Predicting California Bearing Ratio Using Hybrid Least Square Support Vector Regression with IAOA, ESMA, and RKO Meta-Heuristic Algorithms

Jingzhe Li[1, *] and Lan Meng[2]
[1]Department of Basic, Shijiazhuang Institute of Technology, Shijiazhuang, Hebei, 050000, China
[2]China Railway Jingcheng Engineering Testing Co., Ltd, Beijing, 100000, China
E-mail: lijingzhe0617@163.com
[*]Corresponding author

Keywords: california bearing ratio, least square support vector regression, improved arithmetic optimization algorithm, equilibrium slime mould algorithm, runge kutta optimization

Received: November 30, 2024

The California Bearing Ratio (CBR) test is a crucial geotechnical parameter for evaluating soil strength. This study proposes a Least Squares Support Vector Regression (LSSVR) model to predict CBR values using compaction characteristics, moisture content, and soil properties. A dataset comprising 110 soil samples was used, with 70% for training and 30% for testing. To enhance predictive accuracy, three meta-heuristic algorithms—Improved Arithmetic Optimization Algorithm (IAOA), Equilibrium Slime Mould Algorithm (ESMA), and Runge Kutta Optimization (RKO)—were integrated with LSSVR, forming hybrid models LSIA, LSEM, and LSRK. These algorithms optimized the regularization parameter (C) and kernel parameter (Gamma) to improve model generalization. Performance evaluation using R^2, RMSE, and MAE showed that the LSIA model outperformed all others, achieving an R^2 of 0.9975 (training) and 0.9932 (testing), along with the lowest RMSE (0.5489) and MAE (0.3176). The results confirm that LSIA exhibits superior predictive accuracy and robustness, making it a reliable and time-efficient alternative for geotechnical applications.

Povzetek:

1 Introduction

The California Bearing Ratio (CBR) is a widely used parameter in engineering to evaluate the mechanical properties of soil subgrade, which are crucial for designing flexible pavements. Assessing the CBR value is vital for determining the suitable thickness of the roadway layer that can handle the expected traffic volume [1]–[3].

In particular, when the CBR value rises, increasing the pavement layer's thickness is essential to provide enough resistance against the anticipated load. By empowering engineers to make informed decisions on the use of materials, structural design, and building techniques, researchers help build structures that are long-lasting, sustainable, and CBR. A $50\ mm$ diameter circular plunger is used to pierce a soil mass; the pressure per unit area needed is measured and separated by the comparable diffusion in a reference material at a continuous rate of $1.25\ mm$ each minute. This metric is known as the CBR. A ratio of 2.5 to $5\ mm$ is usually used to evaluate penetrations; greater results are obtained with the $5\ mm$ ratio than with the $2\ mm$ ratio [4], [5]. Compacted samples that have been undisturbed or reconstituted can be used in laboratory studies in either a saturated or unsaturated water environment. It is important to remember that current studies have shown that the condition of the material during testing can have a big impact on the CBR values [6]–[8]. CBR is a fundamental parameter in geotechnical engineering used to assess soil strength. Using a standard plunger, the CBR value is obtained by measuring the pressure obligatory to penetrate a soil sample. The main ingredients of CBR include the soil type, moisture content, and compaction level. These factors significantly impact the CBR value and are crucial in accurately predicting soil strength. Understanding the ingredients of CBR is essential in designing and constructing reliable and robust structures that can withstand various loads and conditions [9], [10].

The OMC represents the moisture content at which the soil is most compact and has the greatest strength. In the CBR test, the optimum moisture content (OMC) is the moisture content at which the soil sample has the highest CBR value. Several CBR tests are applied on soil samples with varying moisture contents to determine the OMC. The CBR value is measured for each moisture content, and the results are plotted on a graph with the CBR value on the $y-axis$ and moisture content on the $x-axis$. The moisture content at which the CBR value is the highest is the OMC [7]. The OMC is an important parameter in designing and constructing roads, pavements, and other infrastructure. If the soil is too dry, it may be hard and brittle, leading to cracking and other damage. If the soil is wet, it may be too soft and weak, leading to deformation and failure under load. Therefore, it is important to determine the OMC to ensure that the soil has the required strength and stability for the intended use [11].

However, when done in a laboratory setting, calculating the CBR value may be difficult and costly. As a result, researchers have explored alternative methods, such as utilizing machine learning (ML) models, to predict the CBR value of soil subgrades. ML, a subfield of artificial intelligence [12], focuses on creating algorithms that have the capacity to learn from data and become more effective over time. This study area has clear benefits for effectively handling sizable and intricate information, making it possible to spot underlying patterns and produce accurate forecasts. Several approaches have been thoroughly examined in ML, such as reinforcement learning, supervised learning, deep learning, unverified learning, and others. ML applications may be found in a wide range of industries, including manufacturing, transportation, finance, and engineering [13]–[17].

Table 1 presents the summary of some previous studies related to CBR prediction.

Table 1: Summary of previous studies related to CBR prediction.

Article	Model	Evaluator	
		R^2	RMSE
[18]	MLR	0.841	-
[19]	ANN	0.981	1.187
[20]	ANN	0.945	2.81
[21]	GPR	0.999	0.139
[22]	ANN	0.970	1.691
[23]	MARS-L	0.969	0.036
[24]	RF	0.98	1.43
[25]	LGBM	0.947	0.33

In this research, ML methodologies are employed to address complex systems and analyze multiple variables to estimate the CBR measure. The study assumes a methodological method centered around utilizing the LSSVR technique. Accurate forecast models for CBR are of utmost importance. However, optimizing the parameters of LSSVR is crucial to enhance its predictive performance. To tackle this issue, the paper explores the integration of three meta-heuristic algorithms: the Improved Arithmetic optimization algorithm ($IAOA$), the Equilibrium Slime Mould Algorithm (ESMA), and Runge Kutta Optimization (RKO). By streamlining the CBR −related design and construction procedures, this study emphasizes the positive impact that can have on the substructure sector and the built situation. To evaluate the proposed framework, the study collects a dataset of CBR and conducts comparative analyses to assess the effectiveness of this approach compared to traditional optimization methods.

The choice of IAOA, ESMA, and RKO for optimizing LSSVR in CBR prediction was guided by their unique computational advantages in balancing exploration and exploitation, ensuring efficient hyperparameter tuning. IAOA enhances global search capability while mitigating premature convergence, making it ideal for optimizing non-linear geotechnical data. Inspired by slime mould foraging behaviour, ESMA introduces an equilibrium-based update mechanism that improves search diversity and robustness. RKO, rooted in the Runge-Kutta numerical method, provides a structured and stable optimization process, ensuring precise parameter tuning.

2 Materials and methodology

2.1 Data gathering

As shown in Table 2, the experimental samples comprising the CBR dataset were split into two sets: a training phase comprising 76 samples (70%) and a test phase comprising 33 samples (30%). Based on these predictor factors, an $LSSVR$ model was used to predict the CBR. The dataset utilized in this investigation came from an investigation carried out by Ikeagwuani [26]. It uses a combination ratio obtained using the response surface and Taguchi array experimental techniques. The stabilizers and modified extensive soil qualities shown in Table 2 are linked to the output variable, CBR value, and have inherent geotechnical characteristics. Compaction traits and atterberg limits, such as liquid limit (LL), plasticity index (PI), plastic limit (PL), OMC, and maximum dry density (MDD), are among the physical features of the changed soil. The input variables are the stabilizers, namely SDA, OPC, and QD.

Table 2: The statistical properties of model input and target values.

Indicator	Statistical Properties								
	LL	PL	PI	MDD	OMC	SDA	QD	OPC	CBR
Max	52.1	37.2	19.5	1.777	29.5	20	20	8	66.75
Min	21.2	17.9	2.1	1.365	18.9	0	0	2	19.69
Avg	35.85	26.68	9.162	1.493	24.14	10.66	10.64	4.945	39.959
St. Dev.	6.154	4.281	4.115	0.088	2.427	7.155	8.196	2.380	10.866

2.2 Least square support vector regression (LSSVR)

As a development to SVR that can handle issues with function estimation, regression, and classification, Vandewalle and Suykens introduced $LSSVR$ [27]. Their articles contained documentation of this. SVR was developed as a supervised ML approach in 1995 by Vapnik and his coworkers. However, since LSSVR employs linear equations rather than quadratic ones, which simplifies the optimization process, it has an advantage over SVR [28]–[31]. The $LSSVR$ process, showed in Fig. 1, includes utilizing the input x_i and the output y_i time series to direct the $LSSVR$ function as presented below.

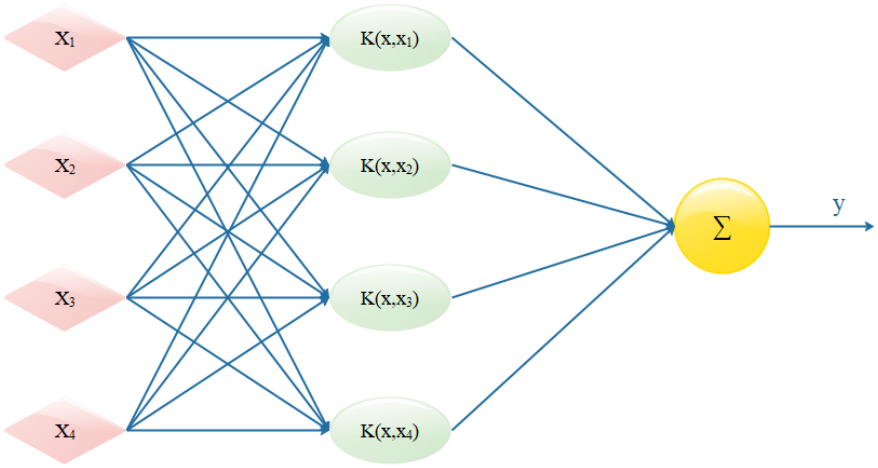


Figure 1: The $LSSVR$ model is used to forecast wind power and speed.

$$y(x) = \varpi^M \psi(x) + d \tag{1}$$

In Eq. (1), x can assumed that an input and the output (y) is resolute by the weight vector (ϖ), which has min extents, the partiality term d and the charting term ψ [32]. The following may be used to exemplify the cost function for $LSSVR$:

$$minJ(\varpi, a) = \frac{1}{2}\varpi^M \varpi + \frac{\tau}{2}\sum_{i=1}^{N} a_i^2 \tag{2}$$

which is limited by the following:

$$y_i = \varpi^M \psi(x_i) + d + a_i, \quad (i = 1,2,3, \dots, N) \tag{3}$$

For a given input x_i, the values of the training error and the regularization constant are provided by a_i and τ, respectively. The Lagrange multiplier (LM) optimum programming technique is used for solving Eq. (2). The constraint problem needs to be changed into a non-constraint problem to use this method to derive the objective function [33], [34]. The Lagrange function (L) can be intended in the following:

$$L(\varpi, d, a, \delta) = J(\varpi, a) - \sum_{i=1}^{N} \delta_i \{\varpi^M \psi(x_i) + d + a_i - y_i\} \tag{4}$$

Here δ_i is the LM.

