Prediction of Blast-induced Ground Vibration using Eight New Intelligent Models

Zhang Zihan, Liu Xiaosheng, Wu Lijun, Hu Guangqiu

***Abstract*—Blast-induced ground vibration is the key negative effect of blasting, so its prediction and control are of great engineering value. With this aim, this study uses support vector regression (SVR) and gaussian process regression (GPR) as prediction tools, and four optimization algorithms, namely, the chameleon swarm algorithm (CSA), marine predators algorithm (MPA), young's double-slit experiment optimizer algorithm (YDSE), and exponential distribution optimizer algorithm (EDO) are used to optimize the hyperparameters, to establish eight prediction models for predicting blast-induced ground vibration. Moreover, the *k*-fold cross-validation method was applied to avoid overfitting and underfitting in the model training process. This study used a variety of evaluation indicators to quantitatively evaluate the eight prediction models developed. The prediction results show that CSA-GPR, MPA-GPR, YDSE-GPR, and EDO-GPR have the same prediction results, and their relative errors are less than 5%. The evaluation results show that MPA and YDSE have better optimization finding ability than CSA and EDO, and GPR has a more significant advantage than SVR in predicting such kind ground vibration.**

***Index Terms*—Blast-induced ground vibration, Support vector machine, Gaussian process regression, Hyperparameters optimization, *K*-fold cross-validation**

I. Introduction

BLASTING is the most economical method of rock breaking [1]. However, the destruction of rocks by blasting is inevitably accompanied by many adverse effects, such as vibration, flying rocks, air shock waves, and dust [2]. Among them, the blasting seismic effect, as the most dominant harmful effect in the blasting process, has a great impact on the slope stability as well as the neighboring buildings (structures) [3]. Therefore, how to effectively control blasting-induced vibration is a key problem that needs to be treated, both from the technical point of view and from the economic aspect. The prediction of such kind vibration can reduce the blasting seismic effect to a large extent [4]. In engineering practice, blasting-induced vibration is usually expressed in terms of vibration intensity, frequency, and duration [5]. The vibration intensity is determined by measuring the peak particle velocity (PPV), so the blasting vibration prediction is usually to predict the PPV [4]. Numerous studies have shown that PPV is impacted by factors like blast feature, explosive properties, and rock mass [6]–[10]. Due to the anisotropy of the rock mass, it is difficult to accurately estimate PPV at a specific distance during the propagation of the blast seismic wave.

In the past decades, researchers have proposed various empirical models based on a large number of field experiments for predicting PPV, and most of these empirical models mainly take into account the effects of the two main factors, namely, the maximum charge of each blast, and the distance in every monitoring point and blast center [11], [12]. Although the empirical models have the advantages of calculation simplicity and convenience, however, due to the simplification of the PPV influencing factors, the prediction precision of the conventional models is limited, and they are only applicable to specific site conditions, which can no longer satisfy the increasing environmental and safety requirements.

To overcome the problem of prediction accuracy of empirical models, many researchers have tried to use machine learning method to predict PPV, such as artificial neural network [13]–[18], extreme learning machine [8], [19], adaptive network[20]–[22], support vector regression (SVR) [3], [9], [23], [24], which have achieved better prediction effects. Although machine learning method was widely used in predicting PPV, there are some shortcomings, especially in the selection of model hyperparameters, where they may ignore superior hyperparameters, resulting in predictive models unable to provide convincing results. In addition, innovative methods for more accurate prediction are needed in the blasting engineering and scientific community.

In this study, the SVR [25], [26] and GPR [27], [28], which have great advantages in solving small sample and nonlinear problems, are used as the main prediction tools to predict PPV, and four optimization algorithms including chameleon swarm algorithm (CSA) [29], marine predators algorithm (MPA) [30], [31] YDSE [32] and exponential distribution optimizer method[33] to search for the optimal hyperparameters of SVM and GPR. After the models (CSA-SVR, MPA-SVR, YDSE-SVR, EDO-SVR, CSA-GPR, MPA- GPR, YDSE- GPR, EDO- GPR) are trained using the PPV datasets collected in the actual blasting projects, the R^2, $RMSE$, MRE and Taylor diagram were applied to assess the model prediction property and to select the best model.

Manuscript received Nov.23, 2023; revised April 29, 2024.

This work was supported in part by the National Natural Science Foundation of China (No.42171437).

Zhang Zihan is a postgraduate student at Jiangxi University of Science and Technology, Ganzhou 341000, China. (Corresponding author to provide phone: 15917050898; e-mail: 13297493934@163.com).

Liu Xiaosheng is a professor at Jiangxi University of Science and Technology, Ganzhou 341000, China. (e-mail: lxs9103@163.com).

Wu Lijun is a doctoral student at University of Science and Technology Beijing, Beijing 100000, China. (e-mail: 369136809@qq.com).

Hu Guangqiu is a blasting engineer at Guangdong Xiyuan Blasting Technology Co., Ltd, Huizhou 516000, China. (e-mail: 597715410@qq.com).

II. METHOD

A. Predictive algorithms used

1) SVR

In 1995, Vapnik proposed a support vector machine (SVM) method , which can be used to treat nonlinear regression (SVR) and classification (SVC) problems [25]. Assuming that the learning sample D consists of input parameters x_i and output parameters y_i, the main task of the SVR lies in constructing a robust mathematical relationship between the x_i and y_i . For nonlinear regression problems, the mathematical relationship between x_i and y_i constructed by the SVR is [34]:

$$T(x) = \langle \alpha, \eta(x) \rangle + \beta \quad (1)$$

where α, β, and $T(x)$ represent the weights, constant values, and high-dimensional feature space, respectively. α and β can be obtained by the following (2):

$$R(C) = \frac{\|\alpha^2\|}{2} + \frac{C}{P}\sum_{i=1}^{N} \rho_\epsilon(y_i, T(x_i)) \quad (2)$$

where $\frac{\|\alpha^2\|}{2}$ is the regularization term; C is the penalty factor, and a larger value of C represents a larger penalty for samples with errors larger than the set standard.

After the introduction of two forward relaxation variables ξ_i and ξ_i^* applied to express the distance of true results and the boundary values, the Largrange function was introduced to convert the computational equation to the dyadic form [25], [26]:

$$\max \quad H(\alpha, \alpha^*) = -\frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(x_i, x_j) + \sum_{i=1}^{N} y_i(\alpha_i - \alpha_i^*) - \varepsilon\sum_{i=1}^{N} y_i(\alpha_i - \alpha_i^*) \quad (3)$$

The above (3) can be converted:

$$T(x) = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)K(x_i, x) + \beta \quad (4)$$

where $K(x_i, x)$ is the kernel function to transform the input variable.

