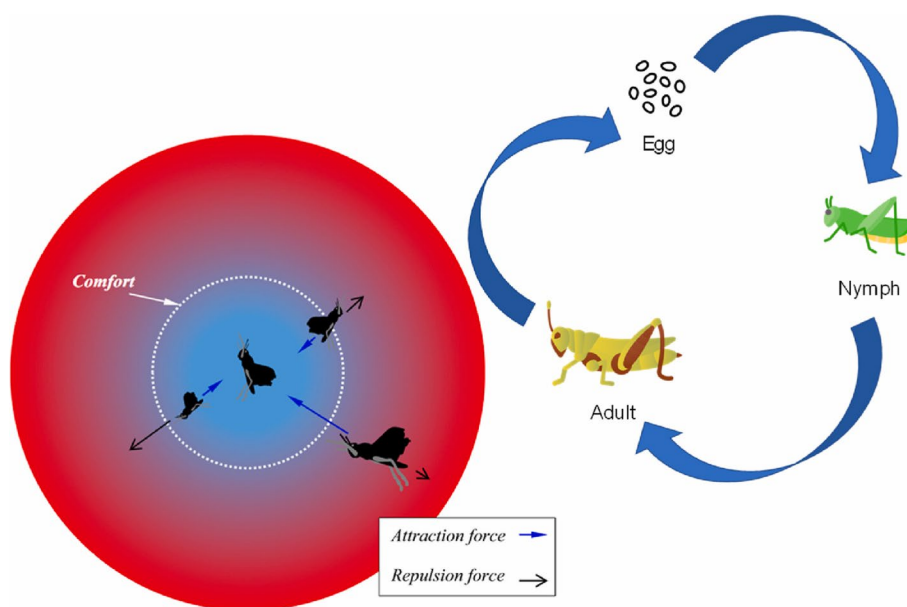


Fig. 2 Patterns of relations between individuals in a group of grasshoppers [51]

Based on this equation, s stands for the power of social forces ($s_r = fe^{-r/l} - e^{-r}$), l stands for the seductive interval scale, and f for attraction's potency [50]. N shows the grasshoppers' number. A vector among both two grasshoppers is called d_{ij} ($d_{ij} = |x_j - x_i|$), and \widehat{d}_{ij} represents the exact interval among the i th and j th grasshopper ($\widehat{d}_{ij} = (x_j - x_i)/d_{ij}$). Artificial grasshoppers' social interactions are influenced by the s function. This function divides the interval among every duo of grasshoppers into three pieces (attraction region, repulsion region, and comfort zone). In their investigation of intervals from 0 to 15, Saremi et al. [50] found repulsion among [0 2.079]. They proposed calculating the safe interval as the separation of two artificial grasshoppers by 2.079 units. The safe area is not where attraction or repulsion operate. This area is altered by the f and l . However, the s function turns to zero if this interval exceeds 10. Consequently, this function cannot generate powerful pressures throughout long intervals among grasshoppers. Another element of X_i is G_i (gravitational power) [50]:

$$G_i = -g\widehat{e}_g \tag{3}$$

In Eq. (3), g stands for the gravity constant and \widehat{e}_g for the unity vector pointing toward the middle of the earth. The last element of X_i is A_i (wind advection):

$$A_i = u\widehat{e}_w \tag{4}$$

According to this equation, respectively, \widehat{e}_w and u represent a unity vector and constant drift in the direction of the wind. Traditional swarm-based methods imitate the swarm as it explores and uses the search area all over an answer. The GOA model of X_i replicates the interactions of a swarm of grasshoppers since the mathematical formulas are in a free area. It emulates how a grasshopper might act in multiple spatial dimensions, including 2D, 3D, and hyper-dimensional areas [50].

$$X_i^d = c \left(\sum_{\substack{j=1 \\ j \neq i}}^N c \frac{ub_d - lb_d}{2} s(x_j^d - x_i^d) \frac{x_j - x_i}{d_{ij}} \right) + \widehat{T}_d \tag{5}$$

The lower and upper boundaries in the D_{th} dimension s_r are known as lb_d and ub_d . The comfort, repulsion, and attraction areas are reduced by the lowering coefficient c , which makes \widehat{T}_d the optimal (target) answer. Every search unit in GOA has a one-location vector, which calculates every search unit's next location. Equation (5)'s first step, the summation, replicates grasshopper interaction by considering the locations of several grasshoppers. \widehat{T}_d reflects their propensity to migrate in the direction of food resources. Eventually, c simulates the grasshoppers' slowing as they reach the food resource in Eq. (6) [50].

