

An Efficient Parameter Adaptive Support Vector Regression Using K-Means Clustering and Chaotic Slime Mould Algorithm

ZIYI CHEN^{ID}, (Graduate Student Member, IEEE), AND WENBAI LIU

College of Economics and Management, Shanghai Maritime University, Shanghai 201306, China

Corresponding author: Ziyi Chen (ziyichen_smu@yeah.net)

This work was supported in part by the National Natural Science Foundation of China under Grant 51078228, and in part by the Shanghai Graduate Education Innovation Plan Implementation Project under Grant 20131129.

ABSTRACT Support vector regression (SVR) performs satisfactorily in prediction problems, especially for small sample prediction. The setting parameters (e.g., kernel type and penalty factor) profoundly impact the performance and efficiency of SVR. The adaptive adjustment of the parameters has always been a research hotspot. However, the substantial time cost and forecast accuracy of parameter adjustment are challenging to many scholars. The contradiction of big data prediction is especially prominent. In the paper, an SVR-based prediction approach is presented using the K-means clustering method (KMCM) and chaotic slime mould algorithm (CSMA). Eight high- and low-dimensional benchmark datasets are applied to obtain appropriate key parameters of KMCM and CSMA, and the forecast accuracy, stability performance and computation complexity are evaluated. The proposed approach obtains the optimal (joint best) forecast accuracy on 6 datasets and produces the most stable output on 3 datasets. It ranks first with a score of 0.024 in the overall evaluation. The outcomes reveal that the proposed approach is capable of tuning the parameters of SVR. KMCM, CSMA and SVR are skillfully integrated in this work and perform well. Although the performance is not outstanding in terms of stability, the proposed approach exhibits very strong performance with respect to prediction accuracy and computation complexity. This work validated the tremendous potential of the proposed approach in the prediction field.

INDEX TERMS Machine learning, forecasting problem, K-means clustering method, chaotic slime mould algorithm, support vector regression.

I. INTRODUCTION

As a part of support vector machine (SVM) [1]–[3], support vector regression (SVR) has produced gratifying achievements in the field of prediction. However, poor choices of penalty factor, kernel type and others can markedly reduce the SVR's capability. Hence, many heuristic algorithms are combined with SVR to realize SVR parameter tuning, including the Henry gas solubility optimization (HGSO) [4], salp swarm algorithm (SSA) [5], dragonfly algorithm (DA) [6], ant lion optimizer (ALO) [7], particle swarm optimization (PSO) [8], firefly algorithm (FA) [9] and others. With the addition of these heuristic algorithms, the prediction accuracy and generalization ability have been greatly improved.

The associate editor coordinating the review of this manuscript and approving it for publication was Inês Domingues^{ID}.

However, the parameter tuning requires constant iterations, which involves running SVR many times. The enormous time cost is challenging for many scholars, especially in high-dimensional data prediction.

The integration of the K-means clustering method (KMCM) is an interesting and feasible solution [10]. KMCM is applied to SVR for sales forecasting [11]. For accurate export trade value prediction, a three-stage model is proposed, which consists of FA-based SVR, FA-based KMCM and wavelet transform [12]. The running times of these methods are excellent. However, the prediction accuracies of these methods are not ideal.

Two meaningful findings can be gained from the above-mentioned studies.

- 1) The hybrid approach of heuristic algorithms and SVR produces ideal prediction accuracy but requires

considerable time. KMCM makes the running time of hybrid SVR algorithms very short. However, it reduces prediction accuracy and stability.

- 2) Almost all research focuses on a particular area. The universality of the development algorithm requires further development.

Motivated by these findings, this paper aims to fill the gap and make contributions in the following two respects. First, based on the three-stage model of KMCM-heuristic algorithm-SVR, a new heuristic algorithm is sought to enable the prediction accuracy, stability and calculation complexity to reach the ideal level. Second, the datasets of different fields are applied to verify the proposed algorithm.