The typical SVR kernel functions include the linear, polynomial, Gaussian kernel function, and the third function has been widely used for its superior mapping performance [26]. Therefore, in this study, the gaussian kernel function was applied to construct a SVR model to address the PPV prediction problem. In this case, the penalty factor C and the parameter σ in the gaussian kernel function are hyperparameters that determine the SVR model property.

2) GPR

The performance and model structure of the GPR [27] model are determined by the mathematical expectation function $m(x)$ and covariance function $k(x, x')$, which can be expressed in terms of the mathematical relationship :

$$f(x) \sim GP(m(x), k(x, x')) \quad (5)$$

where the two kinds function take the form:

$$\begin{cases} m(x) = E[f(x)] \\ k(x, x') = E\{[f(x) - m(x)]\,[f(x') - m(x')]\} \end{cases} \quad (6)$$

Assuming that D represents the training data for a GPR model consisting of input parameters x_i and output parameters y_i, a standard linear regression equation can be constructed based on (7) [28]:

$$y = f(x) + \varepsilon \quad (7)$$

where ε is an independent random variable.

This study chooses the square exponential kernel as the kernel function of the GPR, whose equation is given in (8) [27]. Thus, the hyperparameters of GPR are σ_f^2 and l in the squared exponential kernel and the random variables ε.

$$k(x, x') = \sigma_f^2 e^{-\frac{\|x - x'\|^2}{2l^2}} \quad (8)$$

B. Hyperparameter optimization methods used

Considering the impact of hyperparameters on the performance of SVR and GPR models, it is necessary to determine the optimal hyperparameters to obtain the best model performance. In this study, CSA [29], MPA [30], YDSE [32], and EDO [33] are used to search for the optimal hyperparameters for SVR and GPR.

1) CSA

CSA [29] is a meta-heuristic method proposed by Malik Shehadeh Braik in 2021, which is basically inspired by the chameleons as they roam around trees, deserts, and swamps and search for food sources. The algorithm mathematically models and implements the chameleon's search for hunting food in three stages: searching for prey, eye rotation, capturing prey.

During the prey search phase, the chameleon tracks and finds its prey by constantly changing its position and combining the guidance of its previous position and experience in a mathematical model shown in (9):

$$y_{t+1}^{i,j} = \begin{cases} y_t^{i,j} + p_1 r_2\left(P_t^{i,j} - G_t^j\right) + p_2 r_1\left(G_t^j - y_t^{i,j}\right) & r^i \geq P_p \\ y_t^{i,j} + \mu(r_3(ub - lb) + lb)sgn(rand - 0.5) & r^i < P_p \end{cases} \quad (9)$$

Where: $y_{t+1}^{i,j}$ is the position of chameleon i ; $P_t^{i,j}$ is the optimal position of chameleon i in j-dimension after t-th iteration; G_t^j is the global best position in d j-dimension after t-th iteration; p_1 and p_2 are two positive numbers controlling the chameleon's ability to explore; r_1 , r_2 , r_3 , and r^i are uniformly generated random numbers ; P_p refer to the probability that the chameleon perceives its prey, and takes the value 0.1; $sgn(rand - 0.5)$ can take either 1 or -1; lb and ub are the search region boundary; μ is related to the number of iterations.

In the eye rotation prey discovery stage, the chameleon discovers the exact position of the prey by rotating its eyes through 360°, and this process can be described as follows: first, translating the original position; second, determining the rotation matrix; third, updating the position based on the rotation matrix; then, translating the chameleon to initial position. The mathematical model is shown in (10):

$$y_{t+1}^i = m \times \left(y_t^i - \bar{y}_t^i\right) + \bar{y}_t^i \quad (10)$$

Where $\bar{y}_t^i$ is the pre-rotation position center, and m refer to the rotation matrix of the chameleon rotation.

In the prey capture phase, the closest chameleon to the prey is regarded as the most optimized chameleon that can capture the surrounding prey by extending its tongue to twice its length, i.e., the best result is obtained by performing a localized search. The mathematical model is shown in (11):

$$y_{t+1}^{i,j} = y_t^{i,j} + \left(\left(v_t^{i,j}\right)^2 - \left(v_{t-1}^{i,j}\right)^2\right)/(2a) \quad (11)$$

Where: a is the acceleration rate, which continuously enhances until to 2590 m/s^2; $v_{t-1}^{i,j}$ is the velocity of the chameleon in dimension in the $t-1$st iteration; $v_t^{i,j}$ is the velocity of the chameleon i in the t-th iteration, which is given in (12):

$$v_{t+1}^{i,j} = \omega v_t^{i,j} + c_1\left(G_t^j - y_t^{i,j}\right)r_1 + c_2\left(P_t^{i,j} - y_t^{i,j}\right)r_2 \quad (12)$$

Where: $P_t^{i,j}$ refer to the best position of i in the j-dimension after t-th iteration; G_t^j means the global best position in the j-dimension after t-th iteration; r_1 and r_2 represent random numbers with value of [0,1]; ω is the inertia weight associated with the number of iterations.

2) MPA

MPA [30] is an optimization model proposed by Faramarzi et al. in 2020 based the following theory: the predator searches for prey according to the prey's location information, in which the top predator (the optimal solution) has a higher foraging talent. The optimization process include three stage according to speed ratios: 1stage, the prey with higher speed compared to the predator; 2 stage, the two have almost same speed; 3 stage, the predator is faster.

1 stage, the prey has much higher speed compared to the predator, which by default adopts a state of immobility, and the prey adopts a random movement pattern to search for its own food. This stage is the global search stage and the mathematical model is:

$$\begin{cases} S_i = N_c \otimes (E_i - N_c \otimes P_i), i = 1,\cdots,n \\ P_i = P_{i-1} i + D \bullet N \otimes S_i \end{cases} \quad (13)$$

Where: E_i is the predator population; P_i is the prey population; the number of predators and prey are n; S_i is the step size of the movement; N_c is a vector of random numbers , refer to brownian motion; $N_c \otimes P_i$ is to simulate the movement ; D represents a constant term, which is usually taken to be D = 0.5; and N is random numbers vector.

In the middle one-third stage, the two have the same speed, the predator searches for prey through Brownian motion, and the prey updates its position through Lévy motion. Global search and local optimization go hand in hand. Therefore, in this stage, the population is divided into two equal parts: half of the individuals are used for local optimization; and the other half is used for global search.