$$c = c_{max} - l \frac{c_{max} - c_{min}}{L} \tag{6}$$

In this equation, the present iteration and the iterations' highest number, L and l . The lowest and highest values are denoted c_{min} and c_{max} [50]. We utilized the same settings as Saremi et al. [50], who utilized $c_{max} = 1$ and $c_{min} = 0.00001$. In summation, the swarm eventually approaches a fixed goal as the safe area is reduced by the c variable. The swarm also successfully pursues a moving goal by \widehat{T}_d . The grasshoppers will

approach the objective through many repetitions. Algorithm 1 below displays the GOA pseudocode [50].

1.	Initialize $X_i, cmax, cmin$, and <i>max number of iterations</i>
2.	Calculate the fitness of each search agent
3.	Assign T = the best search agent
4.	While $l < \text{max number of iterations}$
5.	Update c by $c = cmax - l((cmax - cmin)/L)$
6.	for each search agent
7.	Normalize the d_{ij} between grasshoppers in [1,4]
8.	Update the position of the current search agent by X_i^d
9.	Bring the current search agent back if it goes outside the boundaries
10.	end for
11.	if there is a better solution, then update T
12.	$l = l + 1$
13.	end while
14.	Return T

Algorithm 1. GOA's Pseudocode

Salp swarm optimization (SSA)

As presented in Fig. 3, salps move highly alluringly, often traveling in a cooperative chain as they seek food in the oceans and seas. This characteristic allows the swarm to move more fluidly and with greater kinetic energy while they search for food [52]. A single leader often leads the salps chain followers. A d -dimensional matrix describes a group chain X with n salps.

$$X = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_d^1 \\ x_1^2 & x_2^2 & \dots & x_d^2 \\ \vdots & \vdots & \dots & \vdots \\ x_1^n & x_2^n & \dots & x_d^n \end{bmatrix} \tag{7}$$

A focused food establishment is also represented by F . Following are the modified positions for a leader (x_j^1):

$$x_j^1 = \begin{cases} F_j + k_1((U_{b-j} - L_{b-j})k_2 + L_{b-j}), & k_3 \geq 0.5 \\ F_j - k_1((U_{b-j} - L_{b-j})k_2 + L_{b-j}), & k_3 < 0.5 \end{cases} \tag{8}$$

Based on this equation, k_2 and k_3 are two randomly generated numbers in the range [0, 1], and U_b and L_b stand for the top and low boundaries. According to the following equation, k_1 maintains an equilibrium between exploitation and exploration:

$$k_1 = e^{\left(\frac{-4n}{N}\right)^2} \tag{9}$$



Fig. 3 Salp chain

In Eq. (9), the iterations n and N stand for the exact and highest iterations. The remaining (or followers) salps' location is determined by the formula below:

$$x_j^i = \frac{x_j^i + x_j^{i-1}}{2} \tag{10}$$

In Algorithm 2, the fundamental SSA pseudo code is described.

1.	Create the salp population $x_i(i = 1, 2, \dots, n)$, bringing U_b and L_b into consideration
2.	while
3.	Calculate every search agent's suitability value (salp)
4.	F is the top search factor
5.	By Eq. (9), update c_1
6.	if ($i == 1$), update c_1 using Eq. (9) for every salp (x_i)
7.	Eq. (8) update the leader location
8.	else
9.	Refresh the followers' locations using Eq. (10)
10.	end
11.	end
12.	Salps should be modified according to $[U_b L_b]$
13.	end
14.	Go back to F

Algorithm 2. SSA's Pseudocode

Support vector regression (SVR)

The statistical learning theory and structural risk minimization concept are the foundations of the nonlinear regression method known as SVR [53]. The core of this method is a nonlinear conversion technique (kernel functions) that converts the initial input area into a novel hyperspace. The complex and nonlinear interactions among input and outcome parameters in transformed hyperspace are described by a linear function [53, 54]. SVR is the procedure of identifying the function $f(x)$ that is as flat as feasible and has a maximum deflection ε from the training samples (x_i, y_i) for $i = 1, \dots, N$. The complexity of the model may be reduced by maximizing the function’s flatness, which affects the model’s overall performance. In fact, according to the learning theory [55], the generalization error may be limited by the sum of two factors, one of which depends on the model’s complexity and the other of which depends on the fault in the training data. The foundation of SVR techniques is the management of model complexity throughout training.

