Parameter Optimization of Support Vector Regression Using Henry Gas Solubility Optimization Algorithm

WEIDONG CAO, **XIN LIU**, (Student Member, IEEE),
AND JIANJUN NI, (Senior Member, IEEE)

College of IoT Engineering, Hohai University, Changzhou 213022, China

Corresponding author: Xin Liu (15B904027@hit.edu.cn)

ABSTRACT Support vector regression (SVR) is one of the most powerful and widely used machine learning algorithms regarding prediction. The kernel type, penalty factor and other parameters influence the efficiency and performance of SVR deeply. The optimization of these parameters is held a hot issue. In this work, we propose a SVR based prediction approach using henry gas solubility optimization algorithm (HGSO), which is a recent meta-heuristic algorithm inspired by Henry's law. First, SVR parameters are randomly generated in some certain ranges to form parameter population. Second, the prediction accuracies (PAs) are obtained using the population and SVR. Thirdly, the population and optimal SVR parameters are updated via PAs and HGSO. We repeat the second and third steps until the cut-off conditions are met. Ten low- and high-dimensional benchmark data sets are utilized to assess the prediction accuracy, convergence performance and computational complexity of the presented approach and other well-known algorithms. The experimental results reveal that our approach has the optimum comprehensive performance.

INDEX TERMS Machine learning, prediction problems, support vector regression (SVR), henry gas solubility optimization (HGSO).

I. INTRODUCTION

As a popular and powerful machine learning algorithm, support vector machine (SVM) has widespread applications. It was first proposed by Vapnik [1]. The invention of software package LIBSVM [2] makes SVM more concerned. Many scholars have noticed the superiorities of SVM, such as excellent generalization performance, high ability to model complex and non-linear relations, etc [3]. Support vector regression (SVR) and support vector classification (SVC) are the two major components of SVM. SVR is the focus of this work. It has been applied in many fields such as stock market price forecasting [4], bearing health monitoring [5], electric load forecasting [6], river stage prediction [7], prediction of global solar radiation [8], predictive model of surface

roughness in lenses precision turning [9], total organic carbon content prediction [10], etc.

Although SVR has great advantages and wide applications, its performance is still affected by the initial setting parameters, such as the kernel type, penalty factor and other parameters in the kernel function. Especially in the processing of high-dimensional data, the effects regarding the generalization performance and accuracy are relatively large. Therefore, searching the optimal parameter set of SVR is a very challenging task. The firefly algorithm (FA) was involved to optimize SVR parameters for the stock market price forecasting [4]. The particle swarm optimization (PSO) was employed to elect parameters of SVR for the prediction of total organic carbon content [10]. The ant lion optimizer (ALO) was adapted to seek for the SVR's optimal parameters for on-line voltage stability assessment [11]. The dragonfly algorithm (DA) was drawn into SVR to obtain the optimal parameters for the prediction and application of

The associate editor coordinating the review of this manuscript and approving it for publication was Zhaojun Li.

porosity [12]. The salp swarm algorithm (SSA) was introduced to optimize SVR for the prediction of long-term temperature effect in structural health monitoring of concrete dams [13]. The fruit fly optimization algorithm (FOA) was involved in SVR for predicting the number of vacant parking spaces after a specific period of time [14]. The genetic algorithm (GA) was applicably administered to determine optimal SVR parameters for forecasting bed load transport rates of three gravel-bed rivers [15].

Three interesting findings can be obtained from the above literatures.

1) The prediction stability of SVR is very excellent. However, the hybrid SVR method combined with other intelligent algorithms loses a lot of stability performance in order to obtain the better prediction accuracy.

2) Almost all the work started from a specific application field. The developed algorithm was not verified in other fields.

3) The convergence performance and computational complexity are rarely discussed and compared.

Motivated by these issues, this work is an attempt to fill the gaps, and makes contributions in the following two areas. First, a novel swarm intelligence optimization approach is sought to optimize SVR parameters and obtain good prediction accuracy, convergence performance and computational complexity with relatively small predictive stability loss. Second, many data sets from various fields are utilized to test the proposed approach.

Henry gas solubility optimization (HGSO) algorithm is an up-to-date meta-heuristic method [16], which was invented by Hashim *et al.* [17], [18] in 2019. It simulates the behavior of Henry's law and imitates the huddling behavior of gas to balance exploitation and exploration in the search space. HGSO algorithm has evaluated on several benchmark data sets and achieved significant superiority against some competitive algorithms. In addition, it has little influence on SVR prediction stability via subsequent experiments.

In this paper, we propose a novel SVR-HGSO model for the first time. HGSO is applied to tune the parameters of SVR for obtaining the high predictive accuracy and stability. First, SVR parameters are randomly generated in some certain ranges to form parameter population. Second, the prediction accuracies (PAs) are obtained using the population and SVR. Thirdly, the population and optimal SVR parameters are updated via PAs and HGSO. The second and third steps are repeated until the cut-off conditions are met. Finally, the optimal SVR parameters are achieved. Ten low- and high-dimensional benchmark data sets are utilized to assess the accuracy, convergence performance and computational complexity of the proposed approach. Simultaneously, the SVR, SVR-FA, SVR-PSO, SVR-ALO, SVR-DA, SVR-SSA and SVR-HHO (harris hawks optimization [19]) algorithms are compared with our approach. The reason for choosing SVR algorithm is to observe the changes of performance on prediction accuracy, convergence stability

and rate. The other hybrid algorithms are selected because they are very similar to the proposed approach and have the outstanding performance and practical applications.

The innovation includes two points. First, HGSO algorithm is involved in SVR parameter optimization. The proposed approach has excellent comprehensive performance against the other competitive algorithms. It has competitive prediction performance, excellent stability and convergence performance. Second, the proposed approach is tested and verified in many fields, such as the airfoil self-noise, air quality, automobile price prediction and so on.

The remainder is arranged as follows. A related literature review is listed in Section II. Section III presents research methodology and analysis methods. The SVR-HGSO integrated approach are developed in this section. Section IV provides the case study and discussion. The conclusion and future work are summarized in Section V.

II. LITERATURE REVIEW

In the literature, the grid search is a traditional method for tuning SVR parameters. It has been proved that some meta-heuristic algorithms are more effective than the grid search [3], [20]. It has become a significant trend that meta-heuristic algorithms are introduced into SVR parameter adjustment. Mimicking ethological, biological, or physics phenomena is the main means of meta-heuristic algorithms in solving optimization problems [21]. The algorithms applied to SVR include GA [15], [22], FA [4], gray wolf optimization (GWO) [23], PSO [10], ALO [11], DA [12], SSA [13], whale optimization algorithm (WOA) [24], [25], elephant herding algorithm (EHO) [26], simulated annealing (SA) [27], etc.