The mathematical model of prey local optimization search is:

$$\begin{cases} S_i = N_L \otimes (E_i - N_L \otimes P_i), i = 1,\cdots,\frac{n}{2} \\ P_i = P_{i-1} i + D \bullet N \otimes S_i \end{cases} \quad (14)$$

Where N_L refer to a vector of random numbers, denoting the lévy motion.

The mathematical model of prey local optimization search is:

$$\begin{cases} S_i = N_C \otimes (N_C \otimes E_i - P_i), i = \frac{n}{2},\cdots,n \\ P_i = E_i + D \bullet C_F \otimes S_i \end{cases} \quad (15)$$

Where C_F denotes the adaptive parameter of step size.

In the second third stage, predator moves faster, and the predator searches for the prey by lévy motion. This stage is the local optimization search stage, and the mathematical model is:

$$\begin{cases} S_i = N_L \otimes (N_L \otimes E_i - P_i), i = 1,\cdots,n \\ P_i = E_i + D \bullet C_F \otimes S_i \end{cases} \quad (16)$$

In addition, at the end of each iteration, the MPA avoids falling into a local optimum by making the predator make longer jumps using the fish aggregation device effects (FADs). Its mathematical model is:

If $r \le F_{ADs}$:

$$\vec{P}_i = \vec{P}_{i-1} + C_F[\vec{X}_{min} + \vec{R} \otimes (\vec{X}_{max} - \vec{X}_{min})] \otimes \vec{U} \quad (17)$$

If $r > F_{ADs}$:

$$\vec{P}_i = \vec{P}_{i-1} + [F_{ADs}(1-r) + r](\vec{P}_{r_1} - \vec{P}_{r_2}) \quad (18)$$

Where: F_{ADs} is the probability of effect, usually taken as 0.2; $\vec{U}$ represent a binary vector containing arrays 0 and 1, if the array below 0.2, the array changed to 0, or else it changed to 1; r is the number of random numbers in [0, 1]; r_1 and r_2 are the random indexes of the preys, respectively; $\vec{X}_{min}$ and $\vec{X}_{max}$ are the minimum and ultimate values of the same dimension in the prey population, respectively.

3) YDSE

YDSE [32] was proposed by Mohamed et al. in 2023 as a metaheuristic method to solve the global and constrained issues based on the principles obtained from Young's double-slit experiment. This method can effectively solve the optimization problems with various benchmarks. YDSE include 3 steps: first, initialization; second, updating of traveling waves and third, destructive and constructive interference. The math model of it is as follows.

That experiment begins by projecting a monochromatic light wave source (S) into a barrier with two nearby slits. Hence, the first step is to create an initial S light consisting of n monochromatic light waves:

$$S^i = lb + rand(0,1) \times (ub - lb) \quad (19)$$

Where: S^i is the variable of the i-th monochromatic wave; lb and ub are the boundaries of space.

After the monochromatic wave passes through the slit barrier, according to Huygens' principle, the monochromatic wave will propagate to various directions. Therefore, the wave source and wave center will be updated. Nevertheless, YDSE assumes that the number of points on the wavefronts flowing out of the two slits is equal to n. Then, n points can be calculated for the wavefronts of the two slits by (20) and (21).

$$FS^i = S^i + L \times rand(-1,1) \times (S^{mean} - S^i) \quad (20)$$

$$SS^i = S^i - L \times rand(-1,1) \times (S^{mean} - S^i) \quad (21)$$

Where: FS^i represens the point i produced from the first slit; SS^i refer to the point i generated from the second slit; L represen the distance in light source and the projection screen; S^{mean} is the average value of the current population S.

Waves traveling from two slits have different travel distance as waves traveling from one slit. In addition, the two waves will produce bright streaks (CI) and dark streaks (DI) on screen after mutual constructive and destructive interference. Hence, to simulate the interference behavior of CI and DI and their paths to reach the screen, the YDSE updates the position using (22).

$$X^i = \frac{FS^i + SS^i}{2} + L \quad (22)$$

Where L is the path difference between FS^i and SS^i, which is calculated from the order of the fringes formed when the wave reaches the screen. For fringes of zero and even orders, constructive interference occurs. L is calculated by (23):

$$L = \begin{cases} 0 & if\ CI\ occurs\ at\ m = 0 \\ (2m+1) \times \frac{\lambda}{2} & if\ DI \text{ occurs when } m \text{ is odd} \\ m\lambda & if\ CI \text{ occurs when } m \text{ is even} \end{cases} \quad (23)$$

Where m is the order of the stripes and λ is the wavelength.

Constructive interference correspond to the superposition of two waves and their mutual enhancement, producing a wave with higher amplitude compared to former interfering

wave. Moreover, the wave amplitude are updated in every iteration, the update formula is as follows:

$$A_{t+1}^{b} = 0.5 \times (1 + \sqrt{|1 - (t \times \cosh(\pi/t)/T)^2|}) \quad (24)$$

Where: A_{t+1}^{b} is the average wave amplitude at CI at $t+1$-th iteration; t and T are the current and maximum number of iterations, respectively.

Destructive interference is a phenomenon that two waves are superimposed and cancel each other. Moreover, the wave amplitude is updated in every iteration, the update formula is as follows:

$$A_{t+1}^{d} = \delta \times tanh^{-1}(-\frac{t}{T} + 1) \quad (25)$$

Where A_{t+1}^{d} is the average wave amplitude at DI at $t+1$-th iteration, and δ is a constant with a value of 0.38.

Both CI and DI denote solutions, moreover, the central stripe correspond to the best solution. Under condition of the destructive interference appeared at an odd order, the fit value of the candidate solutions in the dark region will be lower than that of the candidate solutions in the light region, so searching promising positions in dark region first can enhance the optimization performance of YDSE optimizer. Hance, the position update equation for the destructive interference DI region is as follows:

$$X_{t+1}^{m-odd} = X_{t}^{m-odd} - (r \times A_{t+1}^{d} \times Int_{t+1}^{m-odd} \times X_{t}^{m-odd} - Z \times X_{t}^{best}) \quad (26)$$

$$Z = \frac{t^{2r-1}}{H} \quad (27)$$

Where: X_{t+1}^{m-odd} is DI at $t+1$-th iteration; X_{t}^{m-odd} is DI at t-th iteration; X_{t}^{best} is DI that is best at t-th iteration; r is a random number with value in [0, 1]; A_{t+1}^{d} is calculated by (25); Z is a test vector of dimension d; H represent a random number in [-1, 1]; Int_{t+1}^{m-odd} is the light intensity of DI at $t+1$-th iteration, calculated by (28).

$$Int_{t+1}^{m-odd} = Int_{t+1}^{max} \times cos^2(\frac{\pi d}{\lambda L} \times l_{t+1}^{d}) \quad (28)$$

$$l_{t+1}^{d} = \frac{\lambda L}{d} \times (m + 0.5) \quad (29)$$

$$Int_{t+1}^{max} = C \times t/T \quad (30)$$

Where l_{t+1}^{d} is is the distance in central stripe and DI, and Int_{t+1}^{max} is the maximum light intensity detected by the central stripe.