The slime mould algorithm (SMA) is a novel heuristic algorithm which was presented by Li *et al.* [13] in 2020. It is similar to the bacterial foraging optimization algorithm [14]–[16]. However, SMA involves a distinct mathematical model, using accommodative weights to imitate the generation process of positive and negative feedback of a slime mould propagation wave based on a bio-oscillator, forming the optimal path of food connection with good exploration capacity and development tendency. SMA has been evaluated on an extensive set of benchmarks and produced significant advantages against some classical algorithms (e.g., ant colony optimization (ACO) [17], [18], sine cosine algorithm (SCA) [19], SSA [20], grey wolf optimization (GWO) [21], [22], moth flame optimization (MFO) [23]–[25], and whale optimization algorithm (WOA) [26]).

In this paper, the chaotic slime mould algorithm (CSMA) is presented to improve SMA's search efficiency, and then an up-to-date KMCM-CSMA-SVR model is proposed. KMCM is in charge of cluster analysis of training data for reducing the time cost. CSMA is responsible for tuning the SVR parameters to acquire high forecast accuracy and stability. First, the cluster centers are obtained using KMCM. Second, SVR parameters are randomly generated to build the parameter population. Third, the prediction accuracy data are achieved via SVR and the population. Finally, the optimal parameters and population are renewed by CSMA. The last two steps are duplicated until the cut-off condition is met. Eight varied benchmark datasets are applied to calculate the prediction accuracy, stability and computation complexity. As the comparison algorithms, HGSO-SVR, SSA-SVR, DA-SVR, ALO-SVR, PSO-SVR, FA-SVR and KMCM-SVR also participated in the experiment.

There are two innovations. First, the ingenious combination of KMCM, CSMA and SVR is modeled. KMCM is used to improve the index of operation time. CSMA participates in SVR parameter optimization, which makes up for the impact of KMCM and improves the prediction accuracy and other performances. Second, the presented model has been verified on many occasions (e.g., for appliance energy, Beijing PM2.5, and wave energy converters).

The remaining sections are arranged below. The related work is listed in Section II. The KMCM-CSMA-SVR integrated model is shown in Section III. Section IV offers the

case study, including the three experiments and discussion. The last section presents the conclusion and the prospect for future research.

II. RELATED WORK

The grid search is a classical approach for adjusting SVR parameters. However, the experimental results showed that the heuristic algorithm is more efficient than grid search in this SVR tuning work [27]–[29]. The genetic algorithm (GA) was involved in the parameter tuning of SVR for modelling gravel bed Riverbed transport [30]. To predict the long-term temperature effect of concrete dam structure health monitoring, SSA was utilized to optimize SVR [5]. The chaos-based FA and SVR were combined to perform stock market price forecasting [9]. With the rapid development of these methods, SVR prediction accuracy and generalization ability have been greatly improved in practical application. However, the time required is approximately the number of iterations multiplied by the time that SVR runs alone. The time spent is greatly increased.

Hence, some scholars set out to improve operation efficiency. A rapid data preprocessing procedure was proposed for SVR [10]. The dataset was handled to obtain clusters using KMCM. SVR only trained the centroids of clusters. This method effectively reduced the time cost. The applications included sales forecasting [11], export trade value prediction [12], and hour-ahead frequency security assessment [31]. KMCM processes the training data to reduce the number of samples, thus reducing the operation time. At the same time, it discards some data, which affects the prediction accuracy and stability.

In the no free lunch theorem of optimization [32], it is revealed that there is no way to solve all problems with one algorithm. This encourages us to find a proper way to overcome the above difficulties. SMA is a novice approach. Its magic lies in its distinct mathematical model, which utilizes accommodative weights to imitate the process of generating positive and negative feedback of the propagation wave of slime mould. Compared with other heuristic algorithms, it produced competitive outcomes [13]. The chaotic local search is added to SMA to improve search efficiency. It offers us the feasibility and rationality to build the KMCM-CSMA-SVR model. The experimental results prove that KMCM-CSMA-SVR offers outstanding comprehensive performance.