The no free lunch theorem in optimization [28] proves that there is and will never be an algorithm to resolve all optimization problems. Hence, brand new algorithms may have the potential to outperform the present ones on some problems. The HGSO approach is a new hand. However, it has showed amazing optimization performance. Compared with PSO, gravitational search algorithm (GSA) [29], cuckoo search algorithm (CS) [30], GWO, WOA, EHO and SA, it obtained competitive and superior results [17]. These conclusions provide us with the feasibility and rationality of integrating it into SVR parameter optimization, so that we can combine HGSO and SVR to form a new machine learning approach for prediction. The experimental results also show that the proposed approach has excellent comprehensive performance.

III. THE SVR-HGSO APPROACH

In this section, we describe the novel SVR-HGSO hybrid approach. The general idea of the methodology is as follows. a) do the SVR parameter population representation. b) obtain PAs based on the population using fitness functions and SVR. c) update the population and calculate optimal SVR parameters via PAs and HGSO's core operations. We repeat step b and c until the cut-off condition is met. The cut-off condition is generally set to reach the maximum iteration.

In the idea, there are two issues that need special attention [3]. The first is how the SVR parameter population is represented. The second is the selection of evaluation method for PAs. In SVR-HGSO, the discussions of the issues are shown in the following.

1) SVR parameter population representation: The population X is made up of individuals X_i. We express the population as $X = \{X_1, X_2, \cdots, X_n\}$, where n indicates the number of individuals in the population. The attributes of X_i are determined via SVR parameters. ϵ-SVR [2] is elected as the specific SVR method in this work. We select the four main SVR parameters, the kernel type K, penalty factor C, gamma γ in kernel function and ϵ in loss function. The individual $X_i = \{K, C, \gamma, \epsilon\}$.

2) The selection of evaluation method for PAs: The mean squared error *MSE* is the major index in regression or prediction problems. We can easily achieve *MSE* since ϵ-SVR of LIBSVM itself has the *MSE* output [2]. In this paper, *MSE* is elected as the fitness function (optimization objective). In a broad sense, ϵ-SVR and *MSE* are jointly responsible for the fitness function of HGSO.

The whole process of the SVR-HGSO model is depicted in Fig. 1. The specific steps are described as follows.

Step1 Confirm the lower and upper limits of K, C, γ and ϵ for forming the lower limit sets *LB* and the upper limit sets *UB*. $LB = \{lb_K, lb_C, lb_\gamma, lb_\epsilon\}$. $UB = \{ub_K, ub_C, ub\gamma, ub_\epsilon\}$. So that, we initialize randomly in the limits of *LB* and *UB*. The maximum of the iteration is marked as *MaxIter*. The loop count variable m is set to 1.

Step2 Divide the input samples into training data and testing data.

Step3 Based on X and training data, the supervised learning is done using ϵ-SVR for obtaining training models.

Step4 Obtain *MSE* sets based on testing data using training models.

Step5 Select the optimal *MSE* and corresponding current optimum SVR parameter set X_{opt}.

Step6 If $m > MaxIter$, go to Step9; otherwise, go to Step7.

Step7 Update X based on X_{opt} using HGSO's core operations, such as Henry's coefficient updating, solubility updating, position updating, local optimum escaping and the position updating of the worst individuals.

Step8 $m = m + 1$. Go to Step3.

Step9 Output X_{opt}.

A. HENRY GAS SOLUBILITY OPTIMIZATION

The HGSO algorithm was proposed in 2019 by Hashim *et al.* [17]. Based on Henry's law, it imitates the huddling behavior of gas to balance exploitation and exploration in the search space and avoid local optima. It should be
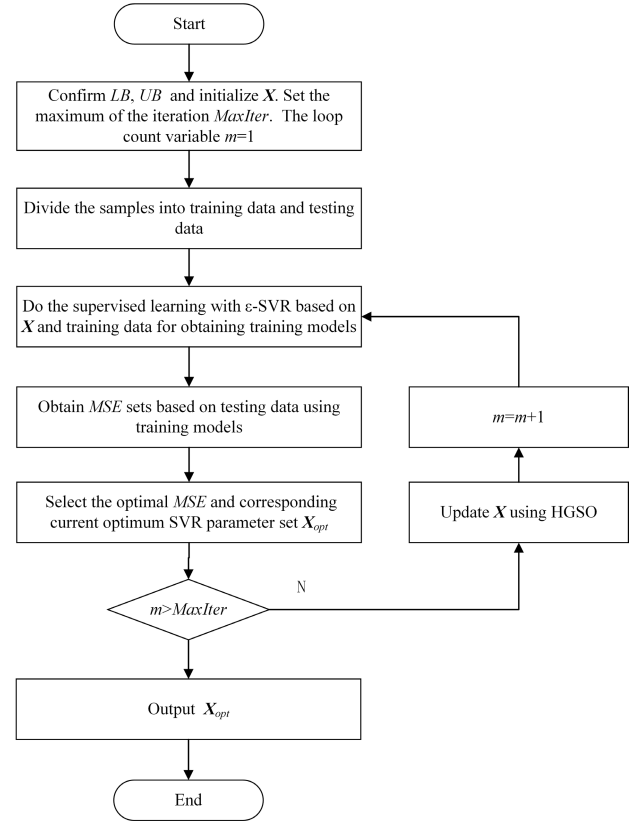

FIGURE 1. Schematic diagram of the SVR-HGSO approach.

noted that the gas population here corresponds to the SVR parameter population. The core operations required for this work are listed as follows.

1) UPDATE HENRY'S COEFFICIENT
Henry's coefficient is calculated via Eq. (1).

$$H_j(m+1) = H_j(m) \cdot \exp\left(-C_j \cdot \left(1/T(m) - 1/T^\theta\right)\right). \quad (1)$$

where $T(m)$ denotes the temperature of mth generation. $T(m) = \exp(-m/MaxIter)$. $T^\theta = 298.15$. It is worth noting that there is another level between population and individual: cluster in HGSO. The population is divided into equal clusters equivalent to the number of gas types. Each cluster shares the same Henry's constant value (H_j). The first generation coefficient $H_j(1) = 0.05 \cdot rand()$. $rand()$ is a function that generates a random number between 0 and 1. Another coefficient $C_j = 0.01 \cdot rand()$.