When the order is even, YDSE finds promising candidate solutions in region CI, resulting in constructive interference. Therefore, YDSE will try to search all the promising regions in region CI in the local search phase with the following position update equation:

$$X_{t+1}^{m-even} = X_{t}^{m-even} - ((1 - r) \times A_{t+1}^{b} \times Int_{t+1}^{m-even} \times X_{t}^{m-even} + r \times l \quad (31)$$

Where: X_{t+1}^{m-even} is CI at $t+1$-th iteration; X_{t}^{m-even} is CI at t-th iteration; r is a random number in [0, 1]; A_{t+1}^{b} is calculated by (24); l is the difference in two selected stripes; Int_{t+1}^{m-even} is the light intensity of CI at $t+1$-th iteration, calculated by (32).

$$Int_{t+1}^{m-even} = Int_{t+1}^{max} \times cos^2(\frac{\pi d}{\lambda L} \times l_{t+1}^{b}) \quad (32)$$

Where l_{t+1}^{b} refer to the distance in central stripe and CI.

In addition, the position of the center stripe is updated with (33).

$$X_{t+1}^{m-z} = X_{t}^{best} + (A_{t+1}^{b} \times Int_{t+1}^{max} \times X_{t}^{m-z} - r \times Z \times X_{t}^{rb}) \quad (33)$$

Where: X_{t+1}^{m-z} is the position of the central stripe at $t+1$-th iteration; X_{t}^{m-z} refer to central stripe position at t-th iteration; X_{t}^{rb} is rb based on a randomly chosen CI, and rb is even.

4) EDO

EDO [33] is a meta-heuristic method proposed by Mohamed Abdel-Basset et al. in 2023, which is based on the relevant distribution model and the memoryless property of the exponential distribution. It consists of both local and global search schemes. In the local search phase, EDO utilizes the memoryless property, the exponential variance between the guided solution and the random variable to update present solution. In the global search phase, EDO randomly selects two solutions from the original population to update present solution.

In the local search phase, EDO follows the criterion that the region near optimized solution is expected to find a globally optimal solution, and introduces the concept of a guided solution, which can effectively prevent local optimum problem appeared, it is calculated by (34):

$$X_{t}^{g} = \frac{X_{t}^{b1} + X_{t}^{b2} + X_{t}^{b3}}{3} \quad (34)$$

Where: X_{t}^{g} is the guided solution at t-th iteration; X_{t}^{b1},X_{t}^{b2} and X_{t}^{b3} are the first three optimal solutions at t-th iteration.

EDO uses an optimization model obeying an exponential distribution to update the current new solution, which is calculated by (35):

$$V_{t+1}^{i} = \begin{cases} a \times (ml_{t}^{i} - \sigma^2) + b \times X_{t}^{g} & if\ X_{t}^{i} = ml_{t}^{i} \\ b \times (ml_{t}^{i} - \sigma^2) + \log(\phi) \times X_{t}^{i} & otherwise \end{cases} \quad (35)$$

$$a = (2 \times rand(-1,1) - 1)^{10} \quad (36)$$

$$b = (2 \times rand(-1,1) - 1)^{5} \quad (37)$$

$$\phi = rand(0,1) \quad (38)$$

Where: V_{t+1}^{i} is the solution at $t+1$-th iteration; ml_{t}^{i} is the solution; X_{t}^{i} is the best set of solutions after t-th iteration, with the same number of solutions as initialized.

In the global search phase, EDO identifies regions o that are prone to have globally optimal solutions and computes a new solution using two randomly selected solutions. It is calculated by (39):

$$V_{t+1}^{i} = X_{t}^{i} - M_{t} + (c \times Z_1 + (1 - c) \times Z_2) \quad (39)$$

$$c = \frac{1-t}{T} \times (2 \times rand(-1,1) - 1) \quad (40)$$

$$Z_1 = M_{t} + X_{t}^{r1} - X_{t}^{r2} \quad (41)$$

$$Z_2 = M_{t} - X_{t}^{r1} + X_{t}^{r2} \quad (42)$$

Where: M_t refer to the mean value of X_{t}^{i}; c associated to the number of current iterations, which is calculated using (40); T is the maximum number of iteration;

EDO decides to perform a local or global search by introducing the parameter $\alpha = rand(0,1)$. $if\ \alpha < 0.5$, EDO performs a local search. $if\ \alpha \geq 0.5$, EDO performs a global search.

III. Dataset

The data in presen research were collected from an actual blasting project in the petrochemical zone of the Daya Bay Development Zone in Huizhou, China, the detailed location of which is shown in Fig.1. The area is bounded by mountains to the north and east, the S30 Huishen coastal highway is 75m to the south, and there is an east-west national oil storage pipeline 75m away. Therefore, the

blasting-induced vibration needs to be strictly controlled to avoid damage to the pipeline.

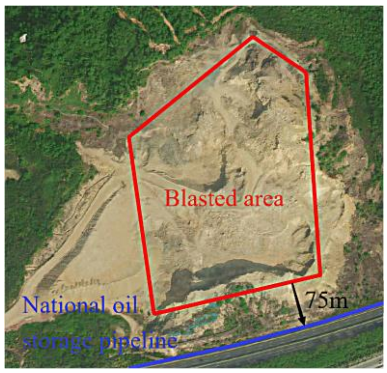

Fig.1 Blasting project location

Many scholars have found that the maximum charge (Q) of each blast, the distance (D), and elevation difference (H) between each monitoring point and the center of each blast are the main factors affecting the PPV [6]–[10]. To effectively controlling the PPV , the present research takes the Q, D, H as the input parameters, the PPV as the output result to establish the mathematical model, and then optimizes Q based on this model. This study collects 72 sets of data, in which the first 54 sets were applied as the training set, and the remaining 18 sets of data are applied to test the performance of the model. TABLE I shows the data sets, and Fig.2 shows the scatter matrix plot of the training set.