$Karush - Cohen - Tucker$ [35], using the partial derivative calculations of Eq. (4) about $\omega, b, e,$ and β independently, as found:

$$\begin{cases} \varpi = \sum_{i=1}^{N} \delta_i \psi(x_i) + d \\ \sum_{i=1}^{N} \delta_i = 0 \\ \delta_i = \tau^{ai} \\ \varpi^M \psi(x_i) + d + a_i - y_i = 0 \end{cases} \tag{5}$$

It is possible to derive the linear equations by removing the variables from a_i and ϖ can be stated as:

$$\begin{pmatrix} 0 & -G^M \\ G & \Omega + \tau^{-1} G \end{pmatrix} \begin{pmatrix} d \\ \beta \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{6}$$

Here $G = (y_1, \dots, y_N), \Omega = (\psi(x_1)^M y_1, \dots, \psi(x_N)^M y_N, P = (1, \dots, 1), \beta = (\beta_1, \dots, \beta_1)$.

Under $Mercer's$ standard (to get more explanations of $Mercer's$ standard, refer to [36]), it can be mentioned that the kernel is articulated as $s(x, x_i) = f(x)^M f(x_i), i = 1,2,3, \dots, H$. Consequently, the $LSSVR$ can be formulated as follows:

$$f(x) = \sum_{i=1}^{N} \delta_i s(x, x_i) + d. \tag{7}$$

$$s(x, x_i) = exp\left(-\frac{\|x - x_i\|^2}{2\varrho^2}\right). \tag{8}$$

2.3 Equilibrium slime mould algorithm (ESMA)

Slime mold's (SM) foraging habits offer a potential basis of inspiration to create successful and efficient optimization techniques [37]. The ESMA algorithm begins by initializing each slime mold's position using a randomization procedure, ensuring a diverse search space exploration. This is mathematically represented by Eq. (9). However, beyond initialization, ESMA primarily mimics the dynamic foraging behavior of slime molds, where individuals adjust their positions based on the concentration of nutrients in their surroundings. This bio-inspired movement strategy allows the algorithm to balance exploration and exploitation, enhancing optimization efficiency.

$$\vec{X_i}(t = 1) = r_1.(UB - LB) + LB, \quad i = 1,2, \dots, N. \tag{9}$$

Here, UB and LB indicate the upper and lower bound. The aligning model for the $i - th\ SM$, indicated as X_i $(j = 1,2, \dots, N)$, in the ensuing iteration $(t + 1)$, is recognized using SMA as follows:

$$\vec{\mathrm{X}}_i(t+1)=\begin{cases} r_1.(UB-LB)+LB, & r_1<z \\ \vec{\mathrm{X}}_{Gbest}+\overrightarrow{step}_a.\left(\vec{\mathrm{U}}.\vec{\mathrm{X}}_C-\vec{\mathrm{X}}_D\right), & r_2<P_i\ (t)\ \ and\ r_1\geq z \\ \overrightarrow{step}_b\ .\ \vec{\mathrm{X}}_i(t), & r_2\geq P_i\ (t)\ \ and\ r_1\geq z \end{cases} \tag{10}$$

The $\vec{\mathrm{X}}_{Gbest}$ represents the global best fitness value attained throughout iterations one through t. Moreover, the variables r_1 and r_2 resemble to change values within the choice of $[0,1]$.

To eliminate and distribute the SM, a chance signified by z is used. Inside the background of this paper, z is a continuous worth of 0.03 [38]. Eq. (11) is applied to type the suitability values in climbing instruction.

$$[sortf, sortIndex]=sort(f),\ \ where\ f=\{f_1,f_2,\ldots..,f_N\} \tag{11}$$

In Eq. (11) sorts the elements of $f=\{f1,f2,\ldots,fN\}$ in ascending order, producing $sortf$, the sorted version of f. The corresponding indices of the original elements are stored in $sortIndex$. Here, $sort()$ denotes the sorting operation, which arranges elements in increasing order unless otherwise specified.

Eq. (12) is working to calculate $\vec{\mathrm{U}}$.

$$\vec{\mathrm{U}}(sortIndex(j))=\begin{cases} 1+r_3\ .\log\left(\frac{f_{Lbest}-sortf\ (j)}{f_{Lbest}-f_{Lworst}}+1\right) & 1\leq j\leq \frac{N}{2} \\ 1-r_3\ .\log\left(\frac{f_{Lbest}-sortf\ (j)}{f_{Lbest}-f_{Lworst}}+1\right) & \frac{N}{2}<j\leq N \end{cases} \tag{12}$$

Eq. (12) calculates $\vec{\mathrm{U}}$, which represents (explain its purpose, e.g., an updated search variable). The equation follows a piecewise definition, where for $j\leq 2$, [explain the significance], and for $j>2$, [explain the significance]. A random number, r_3 is equally dispersed over the [0,1] range. The values obtained for the resident worst and greatest fitness for the present iteration are indicated by f_{Lworst} and f_{Lbest}, correspondingly. Eqs. (13-14) are working to compute these fitness values.

$$f_{Lbest}=sortf(1) \tag{13}$$

$$f_{Lworst}=sortf(N) \tag{14}$$

Eqs. (13) and (14) utilize f_{Lworst} and f_{Lbest}, which represent the worst and best fitness values within the current iteration. These values are obtained by (explain the method, sorting the fitness values from Eq. (11)). To maintain consistency with Eq. (12), we define these as (choose a single consistent term, 'present iteration worst and best fitness values') throughout the text. Under is the formulation that describes the inconstant P_i, which signifies the likelihood of picking the course of $i-th\ SM$:

$$P_i=\tanh|f(X_i)-f_{Gbest}| \tag{15}$$

Eq. (15) defines $P_i=\tanh|f(X_i)-f_{Gbest}|$, where f_{Gbest} represents the best fitness value obtained from the first iteration up to the current iteration. This ensures that P_i reflects the difference between the current solution's fitness and the globally best fitness found so far. For apiece $i=1,2,\ldots,N$, the suitability rate of the $i-th\ SM$ in X_i is resolute by $f(X_i)$. The greatness of the $step\ size$ is designated by $\overrightarrow{step}_a$ and is strongminded by an unchanging delivery rang since $-a$ to a. Also, the size of the step, signified by $\overrightarrow{step}_b$, is resolute by an unchanging delivery rang since $-b$ to b. The morals of a and b are strongminded by Eq. (16); this is determined by the max iteration T and the current iteration T:

$$a=\operatorname{arctanh}\ \left(-\left(\frac{t}{T}\right)+1\right) \tag{16}$$

$$b=1-\frac{t}{T} \tag{17}$$

Eq. (10) shows that even with the $SMA's$ encouraging findings, the search procedure still has to be improved. It is important to recollect that count arbitrary SM can variation the exploration's direction. Local least may restrict the efficacy of the search strategy when choosing individuals to seek for $\vec{X}_D$ and $\vec{X}_C$ from a N slime molds' sample. This part presents the $EOSM$, a novel optimization method. By using an equipoise pool of four bigger location courses, this approach replaces the location vector $\vec{X}_A$. The EO idea is then applied to determine this selection's average position. Eq. (18) provides an exact definition of the equilibrium pool's constituents.

$$\begin{aligned} \vec{X}_{eq(1)}&=X\left(sortIndex(1)\right) \\ \vec{X}_{eq(2)}&=X\left(sortIndex(2)\right) \\ \vec{X}_{eq(3)}&=X\left(sortIndex(3)\right) \\ \vec{X}_{eq(4)}&=X\left(sortIndex(4)\right) \\ \vec{X}_{ave}&=\frac{\vec{X}_{eq(1)}+\vec{X}_{eq(2)}+\vec{X}_{eq(3)}+\vec{X}_{eq(4)}}{4} \end{aligned} \tag{18}$$

The equilibrium pool is built using a collection of five-position vectors, denoted by $\vec{X}_{eq,pool}$.

$$\vec{X}_{eq,pool}=\{\vec{X}_{eq(1)},\vec{X}_{eq(2)},\vec{X}_{eq(3)},\vec{X}_{eq(4)},\vec{X}_{ave}\} \tag{19}$$

In $ESMA$, the place vector for the $i-th\ SM$, X_i $(j=1,2,\ldots,N)$, throughout the original iteration $(t+1)$ is signified by the subsequent equation:

$$\vec{X}_i(t+1)=r_1.(UB-LB)+LB, when\ r_1<z \tag{20}$$

$$\vec{X}_i(t+1)=\vec{X}_{Gbest}+\overrightarrow{step}_a.\left(\vec{U}.\vec{X}_{eq}-\vec{X}_D\right),when\ r_2<p_i(t)\ and\ r_1\geq z \tag{21}$$

$$\vec{X}_i(t+1)=\overrightarrow{step}_b\ .\ \vec{X}_i(t)\ ,\ when\ r_2\geq p_i(t)\ and\ \ r_1\geq z \tag{22}$$

The location vector $\vec{X}_{eq}$ is learned by picking a vector at random from the evenness pool. The algorithmic tool z guarantees the success of $ESMA$ by encouraging exploration throughout the search phase by restricting restricted local occurrence. This is accomplished using a

threshold value of 0.03 that was determined through testing. It is important to remember that the ESMA method combines the local best position from the best-so-far equilibrium pool, the global best position, and a random vector for modifying the position vector in the subsequent iteration. This approach allows for a fair trade-off between exploration and exploitation. The recommended $ESMA$ is fully described in Algorithm 2.