2) UPDATE SOLUBILITY
The solubility is attained using Eq. (2).

$$S_{i,j}(m) = Ks \cdot H_j(m+1) \cdot P_{i,j}(m). \quad (2)$$

where $S_{i,j}(m)$ represents the solubility of gas i in cluster j of mth generation. Ks is a constant. $P_{i,j}(m)$ denotes the partial pressure on gas i in cluster j of mth generation. $P_{i,j}(m) = 100 \cdot rand()$.

3) UPDATE POSITION

The position here corresponds to the SVR parameters in this work. This operation is very critical and updated using Eq. (3).

$$X_{i,j}(m+1) = X_{i,j}(m) + F \cdot r \cdot \eta \left(X_{j,opt}(m) - X_{i,j}(m) \right)$$
$$+ F \cdot r \cdot \alpha \left(S_{i,j}(m) \cdot X_{opt}(m) - X_{i,j}(m) \right)$$
$$\eta = \beta \cdot \exp(-\frac{F_{opt}(m) + 0.05}{F_{i,j}(m) + 0.05}). \qquad (3)$$

where $X_{i,j}(m)$ indicates the position of gas i in cluster j of mth generation. $X_{i,j}(m+1)$ is the next position of $X_{i,j}(m)$. r is a random value between 0 and 1. $\alpha = 1$. β is a constant. F is the flag that changes the direction of the search agent and provides diversity $= \pm$. $X_{j,opt}(m)$ is the best gas in cluster j of mth generation. $X_{opt}(m)$ represents the best gas of mth generation. $F_{opt}(m)$ denotes the fitness (*MSE*) of the best gas of mth generation. $F_{i,j}(m)$ is the fitness of gas i in cluster j of mth generation.

4) ESCAPE FROM LOCAL OPTIMUM

Rank and elect the number of worst individuals (N_w) using Eq. (4).

$$N_w = n \cdot (rand(0.1) + 0.1). \qquad (4)$$

5) UPDATE THE POSITION OF THE WORST INDIVIDUALS

Let the worst individual regenerate within the numerical range using Eq. (5).

$$G_k = LB + ra \cdot (\text{UB-LB}). \qquad (5)$$

where G_k denotes the position of the worst individuals. $1 \leqslant k \leqslant N_w$. ra indicates a random value between 0 and 1.

The pseudo-code of core operations of HGSO is displayed in Fig. 2.

Input: X, *MSE* sets, X_{opt}, F_{opt} and gt_num (number of gas type)

$i=1, j=1$
for $j \leqslant gt\_num$ **do**
   Update Henry's coefficient using Eq. (1).
End for
for $i \leqslant n$ **do**
   for $j \leqslant gt\_num$ **do**
      Update solubility $S_{i,j}$ using Eq. (2).
      Update position $X_{i,j}$ of next generation using Eq. (3) .
   end for
end for
Rank and elect the number of worst individuals using Eq. (4).
Update the position of the worst individuals using Eq. (5).
Output updated $X_{i,j}$

FIGURE 2. Pseudo-code of HGSO's core operations in this work.

B. ϵ-SUPPORT VECTOR REGRESSION

The ϵ-SVR prediction approach is a type of SVR that was developed by Vapnik [1]. We consider a set of training samples, $\{(xt_1, z_1), \cdots, (xt_l, z_l)\}$, where $xt_i \in R^n$ is a feature vector and $z_i \in R^1$ is the target output. The penalty factor $C > 0$ and $\epsilon > 0$. The standard form of ϵ-SVR is

$$\begin{cases} \min_{\mathbf{w},b,\xi,\xi^*} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i + C\sum_{i=1}^{l}\xi_i^* \\ s.t. \ \mathbf{w}^T\phi(xt_i) + b - z_i \leq \epsilon + \xi_i \\ \quad z_i - \mathbf{w}^T\phi(xt_i) - b \leq \epsilon + \xi_i^* \\ \quad \xi_i, \quad \xi_i^* \geq 0, \ i = 1, \cdots, l. \end{cases} \qquad (6)$$

The dual problem is

$$\begin{cases} \min_{\alpha,\alpha^*} \frac{1}{2}\sum_{i,j=1}^{l}(\alpha - \alpha^*)^T Q(\alpha - \alpha^*) + \epsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) \\ \quad + \sum_{i=1}^{l} z_i(\alpha_i - \alpha_i^*) \\ s.t. \ e^T(\alpha - \alpha^*) = 0 \\ \quad 0 \leq \alpha_i, \quad \alpha_i^* \leq C \\ \quad i = 1, \cdots, l. \end{cases} \qquad (7)$$

where $Q_{ij} = K(xt_i, xt_j) = \phi(xt_i)^T \phi(xt_j)$ and means the kernel function. $e = [1, 1, \cdots, 1]^T$.

We apply LIBSVM to resolve the ϵ-SVR problem and output $\alpha^* - \alpha$. It's remarkable that the selection of $K(\cdot)$ has a significant impact on the output and operation efficiency. $K(\cdot)$ must be the symmetric function satisfying Mercer condition. Four classic forms are listed as follows.

1) Linear kernels:

$$K(\cdot) = \mathbf{u}^T \cdot \mathbf{v}. \qquad (8)$$

2) Polynomial kernels:

$$K(\cdot) = \left(\gamma \left(\mathbf{u}^T \cdot \mathbf{v} \right) + coef \right)^d. \qquad (9)$$

3) RBF kernels:

$$K(\cdot) = \exp\left(-\gamma|\mathbf{u} - \mathbf{v}|^2 \right). \qquad (10)$$

4) Sigmoid kernels:

$$K(\cdot) = \tanh\left(\gamma \left(\mathbf{u}^T \cdot \mathbf{v} \right) + coef \right). \qquad (11)$$

RBF and sigmoid kernel functions have high efficiency, especially in the case of predicting high dimensional samples. Therefore, these two functions are elected as our alternatives. The kernel function is elected via the kernel type K in LIBSVM. The coefficient *coef* is set to 0. In general, the remaining parameters we can adjust are K, C, γ and ϵ. In fact, the standard ϵ-SVR just solve the problem which the dimension of target output is 1. In case of multi-dimensional target output problem, it's necessary to split the output into single dimension and forecast separately, and finally average the prediction results.

TABLE 1. Essential information of ten training data sets.