TABLE I
DATA SETS

No.	Q (m)	D (m)	H (m)	PPV (cm/s)	No.	Q (m)	D (m)	H (m)	PPV (cm/s)
1	84	80	3.4	4.85	2	92	90	2.3	4.41
3	80	95	2.9	3.89	4	88	94	1.8	4.03
5	76	99	2.0	3.65	6	87	86	4.6	4.43
7	88	99	1.3	3.71	8	80	79	5.1	4.83
9	88	101	1.0	3.68	10	87	77	4.5	5.37
11	80	99	1.5	3.82	12	84	102	3.1	3.30
13	64	76	6.2	4.39	14	84	75	4.1	5.49
15	80	71	4.6	6.22	16	92	95	4.7	4.14
17	64	78	1.3	4.58	18	92	87	1.5	4.77
19	56	79	2.3	4.14	20	84	84	2.1	4.70
21	60	95	4.0	3.28	22	80	85	2.5	4.41
23	64	88	4.2	3.77	24	76	94	4.4	3.59
25	84	89	2.1	4.17	26	87	101	1.5	3.65
27	60	78	4.6	4.51	28	68	73	3.2	5.37
29	68	86	3.9	4.06	30	83	91	6.4	3.90
31	83	83	1.5	5.17	32	84	74	1.0	5.89
33	64	86	1.9	4.18	34	87	95	1.0	4.27
35	88	77	3.2	5.43	36	80	81	1.9	4.90
37	76	89	3.7	4.06	38	68	83	5.3	4.01
39	76	83	3.1	4.69	40	64	78	1.3	4.70
41	76	72	2.9	5.52	42	84	93	5.4	3.76
43	83	96	4.8	3.73	44	80	97	2.8	3.63
45	88	72	6.4	5.87	46	92	79	2.6	5.73
47	87	87	2.8	4.76	48	56	96	4.0	3.03
49	92	75	3.2	5.83	50	80	90	4.2	4.01
51	88	72	2.1	6.10	52	88	81	1.7	5.44
53	87	85	2.9	4.94	54	68	92	6.8	3.40
55	76	102	4.3	3.29	56	84	79	4.5	5.09
57	60	83	4.0	3.85	58	80	79	4.8	5.07
59	87	86	4.5	4.65	60	88	98	5.6	3.73
61	83	73	6.1	5.69	62	88	72	4.2	6.26
63	68	88	6.7	3.67	64	83	76	2.5	5.54
65	64	68	2.6	5.59	66	76	91	1.2	4.02
67	64	81	3.5	4.28	68	92	88	3.0	4.62
69	64	76	2.4	4.88	70	60	82	3.4	4.08
71	80	88	1.8	4.32	72	68	93	5.6	3.31

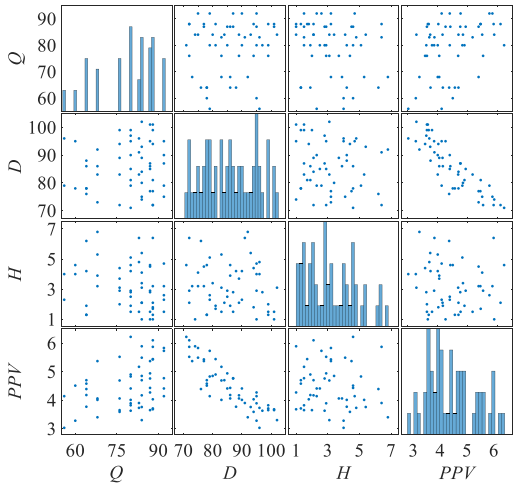

Fig.2 Scatter matrix plot of the training set

IV. DEVELOPMENT MODELS

A. Data pre-processing

Due to the different orders of magnitude of the training parameters, it will cause a loss of training accuracy during training. Most studies use the normalization method to preprocess the data, making it highly susceptible to the ultimate and minimum values and causing bias in the training data. In this study, the data are preprocessed using the standardization method, which is calculated as in (43). Fig.3 shows the box plot .

$$X_i = \frac{x_i - \bar{x}}{\sqrt{\frac{1}{n-1}\sum_{j=1}^{n}(x_j - \bar{x})^2}} \tag{43}$$

Where X refer to the standardized value, x represen detection value, $\bar{x}$ refer to the mean value of x.

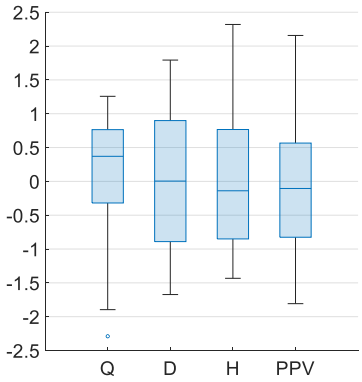

Fig.3 Box plot of the training set

B. Determination of optimal hyperparameters

The hyperparameters can obviously impact the model prediction result. If the hyperparameters are not properly selected, it is very easy for the model to suffer from two problems: the performance of the training set is obviously better compared to the test set (overfitting phenomenon), or the test set performance is obviously superior than the training set (underfitting phenomenon). To avoid the overfitting and underfitting phenomenon of the model, this study uses the k-fold cross-validation algorithm to internally validate the predictive performance of model during the

optimization algorithm to determine the optimal hyperparameters of the model, thus effectively avoiding the interference of the data outliers on the model predictive property. Fig.4 shows the construction and evaluation process of the *PPV* prediction model. In addition, for CSA, MPA, YDSE, and EDO, the initial population is 100 and the largest iterations is 100. Based on the research experience, the search range of hyperparameters of SVM and GPR are set to [0.1, 100]. TABLE II shows the hyperparameters of SVM and GPR obtained after optimization by CSA, MPA, YDSE, and EDO.

TABLE II
HYPERPARAMETERS FOR SVM AND GPR

		CSA	MPA	YDSE	EDO
SVR	C	66.313	30.615	27.094	55.476
	σ	0.80428	0.33052	0.33676	0.80464
GPR	σ_f^2	13.856	16.525	34.804	23.591
	l	19.391	15.631	12.216	24.895
	ε	14.221	34.097	10.876	28.852

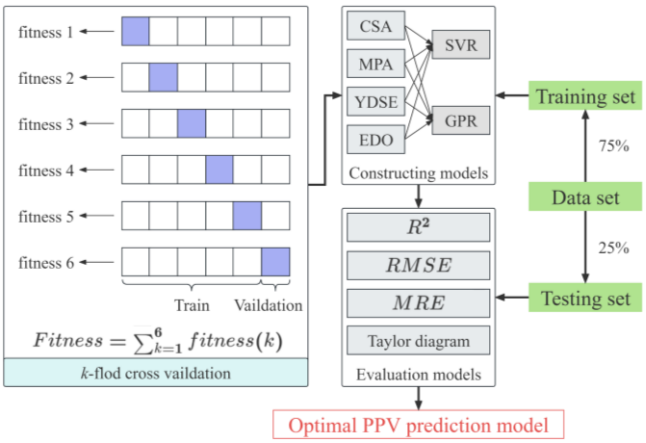

Fig.4 *PPV* prediction models constructing and evaluating process

V. RESULTS AND DISCUSSION

The *PPV* prediction model obtained by training is used to predict the test data in TABLE I and the prediction results can be seen from TABLE III. According to TABLE III, the two models CSA-SVR and EDO-SVR have the same prediction results, and the four models CSA-GPR, MPA-GPR, YDSE-GPR, and EDO-GPR have the same prediction results.