Algorithm 2: $ESMA$ Algorithms pseudo-code

Begin
Inputs: $N, D, T, UB, LB, and\ z.$
Initialization: $x_i = \{x_i^1, x_i^2, \dots, x_i^D\}\ for\ i = 1, 2, \dots, N\ at\ a\ random\ position\ within\ the\ search\ boundary\ [LB, UB] for\ a\ D\ dimension\ at\ initial\ iteration\ t = 1\ to\ form\ X = [\vec{X}_1, \vec{X}_2, \dots, \vec{X}_i, \dots, \vec{X}_N]'.$
While $(t \le T)$
→ $Evaluate\ the\ fitness\ f(x)\ of\ N\ slime\ mould.$
→ $Sort\ the\ fitness\ value\ in\ ascending\ order\ (for\ minimization\ problem)\ using\ Eq.(17) and\ determine\ the\ f_{Lbest}\ using\ Eq.(19), f_{Lworst}\ using\ Eq.(20) and\ constructing\ the\ \vec{X}_{eq,pool}\ using\ Eq.(18).$
→ $Determine\ the\ weighting\ factor\ \vec{U}\ using\ Eq.(18).$
→ $Update\ the\ \vec{X}_{Gbest}.$
→ $Determine\ the\ a\ and\ b.$
For $i = 1 : number\ of\ slime\ mould\ (N)$
→ $Generate\ a\ random\ value\ r_1.$
If $r_1 < z$
$Update\ the\ position\ vector\ \vec{X}_i(t+1) = r_1.(UB - LB) + LB.$
Else If $r_1 \ge z$
$Update\ the\ probability\ p_i,\ \overrightarrow{step}_a\ and\ \overrightarrow{step}_b.$
$Randomly\ choose\ one\ position\ vector\ \vec{X}_{eq}\ from\ the\ equilibrium\ pool.$
$Generate\ a\ random\ value\ r_2.$
If $r_2 < p_i$
$Select\ a\ random\ position\ \vec{X}_D\ a\ vector\ from\ X.$
$Update\ the\ position\ vector\ \vec{X}_i(t+1) = \vec{X}_{Gbest} + \overrightarrow{step}_a\ .(\vec{U}.\ \vec{X}_{eq} - \vec{X}_D).$
Else If $r_2 \ge p_i$
$Update\ the\ position\ vector\ \vec{X}_i(t+1) = \overrightarrow{step}_b\ .\ \vec{X}_i(t).$
End If
End If
End For
→ $t = t + 1$
End while
Output: $\vec{X}_{Gbest}$
End

2.4 Runge Kutta optimization (RUN)

$$Z_{ij} = LB_j + r_1 \times \left(UB_j - LB_j \right) \quad i = 1, 2, \dots, N \quad , \quad j = 1, 2, \dots, P \tag{23}$$

The RKM, which was used to calculate solutions for first-order gap equations, is the foundation of the RUN optimization algorithm [39], [40]. The Runge Kutta method (RKM) generally yields a precise numerical result based on functions alone, without the need for any gradient information. Calculating slope within the RKM framework is crucial to the functioning of the RUN optimization algorithm to simulate the exploration abilities of swarm-based optimization as it enables the algorithm. The following are the phases that make up the mathematical equation of the RUN algorithm:

- According to the bounds of the search space (LB, UB), the first solutions for N agents are created during the startup step. To do this, the following equation is used:

The problem's dimension is factored into the formula and is represented by P, LB_j and UB_j represent the jth variable's $lower$ and $upper$ bounds in the solution set Z_{ij}, here i ranges from 1 to N, on behalf of the complete number of exploration agents. r_1, r_2, and r_3 are step coefficients used to control the magnitude of the function evaluations in each iteration.

- In order to update solutions, the RUN algorithm uses a search mechanism (SM) that makes use of the RKM to change the location of the current solution on each iteration. The expression for this mechanism is:

$$Z_i = \begin{cases} Z_{CF} + S_{FM} + \mu \times randn \times Z_{mc} & ,if\ rand \le 0.5 \\ Z_{mF} + S_{FM} + \mu \times randn \times Z_{ra} & ,\ otherwise \end{cases} \quad (24)$$

In Eq. (24), $Z_{CF} = (Z_c + r \times SF \times g \times Z_c)$ and $S_{FM} = SF \times SM$. $Z_{ra} = (Z_{r1} - Z_{r2})$, $Z_{MF} = (Z_M + r \times SF \times g \times Z_m)$ and $Z_{MC} = (Z_m - Z_c)$. The direction of the search is changed by using the integer number r, which is between -1 and 1. Conversely, the random numbers represented by the symbols μ and g range from 0 to 1, respectively, and 0 to 2. The following details apply to the adaptable factor SF:

$$SF = 2 \times (0.5 - rand) \times f$$
$$f = a \times \exp\left(-b \times rand \times \left(\frac{t}{tmax}\right)\right) \quad (25)$$

t tracks the current iteration in the optimization process. $Tmax$ is a symbol for the total number of iterations. In Eq. (26), the Z_c and Z_m values employed are distinct as:

$$Z_c = \varphi \times Z_i + (1 - \varphi) \times Z_{r1} \quad (26)$$

$$Z_m = \varphi \times Z_b + (1 - \varphi) \times Z_{pb} \quad (27)$$

Z_c and Z_m are weighting parameters that modulate the contribution of each function evaluation in the iterative process. A randomly generated number between 0 and 1, denoted by the φ, is included in Eq. (26 and 27). Here, Z_b and Z_{pb} indicate the top agent at each iteration and the top agent thus far, correspondingly. In Eq. (24), the SM is rationalized employing the subsequent equation:

$$\begin{aligned} &\mathrm{SM} = \frac{1}{6}(Z_{RK})\Delta Z; \\ &Z_{RK} = k_1 + 2 \times k_2 + 2 \times k_3 + k_4 \\ &k_1 = \frac{1}{2\Delta Z}(rand \times Z_w - u \times Z_b) \\ &k_2 = \frac{1}{2\Delta Z}(rand \times (Z_w + rand_1 \times k_1 \times \Delta Z) \\ &\qquad - UZ \\ &k_3 = \frac{1}{2\Delta Z}\Big(rand \times (Z_w \\ &\qquad + rand_1 \times (\frac{1}{2}k_2) \times \Delta Z\Big) \\ &\qquad - UZ_b \end{aligned} \quad (28)$$

$$\begin{aligned} k_4 &= \frac{1}{2\Delta Z}(rand \times (Z_w + rand_1 \times k_3 \times \Delta Z) \\ &\qquad - UZ_{b2} \\ u &= round(1 + rand) \times (1 - rand) \\ UZ &= (U \times Z_b + rand_2 \times k_1 \times \Delta Z) \\ UZ_b &= \left(U \times Z_b + rand_2 \times (\frac{1}{2}k_2) \times \Delta Z\right) \\ UZ_{b2} &= (U \times Z_b + rand_2 \times k_3 \times \Delta Z) \end{aligned}$$

Eq. (28) formulates the update rule for Z_{RK}, which plays a role in the Selection Mechanism (SM) by determining candidate solutions. The parameters u, k_1, and k_2 influence this update process and are defined as follows:"

u: A randomly generated value between 0 and 1, used to introduce stochastic variation.

k_1, k_2: Control parameters that balance the influence of different solutions (or learning factors, if applicable).

The symbols $rand1$ and $rand2$ characterize random statistics. The ΔZ (ΔZ controls how solutions are modified in each iteration) value is calculated as follows:

$$\begin{aligned} \Delta Z &= 2 \times rand \times |Stp|; \\ Stp &= rand \times ((Z_b - rand \times Z_{avg}) + y) \\ y &= rand(Z_n - rand \times (u - l)) \times \exp(-4 \\ &\qquad \times \frac{t}{tmax}) \end{aligned} \quad (29)$$

Stp represents a step size. l is an iteration index and y represents random variables.

The morals of Z_w and Z_b are updated in accordance with the formulas below:

Algorithm 2. Update Rule for Zw and Zb.

```
if
f (Zi) < f (Zpb)
Zb = Zi
Zw = Zpb
else
Zb = Zpb
Zw = Zi
end
```

- A variety of operators are used in the Enhanced Solution Quality stage in order to increase the convergence amount and prevent local ideals. The goal is to improve the superiority of answers, and the following procedure helps to do this:

$$\begin{aligned} Z_{new2} &= \begin{cases} Z_{new1} + r \times \omega \times \left|\left(Z_{new1} - Z_{avg}\right) + randn\right| & ,if\ \omega < 1 \\ \left(Z_{new1} \times Z_{avg}\right) + r \times \omega \times Z_{na} & otherwise \end{cases} \\ Z_{na} &= \left|\left(u \times Z_{new1} - Z_{avg}\right) + randn\right| \quad , \quad c = 5 \times rand \\ \omega &= rand(0,2).\exp\left(-c\left(\frac{t}{tmax}\right)\right), \quad Z_{avg} = \frac{Z_{r1} + Z_{r2} + Z_{r3}}{3} \end{aligned} \quad (30)$$

Z_{avg} represents the average position of the agents in the search space, indicating the mean of the agent positions. c is a constant that controls the influence of the random

$$Z_{new1} = \delta \times Z_{avg} + (1 - \delta) \times Z_b \quad (31)$$

perturbation in the algorithm, balancing exploration and exploitation during optimization. w is the inertia weight, which influences the velocity of the agents, helping to balance exploration and exploitation phases. The condition $if\ w < 1$ has also been clarified to explain its role in controlling the convergence behavior of the optimization process.

A chance amount between 0 and 1 and an integer Z_{new1} represents the updated solution after applying the influence of the average position of the agents (Z_{avg}) and the best agent's position (Z_b). Amount r, which might have values of 1, 0, or -1, are involved in the calculation in Eq. (31). According to [31], if the fitness value of Z_{new2}

is not better than the level of fitness of Z_i, then there's an additional chance to revise the worth of Z_i. This may be done by applying the following equation:

$$Z_{new3} = (Z_{new2} - r_1 \times Z_{new2}) + SF \times D_Z$$
$$D_Z = (r_2 \times Z_{RK} + (v \times Z_b - Z_{new2})) \quad (32)$$

There is a random value in this equation r_1, r_2, and r_3. The value of v (variable) is figured as twice the alteration of r_3 and 0.5, where r_3 is a accidental amount in the variety $[0, 1]$ [41].

2.5 Improved arithmetic optimization algorithm (IAOA)

The AOA algorithm apprises the population using the finest global answer, but if it reaches the best local solution, the population will stop improving. This can lead to premature convergence in certain cases [42]. Additionally, AOA does not fully utilize the diverse information contributed by individual members of the population, which can limit its exploratory capability. To address these limitations, this paper introduces the IAOA. IAOA enhances population diversity through a structured subpopulation mechanism, ensuring better exploration and exploitation. By dynamically adjusting the population's composition and incorporating adaptive strategies, IAOA prevents stagnation in local optima and improves convergence toward the global optimum.