III. METHODOLOGY OF THE PROPOSED ALGORITHM

The framework of KMCM-CSMA-SVR is presented below: a) obtain cluster centers based on datasets and KMCM, b) perform the population presentation of SVR parameters, c) gain prediction results based on the population and cluster centers using SVR, and d) renew the population and optimum parameters by CSMA's operations. Steps c and d are reduplicated until the deadline condition is met. The deadline condition is usually set to be the maximum number of iterations. The SVR parameter population \vec{X} consists of individuals

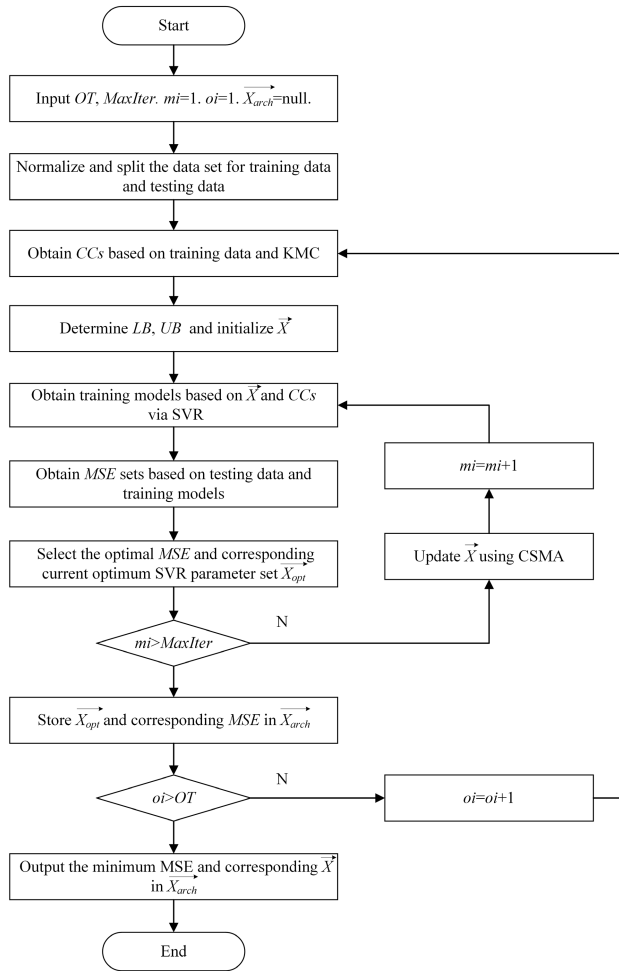


FIGURE 1. Frame diagram of the KMCM-CSMA-SVR model.

\vec{X}_i . $\vec{X} = \{\vec{X}_1, \vec{X}_2, \dots, \vec{X}_n\}$, where n is the number of individuals. The SVR parameters form the attributes of \vec{X}_i . The KMCM-CSMA-SVR algorithm is shown in Fig. 1. The detailed steps are listed below.

- Step 1 Input the operation times OT , KMC's number of clusters K , and the iteration maximum $MaxIter$. The CSMA's loop count variable mi equals 1. The operation count variable oi is equivalent to 1. The optimum SVR parameter archive is marked as \vec{X}_{arch} .
- Step 2 Normalize the data of the same property in the dataset to - 1 to 1 and distribute the dataset to training data and test data.
- Step 3 Obtain cluster centers CCs based on the training data, KMCM and parameter K .
- Step 4 Determine upper and lower limits of KT , PF , γ and ϵ to constitute the upper limit set UB and lower limit set LB . $UB = \{ub_{KT}, ub_{PF}, ub_{\gamma}, ub_{\epsilon}\}$. $LB = \{lb_{KT}, lb_{PF}, lb_{\gamma}, lb_{\epsilon}\}$. \vec{X} is initialized randomly under UB and LB .
- Step 5 Supported by \vec{X} and CCs , perform the supervised learning via ϵ -SVR to achieve training models.
- Step 6 Acquire MSE sets supported by test data and training models.