The procedure is initially explained for a linear function $f(x)$ of the following type:

$$f(x) = w \cdot x + b \tag{11}$$

Based on this equation, x, w , and b show the input vector, the parameters’ vector (or weight), and a constant to be specified, respectively. The data is mapped onto a greater dimensional area utilizing a nonlinear kernel in the context of nonlinear issues:

$$f(x) = w\varphi(x) + b \tag{12}$$

In this equation, the kernel function is shown by $\varphi(x)$.

Data may be mapped onto a greater dimensional characteristic area and can also be used to apply a linear regression method. The w and b may be gained by minimizing the given function:

$$\min \frac{1}{2} \|w\|^2 + c \sum_{i=1}^N (\xi_i + \xi_i^*) \tag{13}$$

Subjected to:

$$\begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \text{with } \xi_i, \xi_i^* \geq 0, i = 1, \dots, N \end{cases}$$

Based on this equation, ξ_i^* and ξ_i show the negative and positive faults as presented in Fig. 4. A hyperparameter allowing for adjustment of the balance among the amount of fault permitted and the flatness of the function $f(x)$ is the constant $C > 0$. The penalty factor C and the kernel function variable (σ) control the generalization and fitting capabilities of the SVR model, respectively. Selecting the optimal C and σ may assist in avoiding under- or over-fitting while also improving the SVR model’s estimating efficiency.

Metrics

The formulations in Eqs. 14–17 depict the metrics values for assessing the models’ robustness and comparing with the literature to choose the outperformed model.

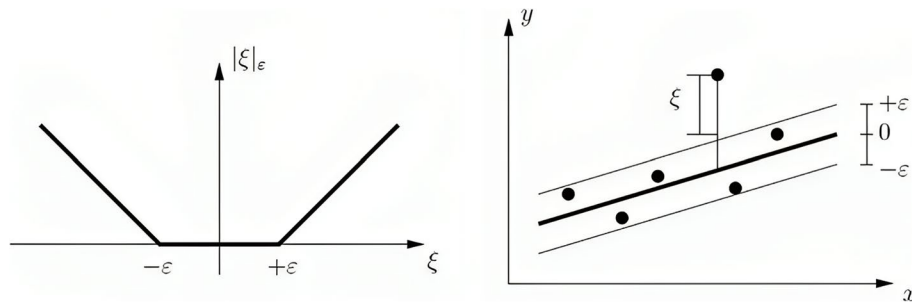


Fig. 4 Setting the soft margin loss for a linear SVR [56]

1) Coefficient of determination (R^2)

$$R^2 = \left(\frac{\sum_{d=1}^D (m_d - \bar{m})(z_d - \bar{z})}{\sqrt{\left[\sum_{d=1}^D (m_d - \bar{m})^2 \right] \left[\sum_{d=1}^D (z_d - \bar{z})^2 \right]}} \right)^2 \tag{14}$$

2) Root-mean-square error ($RMSE$)

$$RMSE = \sqrt{\frac{1}{D} \sum_{d=1}^D (z_d - m_d)^2} \tag{15}$$

3) Mean absolute error (MAE)