2.5.1 Mechanism for population control

The important AOA exploits arithmetic operatives ($Multiplication$ (" × "), $Division$ (" ÷ "), $Subtraction$ (" − "), $and\ Addition$ (" + ")) to haphazardly investigate the area surrounding an ideal answer. Though, this technique can result in a reduction in population diversity. To mitigate this issue, it is necessary to categorize the population.

2.5.1.1 The Original subpopulation

Arrange the populace based on their suitability values and take the top num_best those as the initial subpopulation.

$$num_{best} = round\left(0.1Z + 0.5Z\left(1 - \frac{t}{T}\right)\right) \quad (33)$$

With Z standing for the amount of individuals and t (current iteration) and T (maximum iterations) for the present iteration and max iterations, alternatively, the mathematical model is in the following: Z individuals exchange information about each other to update their locations. The equation dynamically adjusts the number of individuals in the initial subpopulation (num_best) based on the iteration progress. As t increases, the term $0.5Z(1 - t/T)$ decreases, leading to a gradual reduction in the size of the best-performing subpopulation.

$$x_{best_i}(t+1) = x_{best_i}(t) + rand \times \left(best(x) - \frac{x_{best_i}(t) + x_{best_j}(t)}{2} \times \omega\right) \quad (34)$$

$$x_{best_j}(t+1) = x_{best_j}(t) + rand \times \left(best(x) - \frac{x_{best_i}(t) + x_{best_j}(t)}{2} \times \omega\right) \quad (35)$$

Eq. (34) introduces randomness to enhance exploration by perturbing the position of an individual x_{best_j} using a stochastic factor. This ensures diversity and prevents premature convergence. Eq. (35), on the other hand, applies an averaging mechanism to refine the positions of individuals based on mutual influence, promoting exploitation by stabilizing movement around the best-found solution.

The i_th separate location in the next iteration is signified by $x_{best_i}(t+1)$, and the same smears to $x_{best_j}(t+1)$. $best(x)$ symbolizes the worldwide optimal found by people following t iterations. x_{best_j} is selectively selected from the first group, ω signifies the rate of information gathering, which may range from 1 to 2.

2.5.1.2 The next subpopulation

Select num_middle those from the populace to form the next subpopulation.

$$num\_middle = round(0.3 \times Z) \quad (36)$$

These individuals are situated between num_best and num_worst in the po Selectively selected from the first group, ω signifies the rate of information gathering, which may range from 1 to 2. After that, they adjust their position, and the revised model looks like this:

$$x_{mid\_i}(t+1) = x_{mid\_i}(t) + Levy \times (best(x) - x_{mid\_j}) \quad (37)$$

The random selection of x_{mid_j} comes from the second group. $Levy$ is the $Levy$ distribution function [43], and $x_{mid\_i}(t+1)$ indicates where the i_th discrete will be located in the subsequent iteration.

The location of the i_th discrete in the next iteration is signified by $x_{mid\_i}(t+1)$. $Levy$ refers to the $Levy$ distribution function [35], and x_{mid_j} is randomly chosen from the second category.

2.5.1.3 The third subpopulation

To establish the final subpopulation, indicate the number of the population's worst members.

$$num\_worst = Z - (num\_best + num\_middle) \quad (38)$$

The final group of people use the following equation to update their position:

$$x_{worst_i}(t+1) = x_{worst_i} + \left(\frac{t}{T} \times best(x) - x_{worst\_j}\right) \quad (39)$$

After t iterations, $best(x)$ signifies the global optimum discovered through individuals, and the location of the i_th separate in the next iteration is signified by $x_{worst\_i}(t+1)$. During the early stages of $IAOA$, more persons are in the first subpopulation to accelerate the

global optimum update. The operator crowding problem is resolved close to the optimum as the algorithm runs because fewer individuals are in the first subpopulation. Furthermore, the size of the third subpopulation rises, so averting the population from becoming stuck in local optima. Another subpopulation looks for more gifted places by using Levy flying for small-step upgrades. This approach can successfully address the limitations of outdated AOA and enhance its performance. Fig. 2 depicts the $IAOA$'s flowchart.

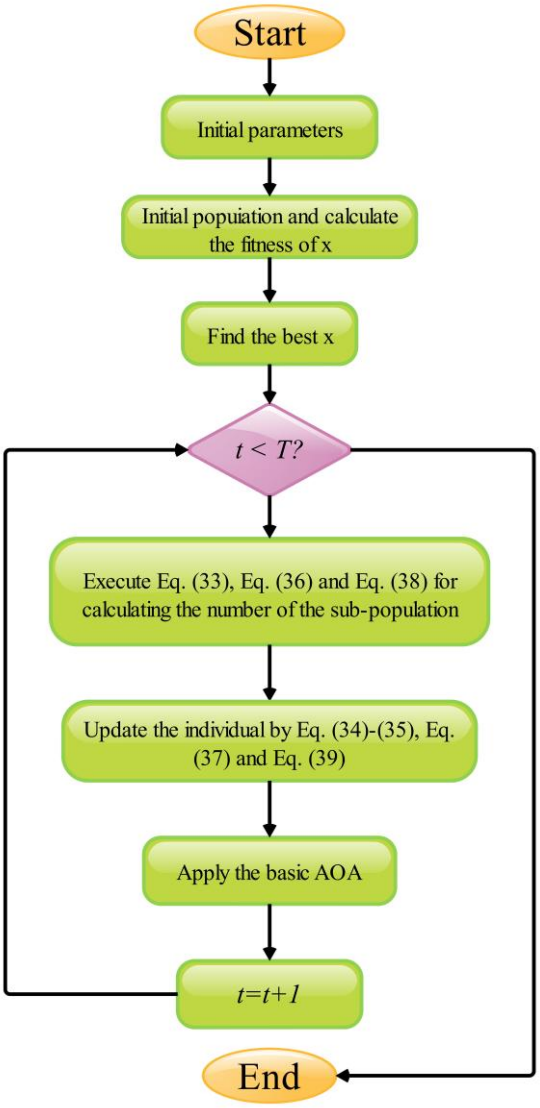


Figure 2: The flowchart of IAOA.

2.6 Performance evaluation methods

This study employed a set of evaluation metrics to assess the performance of the proposed models (LSIA, LSEM, and LSRK) for CBR prediction. Each of these metrics was selected to provide complementary insights into the models' predictive quality, accuracy, and robustness. Below is the justification for the selection of each evaluation metric:

1. **Symmetric Mean Absolute Percentage Error (SMAPE):**
   - **Justification:** SMAPE is a widely used metric in regression analysis, particularly in fields where data ranges vary significantly. Unlike traditional Mean Absolute Percentage Error (MAPE), SMAPE normalizes the error by both the actual and predicted values, making it scale-independent and preventing bias from high values in the dataset. It provides a balanced view of prediction error across different magnitudes of data, making it suitable for geotechnical engineering applications such as CBR prediction, where data variability is expected.
   - **Benefit:** SMAPE is particularly effective in capturing the relative error across the entire dataset and provides an intuitive percentage error that is easier to interpret regarding prediction accuracy.
2. **Root Mean Square Error (RMSE):**
   - **Justification:** RMSE is one of the most commonly used error metrics in regression problems, and it measures the square root of the average squared differences between predicted and actual values. RMSE is sensitive to large errors, giving more weight to large deviations between predicted and actual values, which makes it useful when large errors are particularly undesirable in the context of CBR prediction.
   - **Benefit:** RMSE is valuable for understanding the model's overall fit and is particularly useful when a high deviation from the actual values is considered critical. Its units are the same as the original data, which provides a tangible measure of model accuracy.
3. **Mean Absolute Error (MAE):**
   - **Justification:** MAE is another commonly used metric that measures the average absolute differences between predicted and actual values. Unlike RMSE, MAE does not square the errors and thus treats all errors equally without disproportionately penalizing larger deviations. This makes it a robust measure when understanding the average magnitude of prediction errors.
   - **Benefit:** MAE provides a more straightforward measure of the average error magnitude and is less sensitive to outliers than RMSE, making it a useful metric when a balanced view of prediction errors is needed.
4. **Coefficient of Determination (R^2):**
   - **Justification:** R^2, or the coefficient of determination, measures the proportion of variance in the dependent variable that is predictable from the independent variables. It is a key metric for understanding how well the

model fits the data. A higher R^2 value indicates that the model has a better fit and explains more of the variability in the data.

- **Benefit:** R^2 provides an overall indication of model performance, giving insight into how well the proposed models explain the variability in the CBR data. It is particularly useful for comparing the relative performance of different models.

5. **Scatter Index (SI):**

- **Justification:** SI is a metric used to evaluate the relative error by calculating the mean absolute error ratio to the actual data's mean value. It is useful for assessing how well the model captures the data's general trend while factoring in overestimating and underestimating predicted values.
- **Benefit:** The SI value provides an intuitive measure of the scatter of predicted values relative to the true values. This makes it particularly useful in applications like CBR prediction, where an accurate overall distribution of predictions is as important as fitting specific data points.

The evaluators comprise $SMAPE$, $RMSE$, MAE [44], R^2, SI. Below is a list of the formulae for each of these measurements. While lower values of metrics near to zero, including $RMSE, SMAPE, SI,$ and MAE, are preferable and a high R^2 value near to 1 shows the good performance of the algorithms.

$$R^2 = \left(\frac{\sum_{i=1}^{w}(p_i - \bar{p})(q_i - \bar{q})}{\sqrt{[\sum_{i=1}^{w}(p_i - p)^2][\sum_{i=1}^{w}(q_i - \bar{q})^2]}} \right)^2 \quad (40)$$

$$RMSE = \sqrt{\frac{1}{w}\sum_{i=1}^{w}(q_i - p_i)^2} \quad (41)$$

$$MAE = \frac{1}{w}\sum_{i=1}^{w}|q_i - p_i| \quad (42)$$

$$SMAPE = \frac{100}{w}\sum_{i}^{w}\frac{2 \times |q_i - p_i|}{|q_i| + |p_i|} \quad (43)$$

$$SI = \frac{RMSE}{meam(q_i)} \quad (44)$$

In this background, p_i and q_i define the forecast and experimental values, alternatively. The mean values of the experimental samples and predicted are defined by $\bar{p}$ and $\bar{q}$. Otherwise, w signifies the number of samples being measured.