- Step 7 Elect the smallest MSE and corresponding optimal SVR parameter set \vec{X}_{opt} of the current generation.
- Step 8 If $mi > MaxIter$, store \vec{X}_{opt} and corresponding MSE in \vec{X}_{arch} . Go to Step 11; otherwise, go to Step 9.
- Step 9 Update \vec{X} based on \vec{X}_{opt} using CSMA's operations, i.e., food approaching, wrapping and grabbing.
- Step 10 $mi = mi + 1$. Go to Step 5.
- Step 11 If $oi > OT$, go to Step 12; otherwise, $oi = oi + 1$, go to Step 3.
- Step 12 Output the minimum MSE and corresponding SVR parameters in \vec{X}_{arch} .

A. CHAOTIC SLIME MOULD ALGORITHM

The SMA was proposed in 2020 by Li *et al.* [13]. It was vitalized by the oscillating patterns of slime mould. For forming the optimum path to connect food with outstanding exploratory capacity and exploitation propensity, a distinct mathematical model was presented using accommodative weights to imitate the process of generating positive and negative feedback of the propagation wave supported by the bio-oscillator. However, SMA suffers from some limitations, including relatively slow convergence rate and the dilemma of local optimum. In this paper, chaotic maps are added to SMA to improve the search efficiency and better exploit the locality of the solutions. The operations required for this work are listed below.

Approach Food: To model the approaching behaviour of slime mould, the following rule is presented to simulate the contraction pattern:

$$\vec{X}(t+1) = \begin{cases} \vec{X}_b(t) + \vec{vb}(\vec{W} \cdot \vec{X}_A(t) - \vec{X}_B(t)), & C < p \\ \vec{vc} \cdot \vec{X}(t), & C \geq p \end{cases} \quad (1)$$

where \vec{vb} is a parameter. Its range is $[-a, a]$. a is listed in Eq. (2). \vec{vc} reduces linearly from 1 to 0. t is the current generation. \vec{X}_b is the individual site where the highest concentration of odour has been found. \vec{X} indicates the slime mould's location (i.e., a set of SVR parameters). \vec{X}_A and \vec{X}_B are 2 individuals randomly elected from the swarm. \vec{W} indicates the slime mould's weight and is listed in Eq. (3). p is calculated via Eq. (5). C denotes the random value generated by chaotic maps [33]. In this paper, Logistic map is elected as the chaotic map method. The mathematical model can be seen in Eq. (6).

$$a = \arctan h(-t/max_t) + 1 \quad (2)$$

where max_t represents the maximum number of iterations.

$$\vec{W}(SmellIndex(i)) = \begin{cases} 1 + C \cdot \log\left(\frac{bF - S(i)}{bF - wF} + 1\right), \\ \text{condit} \\ 1 - C \cdot \log\left(\frac{bF - S(i)}{bF - wF} + 1\right), \\ \text{others} \end{cases} \quad (3)$$

where *condit* is $S(i)$ in the top half of the population. $bF(wF)$ represents the optimal (worst) fitness of the current iterative procedure. *SmellIndex* is the ranked sequence of fitness data.

$$SmellIndex = sort(S) \tag{4}$$

$$p = \tanh |S(i) - DF| \tag{5}$$

where $i \in 1, 2, \dots, n$. $S(i)$ indicates the fitness of \vec{X} . DF is the optimum acquired in all generations.

$$C_{t+1} = 4C_t(1 - C_t) \tag{6}$$

where the range of C is $[0,1]$

Wrap Food: Eq. (7) is used to renew the location of slime mould.

$$\vec{X}^* = \begin{cases} rand \cdot (U-L) + L, & rand < z \\ 1 + C \cdot \log\left(\frac{bF - S(i)}{bF - wF} + 1\right), & \textit{condit} \\ 1 - C \cdot \log\left(\frac{bF - S(i)}{bF - wF} + 1\right), & \textit{others} \end{cases} \tag{7}$$

where U and L are the upper and lower boundaries of the search scope. *rand* represents the randomness in $[0,1]$.