$$MAE = \frac{1}{D} \sum_{d=1}^D |z_d - m_d| \tag{16}$$

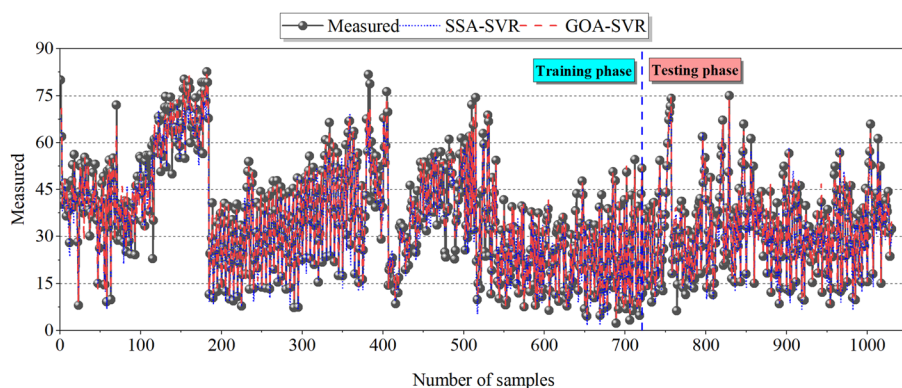
4) $A_{20-Index}$

$$A_{20-index} = \frac{m_{20}}{M} \tag{17}$$

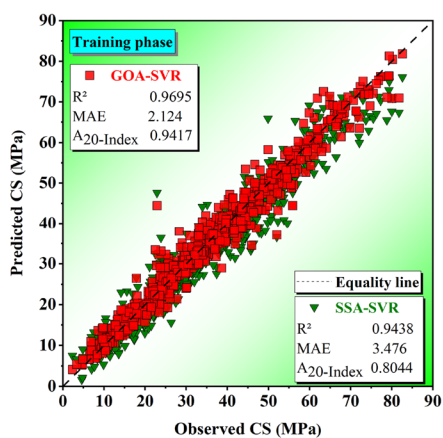
In these equations: m_d defines the records, \bar{m} the mean values of records, z_d the predicted values, \bar{z} the mean values of predictions, and D is the number of a dataset. M is the number of samples, and m_{10} is the number of samples with a recorded/predicted proportion in [0.8, 1.2].

Results and discussion

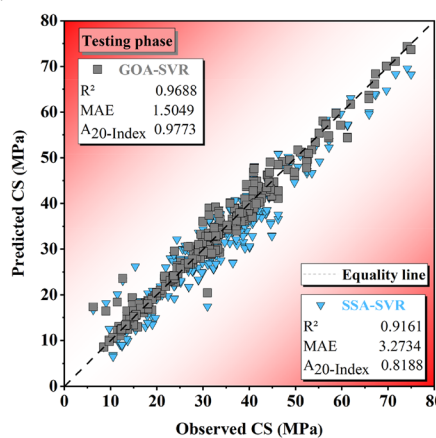
The conclusions of the GOA-SVR and SSA-SVR architectures are provided to forecast the CS of the HPC augmented with BFS and FA. As stated in the previous part, the productivity of SVR is determined by choosing the proper values for the mixture of the key SVR sections. The reported and computed values of the CS of HPC throughout the training and testing stages for the GOA-SVR and SSA-SVR simulations are presented in Fig. 5. Along with the time series plots, a residual CS concentration graph with a normally distributed curves around the zero line is given. R^2 , RMSE, MAE, and $A_{20-Index}$ were calculated to assess the workability of the GOA-SVR and SSA-SVR (see Table 2). Both the GOA-SVR and SSA-SVR techniques show great potential for properly forecasting HPC CS.



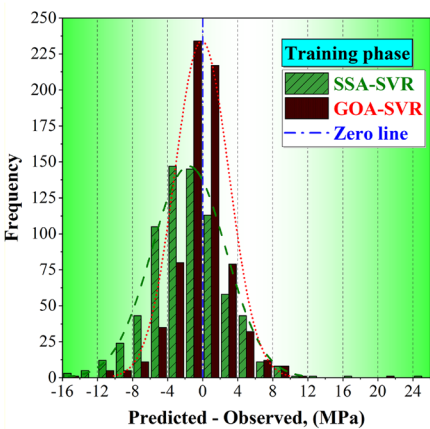
(a)



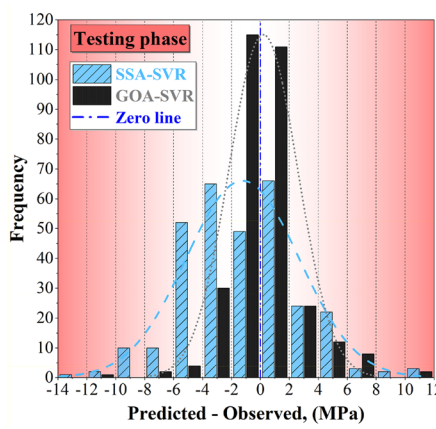
(b)



(c)



(d)



(e)

Fig. 5 Results from the SVR models

This section of the research compares the findings of the statistical identifiers developed for the generated investigations, known as GOA-SVR and SSA-SVR, to see whether one version performs better than others. Furthermore, an attempt has been undertaken to appropriately compare the investigation's findings to previously published ones. The findings reveal that the combined GOA-SVR and SSA-SVR frameworks could perform wonderfully well during predicting, with R^2 values of 0.9695 and