Dataset Id	Dataset	Number of instances	Number of attributes	Attribute characteristics
a	Airfoil self-noise	1503	6	Real
b	Air quality	9358	15	Real
c	Automobile	205	26	Categorical, Integer, Real
d	Communities and crime	1994	128	Real
e	Concrete slump test	103	10	Real
f	Container crane controller	15	3	Real
g	Energy efficiency	768	8	Integer, Real
h	Solar flare	1389	10	Categorical
i	Appliance energy	19735	29	Real
j	Beijing PM2.5	43824	13	Integer, Real

IV. CASE STUDY

The experimental conditions involve (i) a PC with an Intel Xeon W-2123, 3.60GHz, 16GB RAM, and Windows 10, (ii) Matlab R2018a, (iii) LIBSVM software package for Matlab, and (iv) the data sets, which were obtained from the University of California at Irvine (UCI) machine learning repository [31]. The main features of 10 data sets are shown in Table 1. The data sets are varied in their number of instances and attributes for testing the performance to deal with the problems of different complexity. The data set i and j have too high instances. In order to speed up the operation, they are normalized between 0 and 1.

In this work, three experiments are performed. The first experiment is applied for testing the prediction accuracy and convergence stability. The second experiment is to verify the rate of convergence. In the last experiment, the computational complexity is calculated and compared for each approach.

A. EXPERIMENT I

For testing the prediction accuracy and convergence stability, the proposed SVR-HGSO approach is compared with SVR-HHO, SVR-SSA, SVR-DA, SVR-ALO, SVR-PSO, SVR-FA and single ϵ-SVR based on the ten data sets. The initial parameter setting of SVR-HGSO and other algorithms are represented in Table 2. The 10-folds cross-validation training/testing technology is applied. The ratio of training data to testing data was 9 to 1. The optimal prediction accuracy (*MSE*) is obtained via the steps of Section 3 based on the training and testing data. This process is repeated ten times for each algorithm. The average (Avg) and standard deviation (Std) of ten optimal prediction accuracies are revealed in Table 3. The box-plot charts forms are represented in Fig. 3. In this figure, the box denotes the interquartile range, the whiskers indicate the maximum and minimum *MSE*, the cross sign in the box is the mean *MSE*, and the circles express the outliers of the accuracy values. These results reveal that SVR-HGSO achieves the best performance on data set h and j. In addition, SVR-HGSO has a badly stable output on data set a, c, d, e, g, h, i and j via the Std data. Fig. 4 demonstrates these data of all ϵ-SVR based algorithms. According to this figure, SVR-HGSO's convergence stability is optimum compared with the other hybrid algorithms. This result is very outstanding.

Furthermore, the Wilcoxon's test is utilized to test the statistical significance of the difference of the given prediction

outcomes and obtain *P* values with a 5% significance level. The *P* values of SVR-HGSO vs. other algorithms are shown in Table 4. The *P* values more than 0.05 reveal that the SVR-HGSO results have no statistically significant differences compared with other approaches. These data are marked with underline in Table 4. According to the table, there is no significant difference between our approach and SVR-DA on 7 data sets. Compared with SVR-SSA, the number of data sets with no significant difference is 6. Compared with SVR-HHO with the best accuracy, there is no significant difference on the data set a, b, e, i and j. In particular, compared with ϵ-SVR, this value is 0, which shows that there is a significant difference between the proposed approach and ϵ-SVR. The above results confirm the capacity of the HGSO algorithm to optimize the parameters of ϵ-SVR in terms of prediction accuracy and convergence stability.

B. EXPERIMENT II

In order to verify the convergence rate, SVR-HGSO is compared with SVR-HHO, SVR-SSA, SVR-DA, SVR-ALO, SVR-PSO and SVR-FA. The difference from Experiment I is the cut-off condition. The other experimental conditions are the same as Experiment I. This experiment's deadline condition is changed to make the prediction accuracy reach the maximum value of *MSE* of each algorithm in each data set of Table 3. The running iteration number is recorded. The process is repeated 10 times and the average number of iteration is calculated. The smaller the number, the faster the convergence rate. The results are revealed in Table 5.

SVR-HGSO obtains the minimum number of iterations on data set b, d, f and j. SVR-HHO achieves the best results on data set c, e and h. SVR-SSA gains the optimal convergence rate on data set a, f and i. SVR-DA performs the best on data set d and e. SVR-PSO has the optimal result on data set d and g. SVR-FA obtains the minimum number on data set b and d. SVR-ALO do not achieve optimal performance. For offering the comprehensive and intuitive evaluation (*RCRank*) on the convergence rate of these approaches, Eq. (12) is applied.

$$RCRank = \frac{1}{nd} \sum_{i=1}^{nd} RCrate_i$$
$$RCrate = \frac{RCold - RCMin}{RCMax - RCMin}. \qquad (12)$$

where *nd* is the total number of data sets. *RCrate* represents the normalized value of iteration number. *RCold* indicates the

TABLE 2. Initial parameter setting of SVR-HGSO and other algorithms.

Algorithm	Parameter Item	Value
SVR-HGSO	Number of search agents	30
	Number of maximum iterations	100
	Number of gas types	5
	Ks	1
	Minimum values of SVR parameters	(2,0.0001,0,0)
	Maximum values of SVR parameters	(3,100,100,100)
SVR-HHO	Number of search agents	30
	Number of maximum iterations	100
	The constant in Levy flights	1.5
	Minimum values of SVR parameters	(2,0.0001,0,0)
	Maximum values of SVR parameters	(3,100,100,100)
SVR-SSA	Number of search agents	30
	Number of maximum iterations	100
	Minimum values of SVR parameters	(2,0.0001,0,0)
	Maximum values of SVR parameters	(3,100,100,100)
SVR-DA	Number of search agents	30
	Number of maximum iterations	100
	Separation weight	Randomness between 0 and 0.192
	Alignment weight	Randomness between 0 and 0.192
	Cohesion weight	Randomness between 0 and 0.192
	Food attraction weight	Randomness between 0 and 2
	Enemy distraction weight	0.096
	Minimum values of SVR parameters	(2,0.0001,0,0)
	Maximum values of SVR parameters	(3,100,100,100)
SVR-ALO	Selection mechanism	Roulette wheel
	Number of search agents	30
	Number of maximum iterations	100
	Minimum values of SVR parameters	(2,0.0001,0,0)
	Maximum values of SVR parameters	(3,100,100,100)
SVR-PSO	Number of search agents	30
	Number of maximum iterations	100
	Personal learning coefficient	1.5
	Global learning coefficient	2.0
	Inertia weight	1
	Inertia weight damping ratio	0.99
	Minimum values of SVR parameters	(2,0.0001,0,0)
	Maximum values of SVR parameters	(3,100,100,100)
SVR-FA	Number of search agents	30
	Number of maximum iterations	100
	Coefficient α	0.5
	Attractiveness coefficient	0.2
	Absorption coefficient	1
	Minimum values of SVR parameters	(2,0.0001,0,0)
	Maximum values of SVR parameters	(3,100,100,100)
ϵ-SVR	SVR parameters	$(2,1,1/num\_feature, 0.1)$

Tips: $num\_feature$ means the number of feature attributes.