Grabble Food: \vec{vb} floats randomly in $[-a,a]$ and approaches 0 with the increase of iterations. The \vec{vc} floats from -1 to 1 and ultimately trends to 0.

The pseudo-code of CSMA is listed in Fig. 2. In this work, CSMA is utilized to update \vec{X} based on \vec{X}_{opt} according to the loop body of the following pseudo-code. The updated \vec{X} is used as the input parameters of SVR to make a new prediction and obtain the new \vec{X}_{opt} . The iteration does not stop until the cut-off condition is met. Hence, CSMA's operations are the key of KMCM-CSMA-SVR.

```

Initialize the parameters popsiz, max_t;
Initialize the positions of slime mould  $X_i(i=1,2,\dots,n)$ ;
While ( $t \leq max\_t$ )
    Calculate the fitness of all slime mould;
    Update bestFitness,  $X_b$ 
    Calculate the  $W$  by Eq. (3);
    For each search portion
        Update  $p$ ,  $vb$ ,  $vc$ ;
        Update positions by Eq. (7);
    End
     $t = t + 1$ ;
End While
Return bestFitness,  $X_b$ 
    
```

FIGURE 2. Pseudocode of CSMA's operations.

B. K-MEANS CLUSTERING METHOD

Although KMCM was proposed more than 60 years ago, and countless clustering algorithms have been published since then, KMCM is still diffusely applied [34]. Let $Y = \{y_1, y_2, \dots, y_m\}$ be the set of m d-dimensional points to be

polymerized into a set of K clusters, $Cl = \{c_1, c_2, \dots, c_K\}$. KMCM seeks out a partition to minimize the square error (SE) of the empirical mean and the points in the cluster. Let μ_i be the average of cluster $c_i, i = 1, 2, \dots, K$. The SE between μ_i and the points in cluster c_i is calculated below.

$$J(c_i) = \sum_{y_j \in c_i} \|y_j - \mu_i\| \tag{8}$$

The objective of KMCM is minimization of the sum of the SE of all K clusters,

$$J(Cl) = \sum_{j=1}^K \sum_{y_j \in c_i} \|y_j - \mu_i\| \tag{9}$$

KMCM starts from the original segmentation of K clusters and allocates patterns to clusters in order to decrease SE. Since SE always decreases with an increase in the number of clusters K (with $J(Cl) = 0$ when $K = m$), it can only be minimized in a fixed number of clusters. The key steps of KMCM are listed below [34].

- Step 1: Elect an original segmentation using K clusters; iterate steps 2 and 3 until cluster membership stabilizes.
- Step 2: An updated segmentation is generated by assigning each pattern to its nearest cluster centre.
- Step 3: Calculate new cluster centers.

KMCM demands 3 customized parameters: number of clusters K , cluster initialization, and distance metric. The cluster initialization can choose a random operation. The distance index can be a Euclidean metric, Cosine metric, etc. The hardest parameter to choose is K . In this paper, a cross-validation method is utilized to find the appropriate K for different datasets (i.e., K varies with the dataset).