2.7 Hyperparameter

Table 3 provides the optimized hyperparameters (C and Gamma) for the three hybrid models—LSIA, LSEM, and LSRK—demonstrating variations in parameter selection due to different optimization strategies. These hyperparameters were determined through random search, an efficient tuning method that explores a predefined range of values without requiring an exhaustive grid search. The regularization parameter (C) influences model complexity and error minimization trade-offs. Higher values, such as 382 in LSIA, suggest a model that prioritizes reducing training error, while lower values, such as 194 in LSRK, indicate a model with more flexibility to generalize to unseen data. Similarly, the Gamma parameter, which controls the influence of training samples in the RBF kernel, was highest in LSIA (2.123) and lowest in LSRK (1.193), signifying different levels of local adaptability in decision boundaries. The random search approach enabled an efficient and effective selection of hyperparameters, preventing the risk of overfitting while ensuring robust performance. Each optimization algorithm refined the chosen parameters, including IAOA for LSIA, ESMOA for LSEM, and RKO for LSRK. LSIA, which utilized the IAOA, exhibited the highest values for both C and Gamma, suggesting a more aggressive tuning strategy that improved model precision. LSEM and LSRK, optimized with the ESMOA and RKO methods, resulted in different parameter distributions, reflecting variations in their exploration-exploitation mechanisms.

Table 3: Hyperparameter of the developed hybrid models

Model	Hyperparameter	
	C	Gama
LSIA	382	2.123
LSEM	243	1.970
LSRK	194	1.193

2.8 Convergence

Fig. 3 shows the convergence of the developed hybrid models based on RMSE. The LSIA obtained the lower RMSE value compared to the other two models. The algorithms were initialized with a population size of 50, a maximum of 200 iterations, and a convergence threshold of RMSE. The experiments were conducted on a system, an Intel Core i7 processor, 64GB RAM, and an 11th Gen Intel(R), using Python 3.9 and Scikit-learn.

Beyond accuracy metrics, the computational efficiency of LSIA was assessed. The average training time for LSIA was 70 seconds, significantly lower than the LSRK and LSEM models, which required 4500 and 1500 seconds, respectively. Additionally, the LSIA model achieved convergence in fewer iterations (175 iterations), demonstrating its faster optimization process. While energy consumption was not explicitly measured, LSIA's reduced computational load suggests lower power usage, making it a feasible choice for large-scale geotechnical applications.

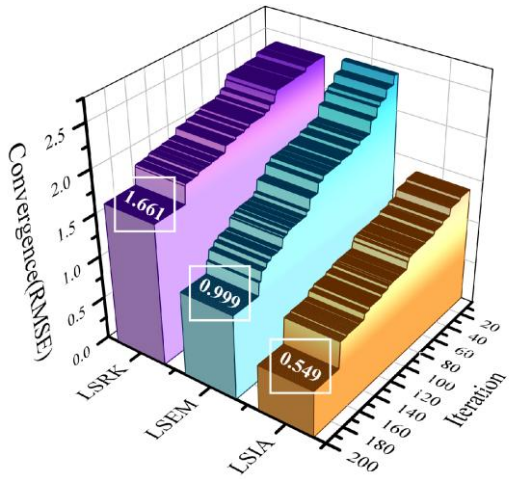


Figure 3: Convergence of the developed hybrid models.

2.9 Hybridization

The hybridization process in this study combines LSSVR with three optimization algorithms: IAOA, ESMA, and RKO. These algorithms fine-tune the LSSVR model's hyperparameters, enhancing its ability to predict the CBR. The hybridization process begins with initializing the LSSVR model and then applying each optimization algorithm to search for the optimal hyperparameters, such as C and gamma. The best-performing hyperparameters, derived through optimization, are then used to train the LSSVR model, ensuring improved prediction accuracy. Figs. 4 to 6 indicate the hybridization process for the LSIA, LSRK, and LSEM, respectively.

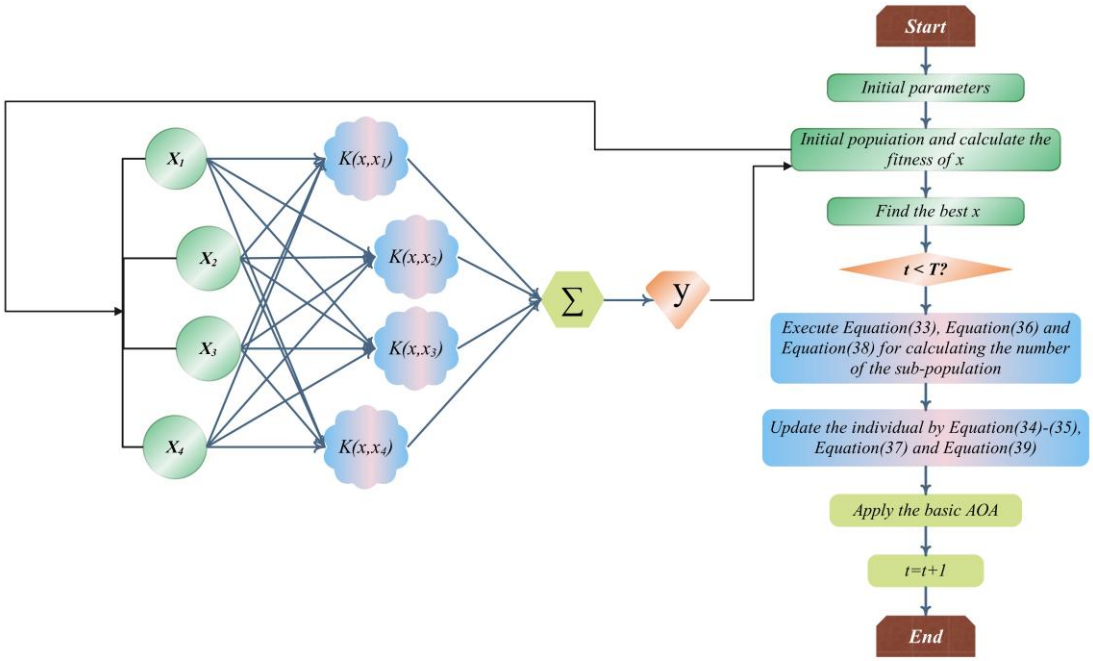


Figure 4: Hybridization of the LSSVR+IAOA.

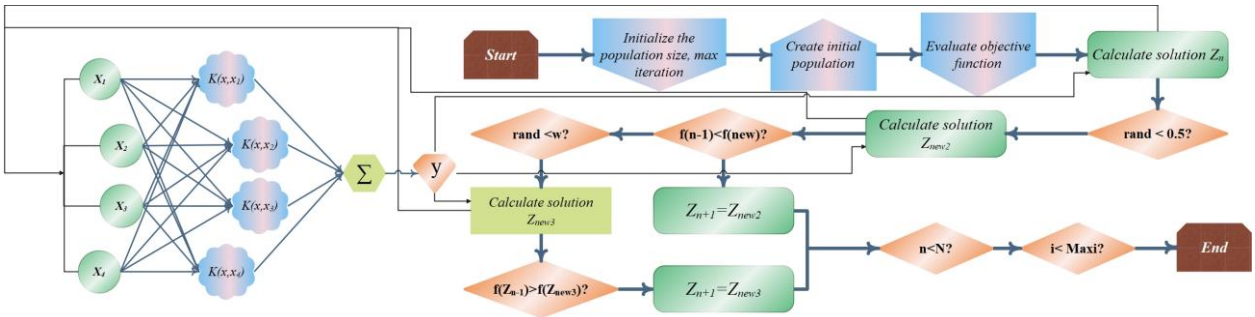


Figure 5: Hybridization of the LSSVR+RKO.

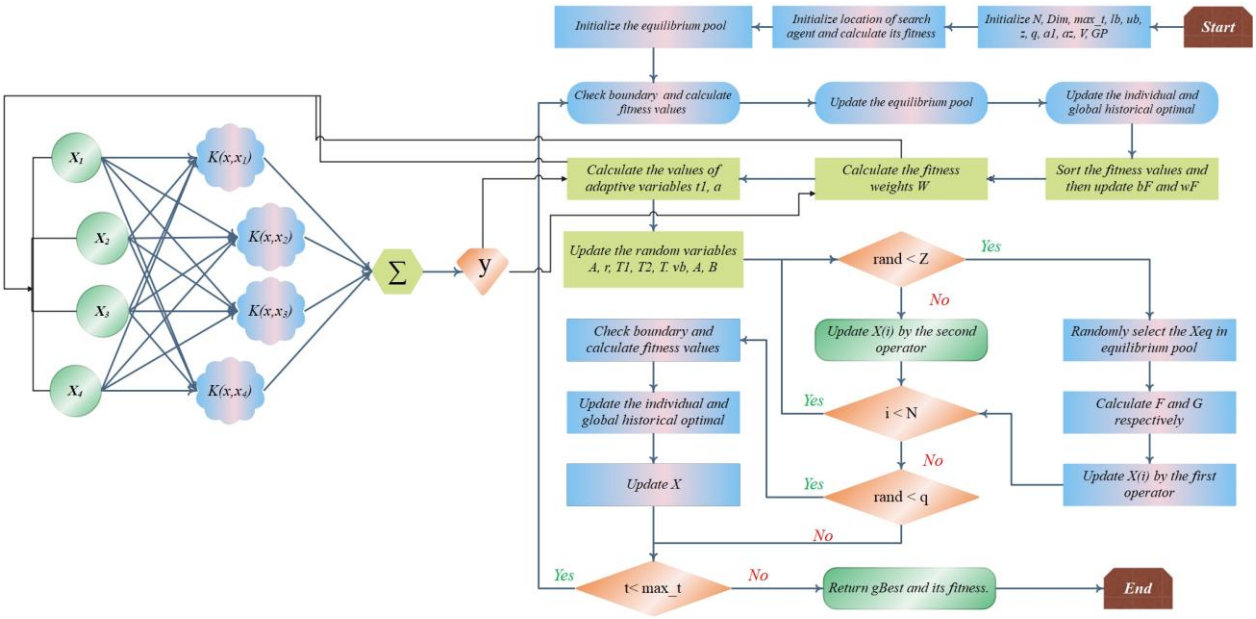


Figure 6: Hybridization of the LSSVR+ESMOA.