TABLE 3. Prediction accuracy results for all data of each algorithm.

Dataset Id	SVR-HGSO Avg±Std	SVR-HHO Avg±Std	SVR-SSA Avg±Std	SVR-DA Avg±Std	SVR-ALO Avg±Std	SVR-PSO Avg±Std	SVR-FA Avg±Std	ϵ-SVR Avg±Std
a	34.84±0.27	35.51±0.83	35.95±1.94	43.33±8.07	35.97±1.59	34.79±1.78	52.95±12.38	67.81±0
b	0.09±0.02	0.08±0	0.08±0	0.11±0.04	0.08±0	0.09±0.01	0.25±0.32	43.32±0
c	20.23±0	13.25±7.48	14.68±7.17	19.97±0.42	15.77±5.82	19.32±2.86	20.15±0.27	44.01±0
d	0.04±0	0.02±0	0.02±0	0.03±0.01	0.02±0.01	0.08±0.04	0.11±0.02	0.02±0
e	175.5±0	172.78±9.33	175.5±0	175.61±0.23	175.65±0.48	175.8±0.64	176.41±0.78	263.2±0
f	0.01±0.01	0±0	0±0	0±0	0±0	0.01±0.01	0.01±0.01	0.01±0
g	0.94±0	0.93±0	0.94±0.02	1±0.2	0.99±0.05	0.94±0	1±0.07	16.95±0
h	0.22±0	0.23±0	0.24±0	0.24±0	0.23±0	0.49±0.77	0.73±1.03	0.24±0
i	0.03±0.01	0.03±0.01	0.09±0.21	0.1±0.2	0.07±0	0.05±0.02	0.13±0.19	0.12±0
j	0.04±0.01	0.05±0.02	0.04±0.02	0.06±0.06	0.06±0.01	0.06±0.01	0.06±0.06	0.07±0

original value of iteration number. *RCMin* (*RCMax*) is the minimum (maximum) of iteration number in light of data set.

The result is displayed in Fig. 5. According to the figure, SVR-HGSO obtains the first place with a score of 0.22 (lower is better). SVR-HHO and SVR-DA also have a good performance. Compared with our approach, SVR-FA shows a relatively wide gap. The above results confirm the proposed approach has the capacity of rapid convergence.
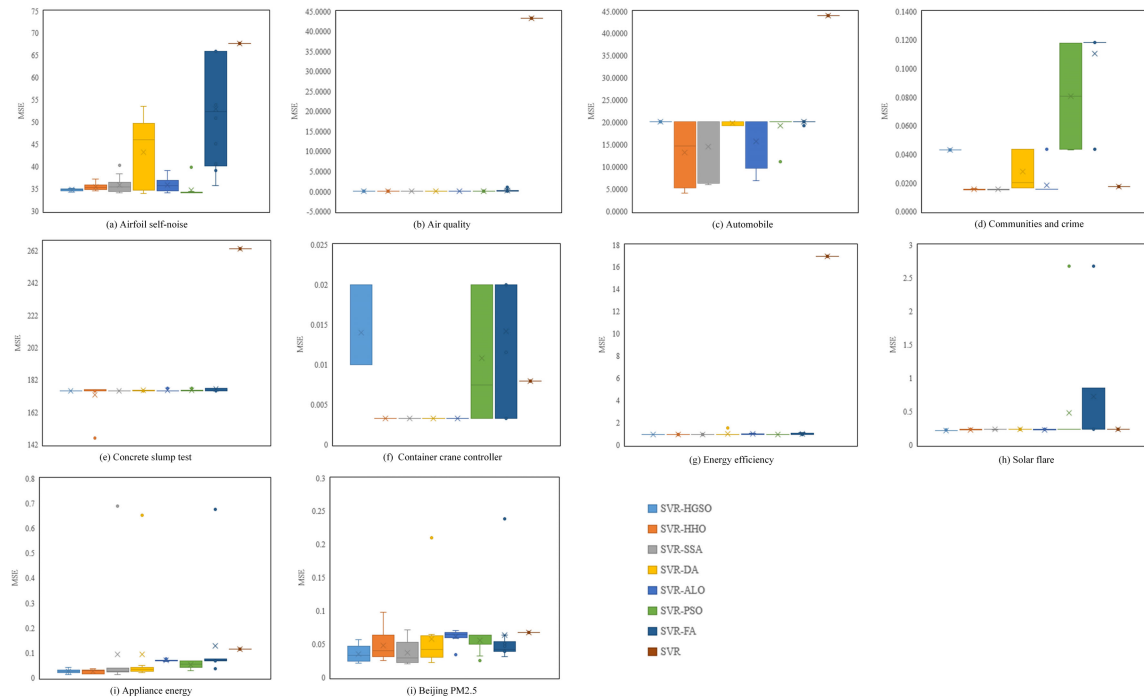
FIGURE 3. Box-plot charts of prediction accuracies for SVR-HGSO and other algorithms based on 10 data sets.

TABLE 4. P values of SVR-HGSO vs. other algorithms using Wilcoxon's statistical test (P > 0.05 are underlined).

Dataset Id	SVR-HHO	SVR-SSA	SVR-DA	SVR-ALO	SVR-PSO	SVR-FA	ϵ-SVR
a	5.39E-02	1.40E-01	1.04E-01	7.57E-02	3.61E-03	1.83E-04	6.39E-05
b	9.38E-01	4.97E-01	3.85E-02	1.05E-01	4.26E-01	1.47E-02	6.16E-05
c	1.19E-02	2.54E-02	5.09E-02	2.54E-02	1.92E-01	1.92E-01	2.43E-05
d	5.98E-05	5.98E-05	2.27E-01	1.27E-03	5.11E-05	2.43E-05	1.59E-05
e	3.73E-01	3.68E-01	1.67E-01	3.68E-01	1.67E-01	5.02E-03	1.59E-05
f	5.76E-05	5.76E-05	5.76E-05	5.76E-05	3.64E-01	7.77E-01	5.76E-05
g	1.00E-03	8.50E-01	9.70E-01	2.46E-04	1.85E-01	3.30E-04	6.39E-05
h	1.81E-04	1.82E-04	1.73E-04	1.83E-04	1.71E-04	1.10E-04	6.39E-05
i	8.50E-01	5.21E-01	6.40E-02	1.83E-04	7.00E-04	2.46E-04	6.39E-05
j	1.62E-01	9.10E-01	2.12E-01	7.69E-04	4.82E-03	8.90E-02	6.39E-05

TABLE 5. The average iteration numbers of SVR-HGSO vs. other algorithms based on 10 data sets.