Table 2 Findings of developed SVR s and comparison with previous articles

Phase	Metric	Present	Present	Comparison purposes								
						[38]	[28]	[39]	[19]	[40]	[41]	[42]
				GOA-SVR	SSA-SVR	GEP	Multi-GGP	GPR	ANNs	SEM	GPR	XGBoost (all data)
Train	R^2	0.9695	0.9438	0.8224	0.7885	0.9877	—	0.84	0.888	0.944		
	RMSE	3.1008	4.5016	—	7.36	—	—	6.3	5.59	3.878		
	MAE	2.124	3.476	5.202	5.56	—	—	4.91	3.996	2.592		
	$A_{20-Index}$	0.9417	0.8044	—	—	0.9753	—	0.68	—	—		
Test	R^2	0.9688	0.9161	0.8354	0.8046	0.8858	0.8649	0.8567	0.888	—		
	RMSE	2.3906	4.0861	—	7.31	—	6.329	5.968	5.597	—		
	MAE	1.5049	3.2734	5.19	5.48	—	4.421	4.482	3.913	—		
	$A_{20-Index}$	0.9773	0.8188	—	—	0.757	—	0.752	—	—		

0.9688 for GOA-SVR and 0.9438 and 0.9161 for SSA-SVR for the training and testing sections, respectively. However, analyzing and evaluating the generated signals is necessary to choose the appropriate technique. When compared to the SSA-SVR, the GOA-SVR RMSE’s value decreased somewhat in the training portion from 3.6962 to 2.4815 MPa. A stunning drop from 4.0861 to 2.3906 MPa was achieved in the testing stage. Another criterion, MAE, yielded similar findings to RMSE and shows that the GOA-SVR has a better ability for CS prediction at $MAE_{Train}= 2.124$ and $MAE_{Test}= 1.5049$ because its values were smaller than those of the SSA-SVR at $MAE_{Train}= 3.476$ and $MAE_{Test}= 3.2734$. A similar pattern might be found in the $A_{20-Index}$ indicator, which showed a 14 percent increase in the train portion and a 16 percent increase in the test section for GOA-SVR.

The computations in this article were compared with [19, 28, 38–42] in order to give a full review and validation. Where different types of techniques used like Gene Expression Programming (GEP) [38], Semi-empirical method (SEM) [40], Gaussian process regression (GPR) [41, 39], Extreme-gradient-boosting (XGBoost) [42], Artificial neural networks (ANNs) [19], and Multi-gene genetic programming (Multi-GGP) [28]. As shown in Table [38], for example, exhibited a little lower R^2 and a much higher MAE than GOA-SVR by 0.8224 and 5.202, respectively. SEM [40], as a new technique, performed worse than GOA-SVR, with R^2 at 0.84 significantly lower than 0.9695, RMSE at 6.3 lower than 3.1108 (a drop of greater than 50%), MAE at 4.91 greater than 2.124, and $A_{20-Index}$ at 0.68 lower than 0.9417. Moreover, the GOA-SVR beat the GPR [41] watching to R^2 , RMSE, and MAE. Another GPR [39] depicted marginally better in the training phase compared to this article, but it was terribly poor in the testing section. Other approaches, such as Multi-GGP [28] and ANNs [19] performed weakly than GOA-SVR, with R^2 values of 0.8046 and 0.8469, respectively, which are interestingly less than 0.9688. Finally, the most recently developed technology, XGBoost [42], got close, although it was also weaker than GOA-SVR. Finally, the suggested model is the GOA-SVR framework created for simulating the CS of HPC with FA and BFS.

Conclusions

The main objective of the current research is to supply a practical way to evaluate the effectiveness of Intelligent machines in calculating high-performance concrete (HPC)'s compressive strength (CS) via stringent testing. Using the support vector regression (SVR) technique, we attempted to construct models that might predict HPC's qualities while it was fresh and hardened properties. It used the grasshoppers' optimization algorithm (GOA) and salp swarm optimization (SSA) algorithms in this work to pinpoint the most critical components of SVR that need to be changed. The created techniques were assessed by utilizing 1030 experiments, eight input parameters, like admixtures, concrete age, and the major constituent of mixes, as well as the CS as the forecasting objective. The findings were then compared to other studies in the field.