3 Results

Three models, namely LSIA, LSEM, and LSRK, were utilized in the research to forecast the CBR as output. The models were assessed based on their accuracy against experimental measurements during the training and testing stages. The experimental data were split into train (70%) and test (30%) sets to ensure an impartial evaluation of the models. The study utilized 5 statistical metrics containing R^2, $RMSE, SMAPE, SI$, and MAE to evaluate lengthily and comparison the research's algorithms. In the study, these evaluators were utilized to determine the efficacy of the algorithms secondhand.

Table 4 shows the results of the developed models. R^2 values, which indicate the percentage of the dependent variable's variation that the independent variable accounts for, were used to evaluate the models. The LSIA model indicated excellent predictive accuracy, with the highest R^2 values of 0.9975 and 0.9932 in the train and test stages. The LSEM model exhibited moderately high R^2 values of 0.9911 and 0.9828 in the train and test stages, while the LSRK model demonstrated lower R^2 values of 0.9756 and 0.9681 in the training and testing phases alternatively. However, all models demonstrated acceptable predictive accuracy.

The findings showed that the LSIA model had lower error indicators in the training and testing phases, indicating superior performance compared to the LSEM and LSRK models. Overall, the results suggest that LSIA may be a more suitable model for predicting CBR than LSEM and LSRK. However, when choosing a model for practical uses, additional aspects, including computational effectiveness, model complexity, and simplicity of implementation, should also be taken into account. Furthermore, the study revealed that all three optimization algorithms successfully enhanced the $LSSVR$ model's ability to forecast CBR, with the IAOA algorithm outperforming others by obtaining the highest R^2 values and the lowest $RMSE$ and MAE.

Table 4: Evaluation of developed hybrid models.

Hybrid Models	*Sections*	*Evaluators*				
		RMSE	R^2	*MAE*	*SI*	*SMAPE*
LSIA	Train	0.5489	0.9975	0.3617	0.014	0.0001
	Test	1.1270	0.9932	0.8324	0.0269	0.0006
LSEM	Train	0.9997	0.9911	0.8286	0.0255	0.0002
	Test	1.5589	0.9828	1.2039	0.0373	0.0008
LSRK	Train	1.6624	0.9756	1.3397	0.0425	0.0004
	Test	2.2305	0.9681	1.7882	0.0533	0.0013

The scatter plot in Fig. 7 illustrates a comparison of the hybrid models' performance according to two parameters: R^2 and RMSE. R^2 represents the level of agreement, while $RMSE$ represents the degree of dispersion. The plot's centerline is at the $X = Y$ coordinates, and the points' distance from the centre line reflects the model's accuracy level. The LSIA model demonstrated a high accuracy level, with its data points tightly clustered around the centerline, indicating a narrow range of dispersion. On the other hand, the data points of the LSEM and LSRK models showed comparatively equal performance levels.

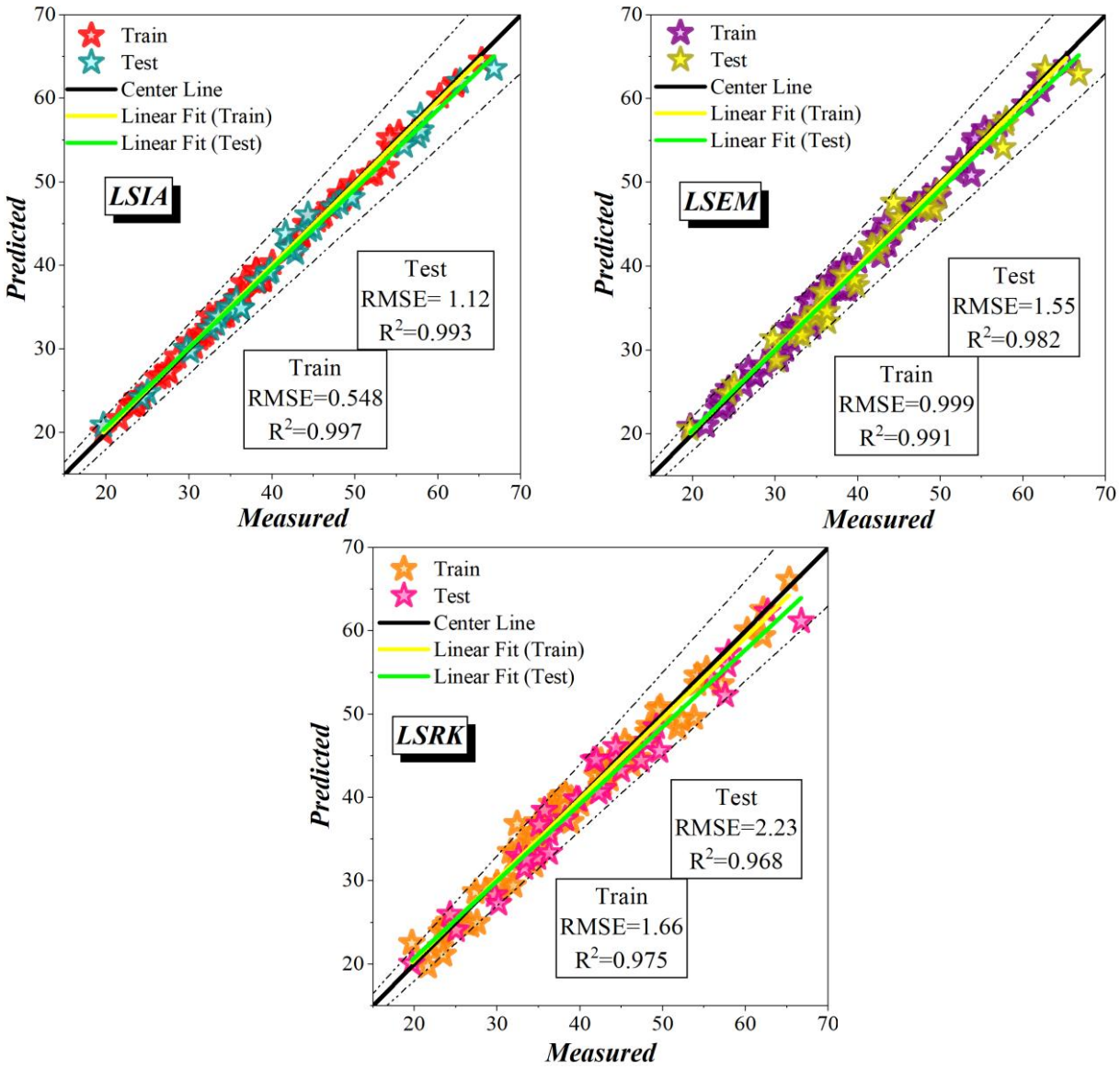
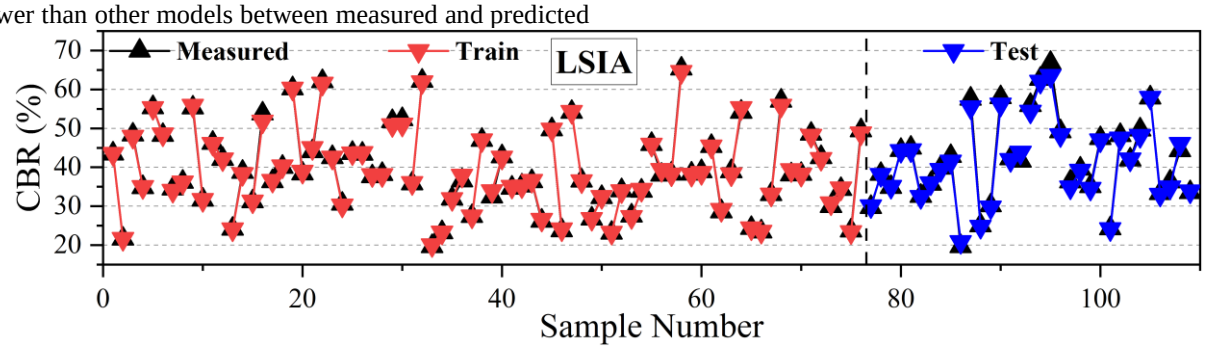


Figure 7: The models' scatter plot, which shows corresponding measured and predicted values

As shown in Fig. 8, the practical variance was influenced by the difference between the actual and estimated values, which decreased significantly in the testing phase. In the training phase, the LSIA model demonstrated minimal dispersion, and the difference was lower than other models between measured and predicted values. Although some samples showed disparities among actual and estimated values in the training phase, resulting in significant divergences, improvements in performance and positive learning outcomes have somewhat reduced this weakness.



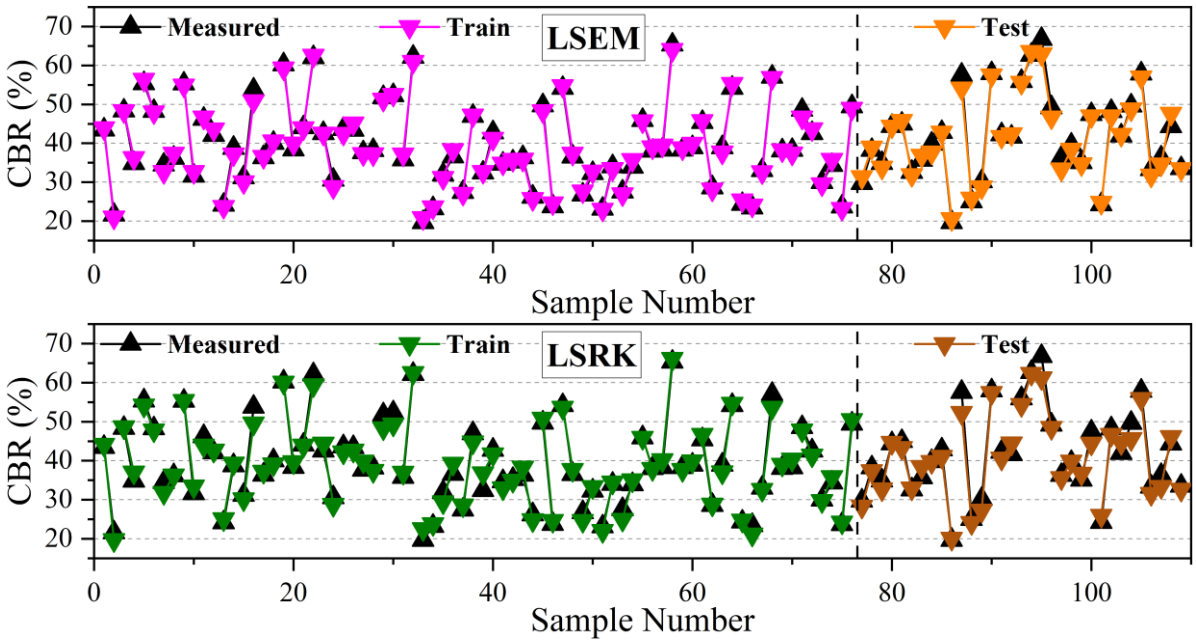


Figure 8: The comparison of the measured and predicted values

Figs. 9 and 610 visually present the error percentage in the models using a normal distribution plot and a violin diagram. In the LSIA model, the highest frequency of errors is below 1%, indicating high accuracy. In Fig. 10, most errors are below 5%, indicating good performance for LSIA. The performance of LSEM is comparable to that of LSIA when the error percentage in both figures is observed. In contrast, LSRK has the worst performance, with the most distributions in its data, as shown by the violin plot. These results suggest that LSIA is the most accurate and reliable model among the provided models.