Dataset Id	SVR-HGSO	SVR-HHO	SVR-SSA	SVR-DA	SVR-ALO	SVR-PSO	SVR-FA
a	43	103	30	95	33	112	839
b	25	60	36	26	34	27	25
c	34	20	28	29	29	25	36
d	8	10	49	8	30	8	8
e	28	15	26	15	31	29	30
f	101	131	101	202	130	165	187
g	519	667	752	495	520	393	793
h	38	16	46	41	41	72	160
i	247	254	238	249	386	295	421
j	559	757	988	761	648	1197	948

C. EXPERIMENT III

The computational complexities of these hybrid algorithms change with the number of sample cases and features, hence it is difficult to calculate them accurately. In this paper, we set the same criteria for these algorithms and obtain the corresponding running time of different samples for testing the computational complexity. The experimental conditions are similar to Experiment II. The deadline condition is the prediction accuracy reaches the average value of *MSE* for all algorithms in each data set of Table 3. The running time is recorded. The process is repeated 10 times and the average time is calculated. The shorter the time, the lower the computational complexity. The results are revealed in Table 6. The unit is seconds.
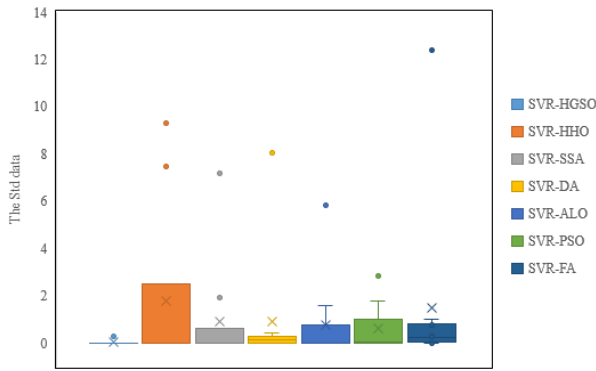
FIGURE 4. The Std data of SVR-HGSO vs. other ϵ-SVR based algorithms.
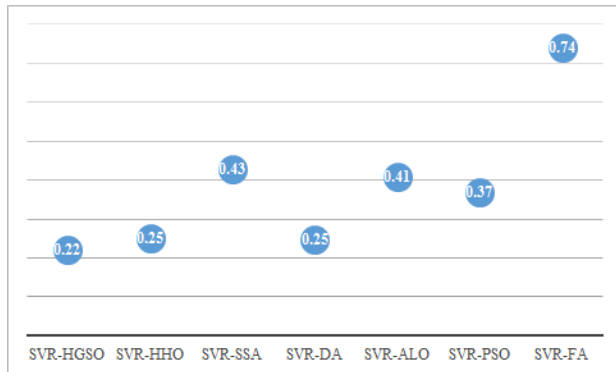

FIGURE 5. The *RCRank* of SVR-HGSO vs. other algorithms.

SVR-HGSO obtains the lowest computational complexity on data set f and j. It has competitive results on the other data sets. SVR-HHO achieves the minimum operation time on data set c and h. SVR-SSA gains the best results on data set a. SVR-DA performs the best on data set b, d, e and i. SVR-PSO has the optimal result on data set g. SVR-ALO and SVR-FA do not achieve optimal performance. For offering the comprehensive and intuitive evaluation (*CCRank*) on the computational complexity of these approaches, Eq. (13) is applied.

$$CCRank = \frac{1}{nd} \sum_{i=1}^{nd} CCrate_i$$
$$CCrate = \frac{CCold - CCMin}{CCMax - CCMin}. \quad (13)$$

where *CCtrate* represents the normalized value of operation time. *CCold* indicates the running time. *CCMin* (*CCMax*) is the minimum (maximum) of operation time for each data set.

The result is shown in Fig. 6. According to the figure, SVR-HGSO wins the first place with a score of 0.18 (lower is better). SVR-DA also have a good performance at the score of 0.22. The other approaches perform not so well. The above results confirm the proposed approach has the synthetically optimum computational complexity.
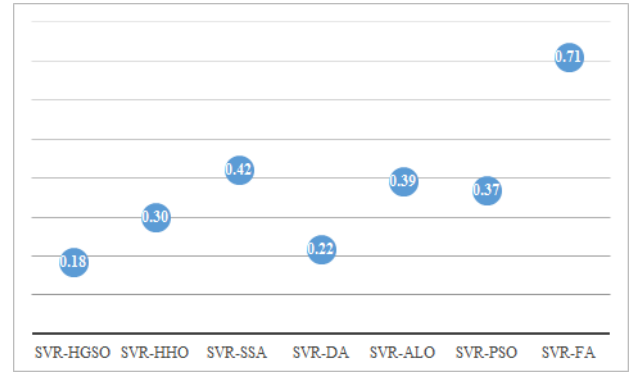

FIGURE 6. The *CCRank* of SVR-HGSO vs. other algorithms.

D. DISCUSSION

As a summary for all data sets in the three experiments, Table 7 lists the statistical values which indicate the number of data sets each algorithm won/tied/lost on prediction accuracy, associated *P* values, and chi-square χ^2. It reveals that SVR-HGSO outperforms other algorithms on 2 data sets. Regarding significance, the outcomes show SVR-HGSO is worse on 4, 3, 1, 2, 0, 0, and 2 data sets compared with SVR-HHO, SVR-SSA, SVR-DA, SVR-ALO, SVR-PSO, SVR-FA and single ϵ-SVR respectively. The χ^2 results indicate that the stability of SVR-HGSO is second only to ϵ-SVR (lower means more stable). That means that SVR-HGSO is better than other hybrid SVR methods in terms of convergence stability.