- The findings reveal that the combined GOA-SVR and SSA-SVR frameworks could perform wonderfully well during predicting, with R^2 values of 0.9695 and 0.9688 for GOA-SVR and 0.9438 and 0.9161 for SSA-SVR for the train and test sections, respectively.
- When compared to the SSA-SVR, the GOA-SVR RMSE's value decreased somewhat in the training portion from 3.6962 M to 2.4815 MPa. A stunning drop from 4.0861 to 2.3906 MPa was achieved in the testing stage. Another criterion, MAE, yielded similar findings to RMSE and shows that the GOA-SVR has a better ability for CS prediction at $MAE_{Train} = 2.124$ and $MAE_{Test} = 1.5049$ because its values were smaller than those of the SSA-SVR at $MAE_{Train} = 3.476$ and $MAE_{Test} = 3.2734$. A similar pattern might be found in the $A_{20-Index}$ indicator, which showed a 14 percent increase in the train portion and a 16 percent increase in the test section for GOA-SVR.
- As shown, the suggested GOA-SVR performed the best compared to the literature. GEP [38] exhibited lower R^2 and a much higher MAE than GOA-SVR by 0.8224 and 5.202, respectively. SEM [40] performed worse than GOA-SVR, with R^2 at 0.84, significantly lower than 0.9695, RMSE with a drop of greater than 50%, and $A_{20-Index}$ at 0.68 lower than 0.9417. Moreover, the GOA-SVR beat the GPR [41], watching to R^2 , RMSE, and MAE. Other approaches, such as ANNs [19] and Multi-GGP [28] performed weakly than GOA-SVR, with R^2 values of 0.8469 and 0.8046, respectively, which are interestingly less than 0.9688. Finally, the most recently developed technology, XGBoost [42], got close, although it was also weaker than GOA-SVR.
- Finally, the suggested model is the GOA-SVR framework created for simulating the CS of HPC updated with FA and BFS.

Abbreviations

HPC	High-performance concrete
CS	Compressive strength
ANN	Artificial neural networks
SVR	Support vector regression
GOA	Grasshoppers' optimization algorithm
ML	Machine learning
SSA	Salp swarm algorithm
RMSE	Root mean square error

MAE	Mean absolute error
SP	Superplasticizer
R ²	Coefficient of correlation
AC	Age of concrete
FA	Fly ash
FA	Fine aggregate
BFS	Blast furnace slag
W	Water
C	Cement
Ca	Coarse aggregate

Acknowledgements

I would like to take this opportunity to acknowledge that there are no individuals or organizations that require acknowledgment for their contributions to this work.

Authors' contributions

LJ activated in methodology, software, validation, and formal analysis section. WJ performed formal analysis, methodology, validation, and language review. YS handled writing—original draft preparation, conceptualization, supervision, and project administration. All authors have read and approved the manuscript.

Funding

No funding was obtained for this study.

Availability of data and material

The authors do not have permissions to share data.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 24 July 2023 Accepted: 9 November 2023

Published online: 21 December 2023

References

1. Cook RA, Goodspeed C, Vanicar S (1998) High-Performance Concrete Defined for Highway Structures. Federal Highway Administration, United States
2. Mousavi SM, Gandomi AH, Alavi AH et al (2010) Modeling of compressive strength of HPC mixes using a combined algorithm of genetic programming and orthogonal least squares. *Structural engineering and mechanics* 36:225–241
3. Domone PLJ, Soutsos MN (1994) Approach to the proportioning of high-strength concrete mixes. *Concr Int* 16:26–31
4. Yeh I-C (1998) Modeling of strength of high-performance concrete using artificial neural networks. *Cem Concr Res* 28:1797–1808
5. Sarkhani Benemaran R, Esmaili-Falak M, Javadi A. Predicting resilient modulus of flexible pavement foundation using extreme gradient boosting based optimised models. *International Journal of Pavement Engineering*. 2022;1–20
6. Benemaran RS (2023) Application of extreme gradient boosting method for evaluating the properties of episodic failure of borehole breakout. *Geoenergy Science and Engineering* 226:211837
7. Esmaili-Falak M, Benemaran RS (2023) Ensemble deep learning-based models to predict the resilient modulus of modified base materials subjected to wet-dry cycles. *Geomechanics and Engineering* 401:132833
8. Sarkhani Benemaran R, Esmaili-Falak M (2023) Predicting the Young's modulus of frozen sand using machine learning approaches: State-of-the-art review. *Geomechanics and Engineering* 34:507–527
9. Basma AA, Barakat SA, Al-Oraimi S (1999) Prediction of cement degree of hydration using artificial neural networks. *ACI Mater J* 96:167–172
10. Ji T, Lin T, Lin X (2006) A concrete mix proportion design algorithm based on artificial neural networks. *Cem Concr Res* 36:1399–1408
11. Lee S-C (2003) Prediction of concrete strength using artificial neural networks. *Eng Struct* 25:849–857
12. Yeh I-C (2007) Modeling slump flow of concrete using second-order regressions and artificial neural networks. *Cem Concr Compos* 29:474–480
13. Kasperkiewicz J, Racz J, Dubrawski A (1995) HPC strength prediction using artificial neural network. *J Comput Civ Eng* 9:279–284
14. Prasad BKR, Eskandari H, Reddy BVV (2009) Prediction of compressive strength of SCC and HPC with high volume fly ash using ANN. *Constr Build Mater* 23:117–128
15. Rajasekaran S, Amalraj R (2002) Predictions of design parameters in civil engineering problems using SLNN with a single hidden RBF neuron. *Comput Struct* 80:2495–2505
16. Rajasekaran S, Suresh D, Vijayalakshmi Pai GA (2002) Application of sequential learning neural networks to civil engineering modeling problems. *Eng Comput* 18:138–147