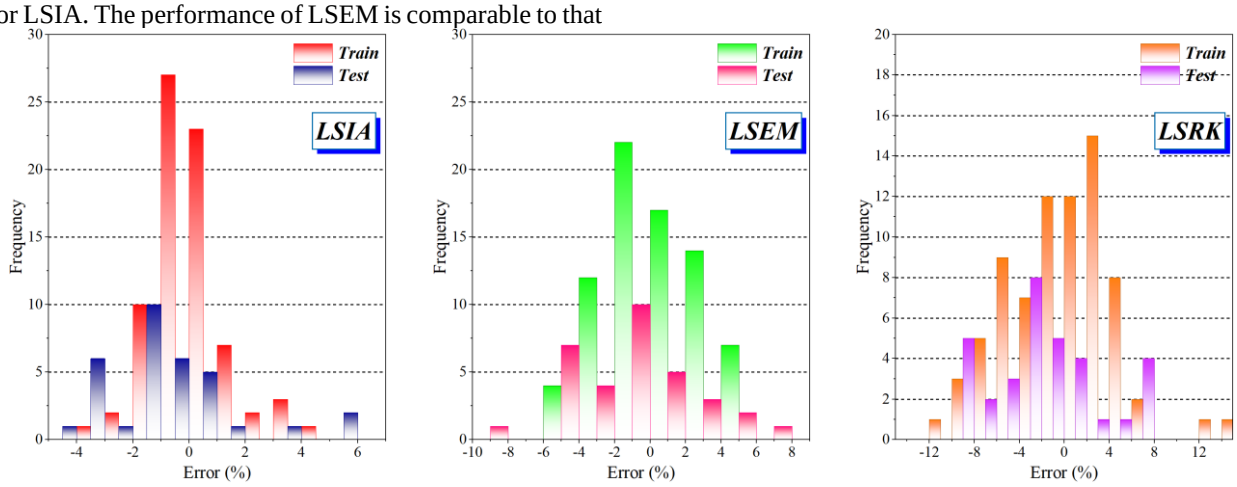


Figure 9: The error percentage of developed models is based on the line plot.

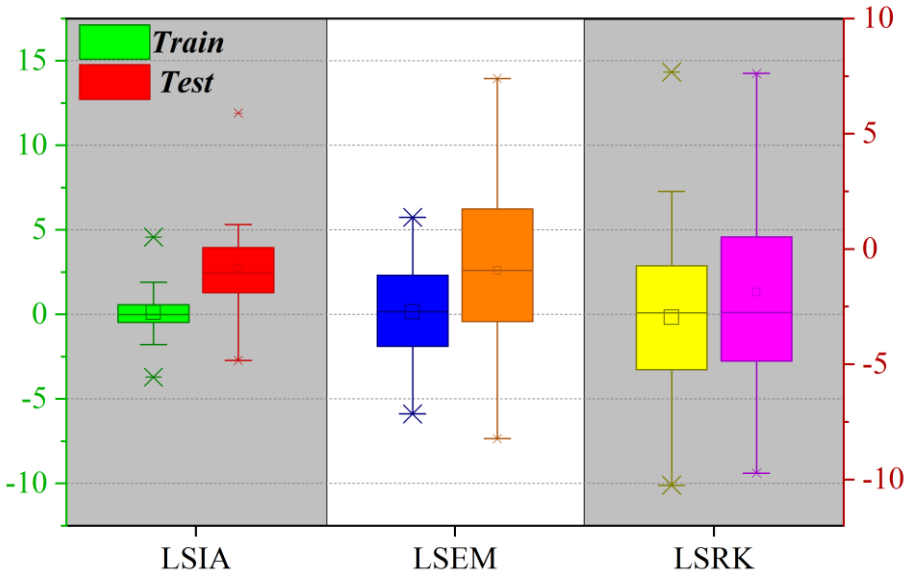


Figure 10: The violin diagram for error percentage of models.

4 Discussion

4.1 Comparison with other published studies

Table 5 presents a comparative analysis of previous studies on CBR prediction, showcasing different ML models along with their respective R^2 and RMSE values. The present study's LSIA model achieves an R^2 of 0.9932 and an RMSE of 1.1270, outperforming several state-of-the-art (STOA) models. Compared to traditional methods like MLR [19] with an R^2 of 0.841 and ANN [21,23] with R^2 values of 0.945 and 0.970, respectively, LSIA demonstrates significantly improved predictive accuracy. The RF model [25] (R^2 = 0.98) and LightGBM [26] (R^2 = 0.947) also fall short in comparison, indicating LSIA's stronger generalization ability.

Among the highest-performing SOTA models, GPR [22] achieves an R^2 of 0.999 and a remarkably low RMSE of 0.139, while MARS-L [24] attains an R^2 of 0.969 and an RMSE of 0.036. While these models yield excellent predictive accuracy, computational constraints often hinder their practical implementation. GPR, for instance, requires significant computational power and does not scale well with large datasets, making it less feasible for real-world applications. MARS-L's accuracy depends on proper feature selection, which adds complexity to model deployment. In contrast, LSIA balances high accuracy with computational efficiency, making it a more viable choice for large-scale geotechnical applications.

LSIA's superior performance is primarily due to its enhanced parameter optimization. Unlike traditional models that rely on manual tuning or grid search, LSIA fine-tuned the C and Gamma of LSSVR. This results in better convergence, improved generalization, and reduced overfitting. Using the RBF kernel further enhances LSIA's predictive capabilities by effectively capturing non-linear relationships in CBR prediction. In contrast, tree-based models like RF and LGBM may struggle with overfitting or underfitting when applied to complex datasets, leading to suboptimal predictions.

A key factor differentiating LSIA from its counterparts is its ability to balance accuracy and computational efficiency. While GPR achieves marginally better accuracy, its resource-intensive nature limits its scalability. LSIA, on the other hand, delivers high accuracy with lower computational demands, making it more practical for large datasets and real-world geotechnical applications. The model's robust generalization ability ensures that it remains reliable across diverse datasets, reducing the risk of overfitting and improving predictive stability.

Comparing LSIA with LSEM and LSRK, it is evident that LSIA consistently outperforms the other two hybrid models. While all three approaches benefit from hybrid optimization, LSIA's use of IAOA proves more effective than ESMA in LSEM and RKO in LSRK. This is reflected in LSIA achieving the highest R^2 and the lowest RMSE among the three, indicating that IAOA is better suited for optimizing LSSVR parameters. While LSEM and LSRK still show significant improvements over traditional models, their optimization strategies do not reach the same level of refinement as LSIA, explaining the slight performance gap.

In conclusion, LSIA stands out as a robust and efficient alternative for CBR prediction, surpassing most existing SOTA models. While GPR and MARS-L achieve high accuracy, their computational demands and feature selection dependencies limit their practicality. LSIA's hybrid optimization approach, leveraging IAOA, enhances both predictive accuracy and efficiency, making it a superior choice for geotechnical applications. The results validate that LSIA is a reliable and scalable solution capable of providing accurate CBR predictions while addressing the limitations observed in previous studies.

Table 5: Summary of previous studies related to CBR prediction.

Article	Model	Evaluator	
		R^2	RMSE
[18]	MLR	0.841	-
[19]	ANN	0.981	1.187
[20]	ANN	0.945	2.81
[21]	GPR	0.999	0.139
[22]	ANN	0.970	1.691
[23]	MARS-L	0.969	0.036
[24]	RF	0.98	1.43
[25]	LGBM	0.947	0.33
Present Study	LSIA	0.9932	1.1270

5 Conclusion

To ensure durable and flexible pavements, it is often necessary to use reliable techniques to assess or validate the California Bearing Ratio (CBR). However, the traditional method of determining the CBR of the subgrade can be challenging due to the significant time required for testing. As a result, there is a need to investigate other approaches, such as creating predictive models, to estimate the CBR of expansive soil subgrades. This justification has led to a greater usage of machine learning than human experimentation. This paper aims to provide an overview of LSSVR as a prevalent machine-learning technique for predicting CBR. Moreover, 3 meta-heuristic algorithms—IAOA, RKO, and ESMA—were combined with the proper model to create a hybrid model that would increase accuracy and reduce possible mistakes. In this study, several performance metrics containing R^2, $RMSE, MAE, SMAPE$, and SI, were applied to evaluate the effectiveness of the models in the train and test phases, in which 70% of samples belonged to training and 30% to testing phases. The evaluation of the achievements of the blended models was conducted using distinct criteria that were applied in the model selection process. As a result, the hybrid LSIA can obtain the highest R^2 with slight differences compared to the other two models. In addition, in RMSE, the LSIA achieved a 45% and 67% difference from LSEM and LSRK. Generally, it can be inferred that the IAOA model, when used with LSSVR, demonstrated superior performance than the other two models in both the training and testing phases.