C. ϵ -SUPPORT VECTOR REGRESSION

The ϵ -SVR prediction method was developed by Vapnik [1]. Consider training samples $\{(x_{t1}, z_1), \dots, (x_{tl}, z_l)\}$, where $x_{ti} \in R^n$ is a feature vector and $z_i \in R^1$ is the objective. $\epsilon > 0$ and the penalty factor $PF > 0$. ϵ -SVR is described as

$$\begin{cases} \min_{w, b, \xi, \xi^*} & \frac{1}{2} w^T w + PF \sum_{i=1}^l \xi_i + PF \sum_{i=1}^l \xi_i^* \\ \text{s.t.} & w^T \phi(x_{ti}) + b - z_i \leq \epsilon + \xi_i \\ & z_i - w^T \phi(x_{ti}) - b \leq \epsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geq 0, i = 1, \dots, l. \end{cases} \tag{10}$$

The dual form is exhibited below.

$$\begin{cases} \min_{\alpha, \alpha^*} & \frac{1}{2} \sum_{i,j=1}^l (\alpha - \alpha^*)^T Q (\alpha - \alpha^*) + \epsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) \\ & + \sum_{i=1}^l z_i (\alpha_i - \alpha_i^*) \\ \text{s.t.} & e^T (\alpha - \alpha^*) = 0 \\ & 0 \leq \alpha_i, \alpha_i^* \leq PF \\ & i = 1, \dots, l. \end{cases} \tag{11}$$

TABLE 1. Basic information of 8 datasets.

ID	Name	Instance number	Attribute number	Feature
D1	Airfoil self-noise	1503	6	Real
D2	Air quality	9358	15	Real
D3	Appliance energy	19735	29	Real
D4	Automobile price	205	26	Categorical, Integer, Real
D5	Beijing PM2.5	43824	13	Integer, Real
D6	Communities and crime	1994	128	Real
D7	Solar flare	1389	10	Categorical
D8	Wave energy converters (Sydney)	72000	49	Real

where $Q_{ij} = KT(xt_i, xt_j) = \phi(xt_i)^T \phi(xt_j)$. $e = [1, 1, \dots, 1]^T$.

LIBSVM is utilized to solve the problem and offer $\alpha^* - \alpha$. The choice of $KT(\cdot)$ has a remarkable influence on production and operational efficiency. $KT(\cdot)$ is the symmetric function satisfying the Mercer condition. Some conventional functions are displayed below.

1) Linear kernel:

$$KT(\cdot) = \mathbf{u}^T \cdot \mathbf{v}. \quad (12)$$

2) Polynomial kernel:

$$KT(\cdot) = \left(\gamma (\mathbf{u}^T \cdot \mathbf{v}) + coef \right)^d. \quad (13)$$

3) RBF kernel:

$$KT(\cdot) = \exp \left(-\gamma |\mathbf{u} - \mathbf{v}|^2 \right). \quad (14)$$

4) Sigmoid kernel:

$$KT(\cdot) = \tanh \left(\gamma (\mathbf{u}^T \cdot \mathbf{v}) + coef \right). \quad (15)$$

The third and fourth functions are very efficient, especially in the prediction of high-dimensional samples [4]. Hence, the functions are selected in this work. $coef$ is zero. The adjustable parameters are KT , PF , γ and ϵ , which are the optimization variables of this work.

The SVR model has two major limitations. First, it is inefficient in processing high-dimensional samples. Second, it is difficult to find the appropriate setting parameters. To overcome these limitations, KMCM is utilized to improve the ability to process high-dimensional data, and CSMA is applied to select reasonable parameter settings.

IV. CASE STUDY

The study conditions include (a) a computer with an Intel Core i5-6600, 3.60 GHz, 8 GB RAM, and Windows 7, (b) Matlab R2017a, (c) LIBSVM development kit, and (d) the datasets, which were acquired from the University of California at Irvine machine learning repository [35]. The information of eight datasets is displayed in Table 1. The number of instances and properties of datasets are different, which are used to test the performance with respect to complexity problems. The minimum number of properties is 6, and the maximum is 128. The minimum number of instances is 205, and the maximum is 72,000. The datasets D1, D4, D6 and D7 are selected to test the ability of each method to handle the

relatively small sample size. D2, D3, D5 and D8 are chosen because they include a large number of cases.

The key parameter setting experiments for KMCM and CSMA are first performed. The second test is utilized to verify the forecast accuracy and stability. The computation complexity of each algorithm is calculated and compared in the third experiment. The comparison with highly similar approaches is provided in the last experiment.