TABLE III
MODEL PREDICTION RESULTS

No.	CSA - SVR	MPA - SVR	YDSE - SVR	EDO - SVR	CSA - GPR	MPA - GPR	YDSE - GPR	EDO - GPR
55	3.81	3.42	3.43	3.81	3.14	3.14	3.14	3.14
56	4.98	4.94	4.95	4.98	5.13	5.13	5.13	5.13
57	4.15	4.04	4.04	4.15	3.99	3.99	3.99	3.99
58	4.92	4.86	4.88	4.92	4.97	4.97	4.97	4.97
59	4.50	4.55	4.55	4.50	4.57	4.57	4.57	4.57
60	4.09	3.84	3.85	4.09	3.56	3.56	3.56	3.56
61	5.80	5.83	5.83	5.80	5.57	5.57	5.57	5.57
62	5.79	5.93	5.92	5.79	6.00	6.00	6.00	6.00
63	3.60	3.53	3.54	3.60	3.67	3.67	3.67	3.67
64	5.46	5.34	5.34	5.46	5.52	5.52	5.52	5.52
65	4.88	5.05	5.04	4.88	5.52	5.52	5.52	5.52
66	4.30	4.33	4.34	4.30	4.12	4.12	4.12	4.12
67	4.51	4.56	4.56	4.51	4.32	4.32	4.32	4.32
68	4.85	4.92	4.91	4.85	4.65	4.65	4.65	4.65
69	4.80	4.86	4.85	4.80	4.82	4.82	4.82	4.82
70	4.23	4.23	4.22	4.23	4.11	4.11	4.11	4.11
71	4.31	4.34	4.35	4.31	4.40	4.40	4.40	4.40
72	3.48	3.25	3.26	3.48	3.40	3.40	3.40	3.40

In present research, the prediction property of eight models is quantitatively assessed based on the R^2, $RMSE$, MRE, and Taylor diagram as evaluation metrics, where the Taylor diagram is a evaluation metric of model via the R^2, $RMSE$, and STD of obtained results. Considering the same prediction results for CSA-SVR and EDO-SVR, only the prediction results of CSA-SVR are evaluated. Considering that the predictions of CSA-GPR, MPA-GPR, YDSE-GPR, and EDO-GPR are the same, only the predictions of CSA-GPR are evaluated.

Fig.5 shows the relative error (RE) of the prediction results, and Fig.6 shows the frequency of RE in the prediction results. TABLE IV shows the R^2, $RMSE$, and MRE of the prediction models. Fig.7 shows the Taylor diagram of prediction results. It was worth to note that the reference point in the Taylor diagram represent the best model with maximum R^2, minimum STD, and minimum $RMSE$.