The study demonstrated that the LSIA model outperforms the LSEM and LSRK models in predicting the CBR, achieving superior performance metrics such as high R^2 and low RMSE. While these results highlight LSIA's predictive accuracy, a more critical evaluation of its computational costs and practical applications within geotechnical engineering is necessary. Although LSIA offers improved accuracy, it may incur higher computational demands, which need to be weighed against its performance, especially in real-world scenarios where resources and time are limited. The manuscript should discuss the computational efficiency of LSIA compared to other methods, particularly regarding training time and resource usage, to help determine if the accuracy improvements justify these costs. In geotechnical contexts, LSIA could be particularly beneficial in projects with heterogeneous soil conditions, where traditional methods might not perform well. However, the paper should explore specific situations where LSIA could provide the greatest value, such as in projects requiring high accuracy and model robustness. Additionally, real-world case studies or field tests would further solidify the practical relevance of LSIA, demonstrating its utility across various soil types and conditions. Future work should include a detailed computational efficiency analysis of LSIA, particularly in comparison with other models, to determine its scalability for large-scale projects. Real-world applications and sensitivity analysis would help refine the model and make it more adaptable to diverse geotechnical datasets. Furthermore, exploring hybrid models that combine LSIA with other algorithms could offer a balance between accuracy and computational efficiency, making it more suitable for practical implementation. In conclusion, while LSIA shows promising potential in CBR prediction, a deeper exploration of its computational feasibility and real-world applicability will enhance its value for geotechnical engineering applications.

References

[1] B. Yildirim and O. Gunaydin, "Estimation of California bearing ratio by using soft computing systems," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 6381–6391, 2011.

[2] Tja. Taskiran, "Prediction of California bearing ratio (CBR) of fine grained soils by AI methods," *Adv. Eng. Softw.*, vol. 41, no. 6, pp. 886–892, 2010.

[3] P. G. Asteris, A. D. Skentou, A. Bardhan, P. Samui, and P. B. Lourenço, "Soft computing techniques for the prediction of concrete compressive strength using Non-Destructive tests," *Constr. Build. Mater.*, vol. 303, p. 124450, 2021.

[4] S. Bhatt, P. K. Jain, and M. Pradesh, "Prediction of California bearing ratio of soils using artificial neural network," *Am. Int. J. Res. Sci. Technol. Eng. Math*, vol. 8, no. 2, pp. 156–161, 2014.

[5] B. Naeim, A. J. Khiavi, P. Dolatimehr, and B. Sadaghat, "Novel Optimized Support Vector Regression Networks for Estimating Fresh and Hardened Characteristics of SCC," 2024.

[6] A. M. Ebid, "35 Years of (AI) in geotechnical engineering: state of the art," *Geotech. Geol. Eng.*, vol. 39, no. 2, pp. 637–690, 2021.

[7] M. W. Kin, "California bearing ratio correlation with soil index properties," *Master degree Proj. Fac. Civ. Eng. Univ. Technol. Malaysia*, 2006.

[8] D. K. Talukdar, "A study of correlation between California Bearing Ratio (CBR) value with other properties of soil," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 4, no. 1, pp. 559–562, 2014.

[9] L. Xiao-xia, "Predicting California-bearing capacity value of stabilized pond ash with lime

and lime sludge applying hybrid optimization algorithms," *Multiscale Multidiscip. Model. Exp. Des.*, vol. 5, no. 2, pp. 157–166, 2022.

[10] T. Varol *et al.*, "Prediction of soil-bearing capacity on forest roads by statistical approaches," *Environ. Monit. Assess.*, vol. 193, no. 8, p. 527, 2021, doi: 10.1007/s10661-021-09335-0.

[11] I. González Farias, W. Araujo, and G. Ruiz, "Prediction of California bearing ratio from index properties of soils using parametric and non-parametric models," *Geotech. Geol. Eng.*, vol. 36, no. 6, pp. 3485–3498, 2018.

[12] Behnam Sedaghat, G. G. Tejani, and S. Kumar, "Predict the Maximum Dry Density of soil based on Individual and Hybrid Methods of Machine Learning," *Adv. Eng. Intell. Syst.*, vol. 002, no. 03, 2023, doi: 10.22034/aeis.2023.414188.1129.

[13] B. Mahesh, "Machine learning algorithms-a review," *Int. J. Sci. Res. (IJSR).[Internet]*, vol. 9, pp. 381–386, 2020.

[14] Z.-H. Zhou, *Machine learning*. Springer Nature, 2021.

[15] H. Wang, Z. Lei, X. Zhang, B. Zhou, and J. Peng, "Machine learning basics," *Deep Learn.*, pp. 98–164, 2016.

[16] G. Biau, "Analysis of a random forests model," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 1063–1095, 2012.

[17] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science (80-.).*, vol. 349, no. 6245, pp. 255–260, 2015.

[18] V. Y. Katte, S. M. Mfoyet, B. Manefouet, A. S. L. Wouatong, and L. A. Bezeng, "Correlation of California bearing ratio (CBR) value with soil properties of road subgrade soil," *Geotech. Geol. Eng.*, vol. 37, pp. 217–234, 2019.

[19] A. K. Sabat, "Prediction of California bearing ratio of a soil stabilized with lime and quarry dust using artificial neural network," *Electron. J. Geotech. Eng.*, vol. 18, pp. 3261–3272, 2013.

[20] K. Othman and H. Abdelwahab, "The application of deep neural networks for the prediction of California Bearing Ratio of road subgrade soil," *Ain Shams Eng. J.*, vol. 14, no. 7, p. 101988, 2023.

[21] M. Ahmad *et al.*, "Predicting California bearing ratio of HARHA-treated expansive soils using Gaussian process regression," *Sci. Rep.*, vol. 13, no. 1, p. 13593, 2023.

[22] T. F. Kurnaz and Y. Kaya, "Prediction of the California bearing ratio (CBR) of compacted soils by using GMDH-type neural network," *Eur. Phys. J. Plus*, vol. 134, Jul. 2019, doi: 10.1140/epjp/i2019-12692-0.

[23] A. Bardhan, C. Gokceoglu, A. Burman, P. Samui, and P. G. Asteris, "Efficient computational techniques for predicting the California bearing ratio of soil in soaked conditions," *Eng. Geol.*, vol. 291, p. 106239, 2021, doi: https://doi.org/10.1016/j.enggeo.2021.106239.

[24] D. Q. Vu, D. D. Nguyen, Q.-A. T. Bui, D. K. Trong, I. Prakash, and B. T. Pham, "Estimation of California bearing ratio of soils using random forest based machine learning," *J. Sci. Transp. Technol.*, pp. 48–61, 2021.

[25] V. Quan and H. Q. Do, "Prediction of California bearing ratio (CBR) of stabilized expansive soils with agricultural and industrial waste using light gradient boosting machine," *J. Sci. Transp. Technol.*, pp. 1–9, 2021.

[26] C. C. Ikeagwuani, "Optimisation of additives for expansive soil reinforcement," *Unpubl. PhD thesis*, 2019.

[27] J. A. K. Suykens and J. Vandewalle, "Kisi, O. Streamflow forecasting and estimation using least square support vector r," *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.*, vol. 47, no. 7, pp. 1109–1114, 2000.

[28] O. Kisi, "Streamflow forecasting and estimation using least square support vector regression and adaptive neuro-fuzzy embedded fuzzy c-means clustering," *Water Resour. Manag.*, vol. 29, pp. 5109–5127, 2015.

[29] M. Kumar and I. N. Kar, "Non-linear HVAC computations using least square support vector machines," *Energy Convers. Manag.*, vol. 50, no. 6, pp. 1411–1418, 2009.

[30] M. M. Ghiasi, A. Shahdi, P. Barati, and M. Arabloo, "Robust modeling approach for estimation of compressibility factor in retrograde gas condensate systems," *Ind. Eng. Chem. Res.*, vol. 53, no. 32, pp. 12872–12887, 2014.

[31] N. M. Mahmoodi, M. Arabloo, and J. Abdi, "Laccase immobilized manganese ferrite nanoparticle: Synthesis and LSSVM intelligent modeling of decolorization," *Water Res.*, vol. 67, pp. 216–226, 2014.

[32] O. Kisi and K. S. Parmar, "Application of least square support vector machine and multivariate adaptive regression spline models in long term prediction of river water pollution," *J. Hydrol.*, vol. 534, pp. 104–112, 2016.

[33] D. Moreno-Salinas, D. Chaos, E. Besada-Portas, J. A. López-Orozco, J. M. de la Cruz, and J. Aranda, "Semiphysical modelling of the non-linear dynamics of a surface craft with LS-SVM," *Math. Probl. Eng.*, vol. 2013, 2013.

[34] S.-G. Cao, Y.-B. Liu, and Y.-P. Wang, "A forecasting and forewarning model for methane hazard in working face of coal mine based on LS-SVM," *J. China Univ. Min. Technol.*, vol. 18, no. 2, pp. 172–176, 2008.

[35] R. Fletcher, "Practical methods ofoptimization," *JohnWiley Sons, New York*, 1987.

[36] J. A. K. Suykens, J. De Brabanter, L. Lukas, and J. Vandewalle, "Weighted least squares support vector machines: robustness and sparse approximation," *Neurocomputing*, vol. 48, no. 1–4, pp. 85–105, 2002.

[37] S. Yin, Q. Luo, and Y. Zhou, "EOSMA: an equilibrium optimizer slime mould algorithm for engineering design problems," *Arab. J. Sci. Eng.*,

vol. 47, no. 8, pp. 10115–10146, 2022.

[38] M. K. Naik, R. Panda, and A. Abraham, "An entropy minimization based multilevel colour thresholding technique for analysis of breast thermograms using equilibrium slime mould algorithm," *Appl. Soft Comput.*, vol. 113, p. 107955, 2021.

[39] I. Ahmadianfar, A. A. Heidari, A. H. Gandomi, X. Chu, and H. Chen, "RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method," *Expert Syst. Appl.*, vol. 181, p. 115079, 2021.

[40] D. Yousri *et al.*, "Modified interactive algorithm based on Runge Kutta optimizer for photovoltaic modeling: justification under partial shading and varied temperature conditions," *IEEE Access*, vol. 10, pp. 20793–20815, 2022.

[41] H. Shaban *et al.*, "Identification of parameters in photovoltaic models through a runge kutta optimizer," *Mathematics*, vol. 9, no. 18, p. 2313, 2021.

[42] R. Szczepanski and T. Tarczewski, "Global path planning for mobile robot based on Artificial Bee Colony and Dijkstra's algorithms," in *2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC)*, 2021, pp. 724–730.

[43] N. E. Humphries *et al.*, "Environmental context explains Lévy and Brownian movement patterns of marine predators," *Nature*, vol. 465, no. 7301, pp. 1066–1069, 2010.

[44] A. Botchkarev, "Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology," *arXiv Prepr. arXiv1809.03006*, 2018.