In order to fully evaluate the approaches, Eq. (14) is utilized to test comprehensive performance (*Rank*) regarding the prediction accuracy, convergence stability and rate, and computational complexity. *Wrate* indicates the normalized winning rate of *MSE*. *Srate* means the normalized stability ratio on χ^2. The bigger the winning rate, the better the prediction accuracy. However, the lower the stability ratio, the better the prediction stability performance. Therefore, the two are opposite and need to be unified. The different normalization formulas shown in Eq. (14). Finally, the lower the value of *Rank*, the better the overall performance. The results are listed in Table 7.

$$Rank = \frac{1}{4}Wrate + \frac{1}{4}Srate + \frac{1}{4}RCRank + \frac{1}{4}CCRank$$
$$Wrate = \frac{WMax - Wold}{WMax - WMin}$$
$$Srate = \frac{Sold - SMin}{SMax - SMin}. \quad (14)$$

where *Wold* denotes the winning rate regrading *MSE*. *Wold* = $\frac{NumW}{NumT}$. *NumW* means the number of data sets each approach won on *MSE*. *NumT* is the total number of data sets. *WMax* is the maximum of *Wold*. *WMin* is the minimum of *Wold*. *Sold* indicates the value of each approach on χ^2. *SMax* is the maximum of *Sold*. *SMin* is the minimum of *Sold*.

According to this table, SVR-HGSO obtains the best score of 0.28, 24% lower than SVR-HHO. The comparison

TABLE 6. The average running time of SVR-HGSO vs. other algorithms based on 10 data sets.

Dataset Id	SVR-HGSO	SVR-HHO	SVR-SSA	SVR-DA	SVR-ALO	SVR-PSO	SVR-FA
a	2.41	6.59	1.84	6.06	2.09	7.80	65.82
b	20.44	55.3	32.83	14.61	30.49	22.05	21.76
c	1.11	0.94	1.23	1.36	1.37	1.18	1.69
d	1.02	1.31	7.27	0.70	4.28	1.02	1.05
e	1.04	0.58	0.93	0.50	1.15	1.05	1.11
f	1.00	1.30	1.02	2.01	1.28	1.65	1.86
g	21.37	43.4	33.74	11.44	14.41	9.14	25.41
h	0.36	0.15	0.43	0.39	0.39	0.67	1.58
i	17.29	18.00	16.14	13.27	45.21	22.83	61.05
j	38.28	51.88	70.61	52.13	44.37	82.00	64.95

TABLE 7. Summary statistical results.

	SVR-HGSO			SVR-HHO			SVR-SSA			SVR-DA			SVR-ALO			SVR-PSO			SVR-FA			ϵ-SVR		
	W	T	L	W	T	L	W	T	L	W	T	L	W	T	L	W	T	L	W	T	L	W	T	L
MSE	2	0	8	7	0	3	2	0	8	1	0	9	2	0	8	1	0	9	0	0	10	0	0	10
P value	N/A	N/A	N/A	4	5	1	3	6	1	1	7	2	2	3	5	0	5	5	0	3	7	2	0	8
χ^2	2.37			8.67			7.53			8.05			7.26			7.71			9.21			0.00		
Rank	0.28			0.37			0.58			0.54			0.56			0.59			0.86			N/A		

proportion with other algorithms is even lower. These mean the proposed algorithm has the optimal comprehensive performance. SVR-HGSO has the optimum prediction accuracy on data set h and j. It holds the best stability of convergence on data set a, c, d, e, g, h and j, except for ϵ-SVR. Regarding the convergence rate, it obtains the optimum on data set b, d, f, i and j. It has the lowest computational complexity on data set f and j. It has competitive performances in many other data sets. These prove that SVR-HGSO has the high effectiveness and efficiency, is better at dealing with high-dimensional data and performs not bad on the low-dimensional data.

In terms of prediction accuracy, convergence stability and rate, and computational complexity, SVR-HGSO demonstrates a very competitive power. In fact, the integrated approaches sacrifice a small part of ϵ-SVR's stability to pursue prediction accuracy. In this respect, SVR-HGSO is the best balanced compared with SVR-HHO, SVR-SSA, SVR-DA, SVR-ALO, SVR-PSO and SVR-FA and obtains the optimum convergence performance and computational complexity. Hence, we state that the HGSO algorithm is worthy of application in the field of SVR parameter optimization. It has the potential to leverage the whole prediction process.

V. CONCLUSIONS
This work presents a novel integrated approach for optimizing SVR parameters using the HGSO algorithm. The proposed SVR-HGSO approach realizes the continuous optimization of SVR parameters via the iterative process. During this period, the prediction accuracy is generated continually until the cut-off conditions are met. We finally obtain the optimal prediction accuracy and the corresponding SVR parameters. The three experiments of prediction accuracy, convergence and computational complexity were implemented. The varied experimental outcomes on the benchmark data sets revealed SVR-HGSO's best comprehensive performance compared with other well-known algorithms.

The study confirms an immense potential of the SVR-HGSO approach in forecasting field.

For future work, the new ideas on improving the SVR model and/or the HGSO algorithm can be proposed. Secondly, the application of real-world engineering optimization problem should be involved.

REFERENCES
[1] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 988–999, Sep. 1999.
[2] 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, Apr. 2011.
[3] I. Aljarah, A. M. Al-Zoubi, H. Faris, M. A. Hassonah, S. Mirjalili, and H. Saadeh, "Simultaneous feature selection and support vector machine optimization using the grasshopper optimization algorithm," *Cognit. Comput.*, vol. 10, no. 3, pp. 478–495, Jun. 2018.
[4] A. Kazem, E. Sharifi, F. K. Hussain, M. Saberi, and O. K. Hussain, "Support vector regression with chaos-based firefly algorithm for stock market price forecasting," *Appl. Soft Comput.*, vol. 13, no. 2, pp. 947–958, Feb. 2013.
[5] A. Soualhi, K. Medjaher, and N. Zerhouni, "Bearing health monitoring based on Hilbert–Huang transform, support vector machine, and regression," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 1, pp. 52–62, Jan. 2015.
[6] W.-C. Hong, "Electric load forecasting by seasonal recurrent SVR (support vector regression) with chaotic artificial bee colony algorithm," *Energy*, vol. 36, no. 9, pp. 5568–5578, Sep. 2011.
[7] C. L. Wu, K. W. Chau, and Y. S. Li, "River stage prediction based on a distributed support vector regression," *J. Hydrol.*, vol. 358, nos. 1–2, pp. 96–111, Aug. 2008.
[8] K. Mohammadi, S. Shamshirband, M. H. Anisi, K. A. Alam, and D. Petković, "Support vector regression based prediction of global solar radiation on a horizontal surface," *Energy Convers. Manage.*, vol. 91, pp. 433–441, Feb. 2015.
[9] X. Wang, M. Kang, X. Fu, and C. Li, "Predictive modeling of surface roughness in lenses precision turning using regression and support vector machines," *Int. J. Adv. Manuf. Technol.*, vol. 87, nos. 5–8, pp. 1273–1281, Nov. 2016.
[10] J. Rui, H. Zhang, D. Zhang, F. Han, and Q. Guo, "Total organic carbon content prediction based on support-vector-regression machine with particle swarm optimization," *J. Petroleum Sci. Eng.*, vol. 180, pp. 699–706, Sep. 2019.
[11] M. Amroune, I. Musirin, T. Bouktir, and M. Othman, "The amalgamation of SVR and ANFIS models with synchronized phasor measurements for on-line voltage stability assessment," *Energies*, vol. 10, no. 11, p. 1693, 2017.