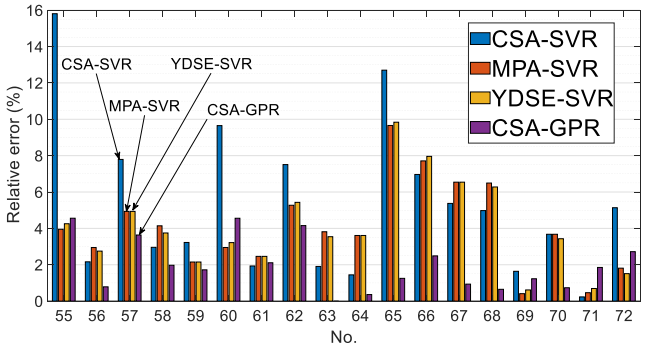

Fig.5 The RE of the prediction results

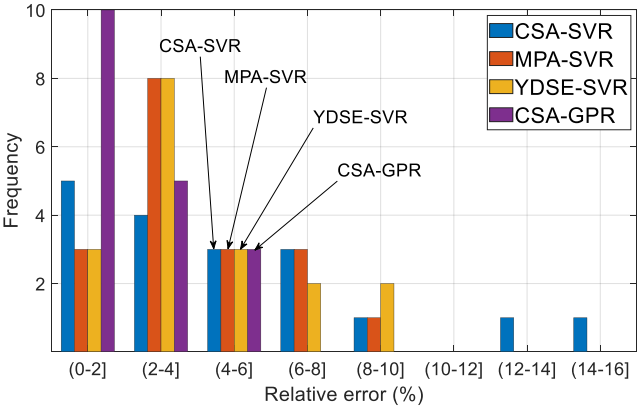

Fig.6 The frequency of relative errors in prediction results

TABLE IV
PERFORMANCE COMPARISONS OF PREDICTION MODELS

	CSA-SVR	MPA-SVR	YDSE-SVR	CSA-GPR
R^2	87.86	92.80	92.74	98.36
$RMSE$	0.29	0.23	0.23	0.11
MRE	5.28	4.06	4.05	1.98

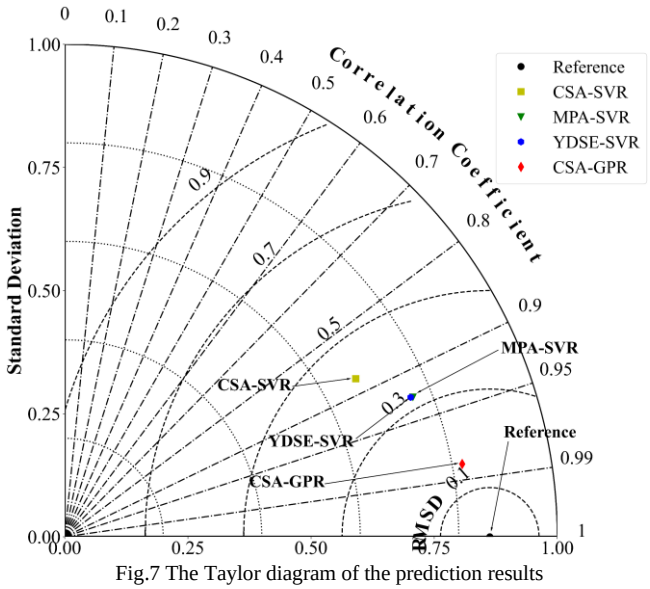

Fig.7 The Taylor diagram of the prediction results

From Fig.5, the RE of the CSA-SVR and EDO-SVR models in predicting the results of No.55 and No.65 are greater than 10%, and the RE of the remaining other models in predicting the results are less than 10%. The RE of the CSA-GPR, MPA-GPR, YDSE-GPR, and EDO-GPR models in predicting the results are less than 5%, which indicates that the GPR has a more significant advantage over SVR in predicting PPV. From Fig. 6, the RE of prediction results of CSA-GPR, MPA-GPR, YDSE-GPR, and EDO-GPR models are between 0 to 2% in eight cases, and between 2% to 4% in five cases. The RE of prediction results of MPA-SVR and YDSE-SVR models are between 0 to 2% in only three cases, and between 2% to 4% in eight cases, the RE of the remaining prediction results are all greater than 4%. The frequency of the RE of the CSA-SVR and EDO-SVR models shows a decreasing trend as the RE increases, and there are two predictions with RE greater than 12%. TABLE IV shows that among the four models CSA-SVR, MPA-SVR, YDSE-SVR, and EDO-SVR, the prediction performance of MPA-SVR and YDSE-SVR on PPV is similar and better than CSA-SVR and EDO-SVR, indicating that the optimization-seeking ability of MPA and YDSE on the hyperparameters of SVR is better than that of CSA and EDO. The R^2, $RMSE$, and MRE of CSA-GPR, MPA-GPR, YDSE-GPR, and EDO-GPR prediction results are 98.36, 0.11, and 1.98, respectively, which are much better than those of the optimized SVR, indicating that the GPR is more competitive in PPV prediction. It is obvious from the Taylor diagram that the optimized GPR has the best prediction performance for PPV, again proving the above conclusion.

VI. Conclusion

Blasting-induced environmental problems are becoming more and more prominent, among which the blasting seismic effect is the most important harmful problem in the blasting process, and a high-precision PPV prediction model is a feasible solution to control the blasting seismic effect. To improve the PPV prediction accuracy, eight new intelligent prediction models with high accuracy (i.e., CSA-SVR, MPA-SVR, YDSE-SVR, EDO-SVR, CSA-GPR, MPA-GPR, YDSE-GPR, and EDO-GPR) are developed in this study. Firstly, two prediction tools (SVR and GPR) and four hyperparameter optimization algorithms (CSA, MPA, YDSE, and EDO) were illustrated detailly. Subsequently, the process of model development is given. Finally, the developed models are tested using test data, and the models property is evaluated using R^2, $RMSE$, MRE, and Taylor diagram. The main conclusions are given below:

(1) Compared to CSA and EDO, MPA and YDSE are better at optimizing SVR hyperparameters.

(2) Compared to the optimized SVR, the optimized GPR is more accurate in predicting the PPV.

References

[1] Q. Guo, S. Yang, Y. Wang, and H. Zhang, "Size measurement of blasted rock fragments based on FRRSnet +," *Measurement*, vol. 218, pp. 1–16, 2023, doi: 10.1016/j.measurement.2023.113207.

[2] D. J. Armaghani, M. Hajihassani, E. T. Mohamad, A. Marto, and S. A. Noorani, "Blasting-induced flyrock and ground vibration prediction through an expert artificial neural network based on particle swarm optimization," *Arab. J. Geosci.*, vol. 7, no. 12, pp. 5383–5396, 2014, doi: 10.1007/s12517-013-1174-0.

[3] M. Hasanipanah, M. Monjezi, A. Shahnazar, D. Jahed Armaghani, and A. Farazmand, "Feasibility of indirect determination of blast induced ground vibration based on support vector machine," *Measurement*, vol. 75, pp. 289–297, 2015, doi: 10.1016/j.measurement.2015.07.019.

[4] Y. Zhang, H. He, M. Khandelwal, K. Du, and J. Zhou, "Knowledge mapping of research progress in blast-induced ground vibration from 1990 to 2022 using CiteSpace-based scientometric analysis," *Environ. Sci. Pollut. Res.*, vol. 30, no. 47, pp. 103534–103555, 2023, doi: 10.1007/s11356-023-29712-1.

[5] A. K. Gorai, V. K. Himanshu, and C. Santi, "Development of ANN-Based Universal Predictor for Prediction of Blast-Induced Vibration Indicators and its Performance Comparison with Existing Empirical Models," *Mining, Metall. Explor.*, vol. 38, no. 5, pp. 2021–2036, 2021, doi: 10.1007/s42461-021-00449-0.

[6] Q. Guo, S. Yang, Y. Wang, and Z. Xiang, "Prediction research for blasting peak particle velocity based on random GA-BP network group," *Arab. J. Geosci.*, vol. 15, no. 15, pp. 1–16, 2022, doi: 10.1007/s12517-022-10615-3.

[7] C. K. Arthur, V. A. Temeng, and Y. Y. Ziggah, "Novel approach to predicting blast-induced ground vibration using Gaussian process regression," *Eng. Comput.*, vol. 36, no. 1, pp. 29–42, 2020, doi: 10.1007/s00366-018-0686-3.

[8] D. Jahed Armaghani, D. Kumar, P. Samui, M. Hasanipanah, and B. Roy, "A novel approach for forecasting of ground vibrations resulting from blasting: modified particle swarm optimization coupled extreme learning machine," *Eng. Comput.*, vol. 30, no. 2, pp. 1849–1863, 2020, doi: 10.1007/s00366-020-00997-x.

[9] X. Ding, M. Hasanipanah, H. Nikafshan Rad, and W. Zhou, "Predicting the blast-induced vibration velocity using a bagged support vector regression optimized with firefly algorithm," *Eng. Comput.*, vol. 37, no. 3, pp. 2273–2284, 2021, doi: 10.1007/s00366-020-00937-9.

[10] H. Fattahi and M. Hasanipanah, "Prediction of Blast-Induced Ground Vibration in a Mine Using Relevance Vector Regression Optimized by Metaheuristic Algorithms," *Nat. Resour. Res.*, vol. 30, no. 2, pp. 1849–1863, 2021, doi: 10.1007/s11053-020-09764-7.

[11] P. Pal Roy, "Vibration control in an opencast mine based on improved blast vibration predictors," *Min. Sci. Technol.*, vol. 12, no. 2, pp. 157–165, 1991, doi: 10.1016/0167-9031(91)91642-U.