17. Rajasekaran S, Lavanya S (2007) Hybridization of genetic algorithm with immune system for optimization problems in structural engineering. *Struct Multidiscip Optim* 34:415–429
18. Erdal HI, Karakurt O, Namlı E (2013) High performance concrete compressive strength forecasting using ensemble models based on discrete wavelet transform. *Eng Appl Artif Intell* 26:1246–1254
19. Chou J-S, Pham A-D (2013) Enhanced artificial intelligence for ensemble approach to predicting high performance concrete compressive strength. *Constr Build Mater* 49:554–563
20. Erdal HI (2013) Two-level and hybrid ensembles of decision trees for high performance concrete compressive strength prediction. *Eng Appl Artif Intell* 26:1689–1697
21. Cheng M-Y, Firdausi PM, Prayogo D (2014) High-performance concrete compressive strength prediction using Genetic Weighted Pyramid Operation Tree (GWPOT). *Eng Appl Artif Intell* 29:104–113
22. Kaloop MR, Kumar D, Samui P et al (2020) Compressive strength prediction of high-performance concrete using gradient tree boosting machine. *Constr Build Mater* 264:120198
23. Asteris PG, Roussis PC, Douvika MG (2017) Feed-forward neural network prediction of the mechanical properties of sandcrete materials. *Sensors* 17:1344
24. Rafiei MH, Khushefati WH, Demirboga R et al (2017) Supervised deep restricted Boltzmann machine for estimation of concrete. *ACI Mater J* 114:237
25. Nguyen T, Kashani A, Ngo T et al (2019) Deep neural network with high-order neuron for the prediction of foamed concrete strength. *Computer-Aided Civil and Infrastructure Engineering* 34:316–332
26. Rafiei MH, Khushefati WH, Demirboga R, et al (2017) Novel Approach for Concrete Mixture Design Using Neural Dynamics Model and Virtual Lab Concept. *ACI Mater J* 114:117–127
27. Angeline PJ (1992) Genetic programming: On the programming of computers by means of natural selection: John R. A Bradford Book, MIT Press, Cambridge MA, Koza (0-262-11170-5, xiv+ 819 pp., US \$55.00. Elsevier; 1994)
28. Gandomi AH, Alavi AH (2012) A new multi-gene genetic programming approach to nonlinear system modeling. Part I: materials and structural engineering problems. *Neural Comput Appl*. 21:171–187
29. Ferreira C (2001) Gene expression programming: a new adaptive algorithm for solving problems. *arXiv preprint cs/0102027* 13(2):87–129
30. Alavi AH, Gandomi AH (2011) A robust data mining approach for formulation of geotechnical engineering systems. *Eng Comput (Swansea)* 28(3):242–74
31. Gandomi AH, Alavi AH, Mirzahosseini MR et al (2011) Nonlinear genetic-based models for prediction of flow number of asphalt mixtures. *J Mater Civ Eng* 23:248–263
32. Baykasoğlu A, Güllü H, Çanakçı H et al (2008) Prediction of compressive and tensile strength of limestone via genetic programming. *Expert Syst Appl* 35:111–123
33. Cevik A, Cabalar AF (2009) Modelling damping ratio and shear modulus of sand–mica mixtures using genetic programming. *Expert Syst Appl* 36:7749–7757
34. Niu Z, Yuan Y, Sun J (2023) Neuro-fuzzy system development to estimate the compressive strength of improved high-performance concrete. *Multiscale and Multidisciplinary Modeling, Experiments and Design* 1–15
35. Islam N, Kashem A, Das P, et al (2023) Prediction of high-performance concrete compressive strength using deep learning techniques. *Asian Journal of Civil Engineering* 1–15
36. Imran M, Khushnood RA, Fawad M (2023) A hybrid data-driven and metaheuristic optimization approach for the compressive strength prediction of high-performance concrete. *Case Studies in Construction Materials* 18:e01890