[12] Z. Li, Y. Xie, X. Li, and W. Zhao, "Prediction and application of porosity based on support vector regression model optimized by adaptive dragonfly algorithm," in *Proc. Energy Sour., A, Recovery, Utilization, Environ. Effects*, Jun. 2019, pp. 1–14.

[13] F. Kang, J. Li, and J. Dai, "Prediction of long-term temperature effect in structural health monitoring of concrete dams using support vector machines with Jaya optimizer and Salp swarm algorithms," *Adv. Eng. Softw.*, vol. 131, pp. 60–76, May 2019.

[14] J. Fan, Q. Hu, and Z. Tang, "Predicting vacant parking space availability: An SVR method with fruit fly optimisation," *IET Intell. Transp. Syst.*, vol. 12, no. 10, pp. 1414–1420, Dec. 2018.

[15] K. Roushangar and A. Koosheh, "Evaluation of GA-SVR method for modeling bed load transport in gravel-bed rivers," *J. Hydrol.*, vol. 527, pp. 1142–1152, Aug. 2015.

[16] E. H. Houssein, Y. Mina, and E. H. Aboul, "Nature-inspired algorithms: A comprehensive review," in *Hybrid Computational Intelligence: Research and Applications*. Boca Raton, FL, USA: CRC Press, 2019.

[17] F. A. Hashim, E. H. Houssein, M. S. Mabrouk, W. Al-Atabany, and S. Mirjalili, "Henry gas solubility optimization: A novel physics-based algorithm," *Future Gener. Comput. Syst.*, vol. 101, pp. 646–667, Dec. 2019.

[18] F. A. Hashim, E. H. Houssein, K. Hussain, M. S. Mabrouk, and W. Al-Atabany, "A modified Henry gas solubility optimization for solving motif discovery problem," *Neural Comput. Appl.*, to be published. [Online]. Available: https://link.springer.com/article/10.1007/s00521-019-04611-0#citeas, doi: 10.1007/s00521-019-04611-0.

[19] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.

[20] X. Zhang, X. Chen, and Z. He, "An ACO-based algorithm for parameter optimization of support vector machines," *Expert Syst. Appl.*, vol. 37, no. 9, pp. 6618–6628, 2010.

[21] A. A. K. Ismaeel, I. A. Elshaarawy, E. H. Houssein, F. H. Ismail, and A. E. Hassanien, "Enhanced elephant herding optimization for global optimization," *IEEE Access*, vol. 7, pp. 34738–34752, 2019.

[22] K.-Y. Chen and C.-H. Wang, "Support vector regression with genetic algorithms in forecasting tourism demand," *Tourism Manage.*, vol. 28, no. 1, pp. 215–226, Feb. 2007.

[23] C. Sunli, S. Jun, M. Hanping, W. Xiaohong, W. Pei, and Z. Xiaodong, "Non-destructive detection for mold colonies in rice based on hyperspectra and GWO-SVR: Non-destructive detection for mold colonies in rice," *J. Sci. Food Agricult.*, vol. 98, no. 4, pp. 1453–1459, Mar. 2018.

[24] S. Osama, H. H. Essam, D. Ashraf, E. H. Aboul, and A. A. Fahmy, "An optimized support vector regression using whale optimization for long term wind speed forecasting," in *Series in Machine Perception and Artificial Intelligence, Hybrid Metaheuristics*. Singapore: World Scientific, 2018, pp. 171–196.

[25] S. Osama, A. Darwish, E. H. Houssein, A. E. Hassanien, A. A. Fahmy, and A. Mahrous, "Long-term wind speed prediction based on optimized support vector regression," in *Proc. 8th Int. Conf. Intell. Comput. Inf. Syst. (ICICIS)*, Cairo, Egypt, Dec. 2017, pp. 191–196.

[26] A. E. Hassanien, M. Kilany, E. H. Houssein, and H. AlQaheri, "Intelligent human emotion recognition based on elephant herding optimization tuned support vector regression," *Biomed. Signal Process. Control*, vol. 45, pp. 182–191, Aug. 2018.

[27] W.-C. Hong, "Traffic flow forecasting by seasonal SVR with chaotic simulated annealing algorithm," *Neurocomputing*, vol. 74, nos. 12–13, pp. 2096–2107, Jun. 2011.

[28] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[29] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009.

[30] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: A Metaheuristic approach to solve structural optimization problems," *Eng. Comput.*, vol. 29, no. 1, pp. 17–35, Jan. 2013.

[31] M. Lichman. (2013). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml
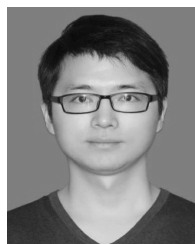
WEIDONG CAO received the Ph.D. degree in mechanical engineering from Chongqing University, Chongqing, China, in 2018.

He is currently working as a Lecturer with the College of IoT Engineering, Hohai University. His research interests include swarm intelligence optimization algorithm and data-driven modeling.
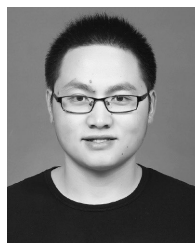
XIN LIU (Student Member, IEEE) received the Ph.D. degree in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2019.

He is currently with the College of Internet of Things Engineering, Hohai University, Changzhou, China. His research interests include nonlinear systems identification, data-driven process modeling, and soft sensor development.

JIANJUN NI (Senior Member, IEEE) received the B.E., M.E., and Ph.D. degrees from the China University of Mining and Technology, Xuzhou, China, in 1999, 2002, and 2005, respectively.

He is currently working as a Professor with the College of IoT Engineering, Hohai University. His research interests include multi-robot systems, information processing, complex systems modeling and control, and water conservancy information.

• • •