[12] R. Rai and T. N. Singh, "A new predictor for ground vibration prediction and its comparison with other predictors," *Indian J. Eng. Mater. Sci.*, vol. 11, no. 3, pp. 178–184, 2004.

[13] G. Paneiro, F. O. Durão, M. Costa e Silva, and P. A. Bernardo, "Neural network approach based on a bilevel optimization for the prediction of underground blast-induced ground vibration amplitudes," *Neural Comput. Appl.*, vol. 32, no. 10, pp. 5975–5987, 2020, doi: 10.1007/s00521-019-04083-2.

[14] K. Taheri, M. Hasanipanah, S. B. Golzar, and M. Z. A. Majid, "A hybrid artificial bee colony algorithm-artificial neural network for forecasting the blast-produced ground vibration," *Eng. Comput.*,

vol. 33, no. 3, pp. 689–700, 2017, doi: 10.1007/s00366-016-0497-3.

[15] M. Hajihassani, D. Jahed Armaghani, A. Marto, and E. Tonnizam Mohamad, "Vibrations au sol prédiction dans quarry dynamitage à travers un réseau neural artificiel optimisé par une concurrence impérialiste algorithme," *Bull. Eng. Geol. Environ.*, vol. 74, no. 3, pp. 873–886, 2015, doi: 10.1007/s10064-014-0657-x.

[16] K. Ram Chandar, V. R. Sastry, and C. Hegde, "A Critical Comparison of Regression Models and Artificial Neural Networks to Predict Ground Vibrations," *Geotech. Geol. Eng.*, vol. 35, no. 2, pp. 573–583, 2017, doi: 10.1007/s10706-016-0126-3.

[17] H. Nguyen, C. Drebenstedt, X. N. Bui, and D. T. Bui, "Prediction of Blast-Induced Ground Vibration in an Open-Pit Mine by a Novel Hybrid Model Based on Clustering and Artificial Neural Network," *Nat. Resour. Res.*, vol. 29, no. 2, pp. 691–709, 2020, doi: 10.1007/s11053-019-09470-z.

[18] M. Amiri, H. Bakhshandeh Amnieh, M. Hasanipanah, and L. Mohammad Khanli, "A new combination of artificial neural network and K-nearest neighbors models to predict blast-induced ground vibration and air-overpressure," *Eng. Comput.*, vol. 32, no. 4, pp. 631–644, 2016, doi: 10.1007/s00366-016-0442-5.

[19] H. Wei, J. Chen, J. Zhu, X. Yang, and H. Chu, "A novel algorithm of Nested-ELM for predicting blasting vibration," *Eng. Comput.*, vol. 38, no. 2, pp. 1241–1256, 2022, doi: 10.1007/s00366-020-01082-z.

[20] W. Zhu, H. Nikafshan Rad, and M. Hasanipanah, "A chaos recurrent ANFIS optimized by PSO to predict ground vibration generated in rock blasting," *Appl. Soft Comput.*, vol. 108, p. 107434, 2021, doi: 10.1016/j.asoc.2021.107434.

[21] A. Shahnazar, H. Nikafshan Rad, M. Hasanipanah, M. M. Tahir, D. Jahed Armaghani, and M. Ghoroqi, "A new developed approach for the prediction of ground vibration using a hybrid PSO-optimized ANFIS-based model," *Environ. Earth Sci.*, vol. 76, no. 15, 2017, doi: 10.1007/s12665-017-6864-6.

[22] D. J. Armaghani, E. Momeni, S. V. A. N. K. Abad, and M. Khandelwal, "Feasibility of ANFIS model for prediction of ground vibrations resulting from quarry blasting," *Environ. Earth Sci.*, vol. 74, no. 4, pp. 2845–2860, 2015, doi: 10.1007/s12665-015-4305-y.

[23] M. Khandelwal, "Blast-induced ground vibration prediction using support vector machine," *Eng. Comput.*, vol. 27, no. 3, pp. 193–200, 2011, doi: 10.1007/s00366-010-0190-x.

[24] G. Xu and X. Wang, "Support vector regression optimized by black widow optimization algorithm combining with feature selection by MARS for mining blast vibration prediction," *Measurement*, vol. 218, p. 113106, 2023, doi: 10.1016/j.measurement.2023.113106.

[25] R. Collobert and S. Bengio, "SVMTorch: Support Vector Machines for large-scale regression problems," *J. Mach. Learn. Res.*, vol. 1, no. 2, pp. 143–160, 2001.

[26] C. C. Chang and C. J. Lin, "LIBSVM: A Library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011, doi: 10.1145/1961189.1961199.

[27] J. Quinonero-Candela and C. E. Rasmussen, "A unifying view of sparse approxima gaussian process regression," *J. Mach. Learn. Res.*, vol. 6, pp. 1939–1959, 2005.

[28] E. Schulz, M. Speekenbrink, and A. Krause, "A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions," *J. Math. Psychol.*, vol. 85, pp. 1–16, 2018, doi: 10.1016/j.jmp.2018.03.001.

[29] M. S. Braik, "Chameleon Swarm Algorithm: A bio-inspired optimizer for solving engineering design problems," *Expert Syst. Appl.*, vol. 174, pp. 1–25, 2021, doi: 10.1016/j.eswa.2021.114685.

[30] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine Predators Algorithm: A nature-inspired metaheuristic," *Expert Syst. Appl.*, vol. 152, pp. 1–48, 2020, doi: 10.1016/j.eswa.2020.113377.

[31] P D Kusuma, and D Adiputra, "Hybrid Marine Predator Algorithm and Hide Object Game Optimization," *Engineering Letters*, vol. 31, no.1, pp262-270, 2023.

[32] M. Abdel-Basset, D. El-Shahat, M. Jameel, and M. Abouhawwash, "Young's double-slit experiment optimizer: A novel metaheuristic optimization algorithm for global and constraint optimization problems," *Comput. Methods Appl. Mech. Eng.*, vol. 403, pp. 1–63, 2023, doi: 10.1016/j.cma.2022.115652.

[33] M. Abdel-Basset, D. El-Shahat, M. Jameel, and M. Abouhawwash, "Exponential distribution optimizer (EDO): a novel math-inspired algorithm for global optimization and engineering problems," *Artif. Intell. Rev.*, vol. 56, pp. 9329–9400, 2023, doi: 10.1007/s10462-023-10403-9.

[34] S Ding, Y Xu, T Sun, J Yu, Lg Wang, and R Zhu, "Roadside Unit Visibility Prediction Method Based on SVR," *Engineering Letters*, vol. 31, no.1, pp419-434, 2023.