37. Khan MI, Abbas YM (2023) Robust extreme gradient boosting regression model for compressive strength prediction of blast furnace slag and fly ash concrete. *Mater Today Commun* 35:105793
38. Mousavi SM, Aminian P, Gandomi AH et al (2012) A new predictive model for compressive strength of HPC using gene expression programming. *Adv Eng Softw* 45:105–114
39. Asteris PG, Skentou AD, Bardhan A et al (2021) Predicting concrete compressive strength using hybrid ensembling of surrogate machine learning models. *Cem Concr Res* 145:106449
40. Nguyen N-H, Vo TP, Lee S et al (2021) Heuristic algorithm-based semi-empirical formulas for estimating the compressive strength of the normal and high performance concrete. *Constr Build Mater* 304:124467
41. Van DD, Adeli H, Ly H-B et al (2020) A sensitivity and robustness analysis of GPR and ANN for high-performance concrete compressive strength prediction using a Monte Carlo simulation. *Sustainability* 12:830
42. Lee S, Nguyen N, Karamanli A, et al (2022) Super learner machine-learning algorithms for compressive strength prediction of high performance concrete. *Structural Concrete* 24(2):2208–2228.
43. Yeh I-C (1999) Design of high-performance concrete mixture using neural networks and nonlinear programming. *J Comput Civ Eng* 13:36–42
44. Yeh I-C (2003) Prediction of strength of fly ash and slag concrete by the use of artificial neural networks. *J Chin Inst Civil Hydraul Eng* 15:659–663
45. Yeh I-C (2006) Analysis of strength of concrete using design of experiments and neural networks. *J Mater Civ Eng* 18:597–604
46. Yeh I-C (1998) Modeling Concrete Strength with Augment-Neuron Networks. *J Mater Civ Eng* 10:263–268
47. Leema N, Nehemiah HK, Kannan A (2016) Neural network classifier optimization using differential evolution with global information and back propagation algorithm for clinical datasets. *Appl Soft Comput* 49:834–844
48. Khorsheed MS, Al-Thubaity AO (2013) Comparative evaluation of text classification techniques using a large diverse Arabic dataset. *Lang Resour Eval* 47:513–538
49. Shi X, Yu X, Esmaili-Falak M (2023) Improved arithmetic optimization algorithm and its application to carbon fiber reinforced polymer-steel bond strength estimation. *Compos Struct* 306:116599
50. Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. *Adv Eng Softw* 105:30–47
51. Aljarah I, Al-Zoubi A, Faris H et al (2018) Simultaneous feature selection and support vector machine optimization using the grasshopper optimization algorithm. *Cognit Comput* 10:478–495
52. Mirjalili S, Gandomi AH, Mirjalili SZ et al (2017) Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv Eng Softw* 114:163–191

53. Smola AJ, Schölkopf B (2004) A tutorial on support vector regression, *statist. Comput* 14:199–222
54. Masoumi F, Najjar-Ghabel S, Safarzadeh A et al (2020) Automatic calibration of the groundwater simulation model with high parameter dimensionality using sequential uncertainty fitting approach. *Water Supply* 20:3487–3501
55. Aghayari Hir M, Zaheri M, Rahimzadeh N (2022) Prediction of Rural Travel Demand by Spatial Regression and Artificial Neural Network Methods (Tabriz County). *Journal of Transportation Research* 20(4):367–386
56. A. M. Andrew (2000) *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods* by Nello Christianini and John Shawe-Taylor, Cambridge University Press, Cambridge 18(6):687–689

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)