A. EXPERIMENT I: KEY PARAMETER SETTING EXPERIMENT FOR KMCM AND CSMA

The main parameters are KMCM's number of clusters K , CSMA's iteration maximum $MaxIter$, number of search agents nsa and parameter z . For setting suitable parameters for different datasets under KMCM-CSMA-SVR, diverse values of the parameters were analysed. In this experiment, other parameter settings are shown in Table 2.

TABLE 2. Some parameter settings of KMCM-CSMA-SVR and other models.

Algorithm	Item	Value
HGSO-SVR	Number of gas types	5
	Ks	1
	Parameter minimum of SVR	(2,0.0001,0,0)
	Parameter maximum of SVR	(3,50,50,50)
SSA-SVR	Parameter minimum of SVR	(2,0.0001,0,0)
	Parameter maximum of SVR	(3,50,50,50)
	Separation coefficient	Random number from 0 to 0.192
	Alignment coefficient	Random number from 0 to 0.192
DA-SVR	Cohesion coefficient	Random number from 0 to 0.192
	Food attraction coefficient	Random number from 0 to 2
	Enemy distraction coefficient	0.096
	Parameter minimum of SVR	(2,0.0001,0,0)
	Parameter maximum of SVR	(3,50,50,50)
	Selection mechanism	Roulette wheel
ALO-SVR	Parameter minimum of SVR	(2,0.0001,0,0)
	Parameter maximum of SVR	(3,50,50,50)
	Personal learning coefficient	1.5
	Global learning coefficient	2.0
PSO-SVR	Inertia weight	1
	Inertia weight damping ratio	0.99
	Parameter minimum of SVR	(2,0.0001,0,0)
	Parameter maximum of SVR	(3,50,50,50)
	Coefficient α	0.5
	Attractiveness coefficient	0.2
FA-SVR	Absorption coefficient	1
	Parameter minimum of SVR	(2,0.0001,0,0)
	Parameter maximum of SVR	(3,50,50,50)
	Cluster initialization	Randomly generated
KMCM-SVR	Distance metric	Euclidean metric
	SVR parameters	(2,1,1/Number of features, 0.1)
	Cluster initialization	Randomly generated
	Distance metric	Euclidean metric
	Chaotic map	Logistic map
	Parameter minimum of SVR	(2,0.0001,0,0)
KMCM-CSMA-SVR	Parameter maximum of SVR	(3,50,50,50)

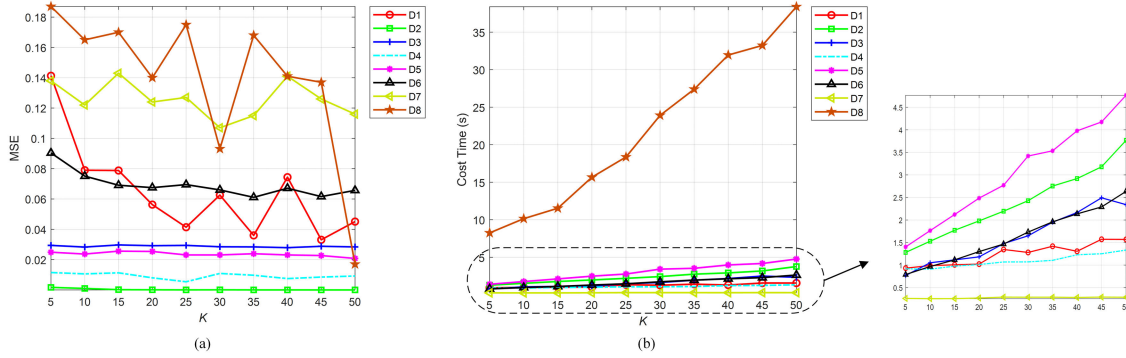


FIGURE 3. The MSE (a) and cost time (b) of KMCM-CSMA-SVR with the increase of K .

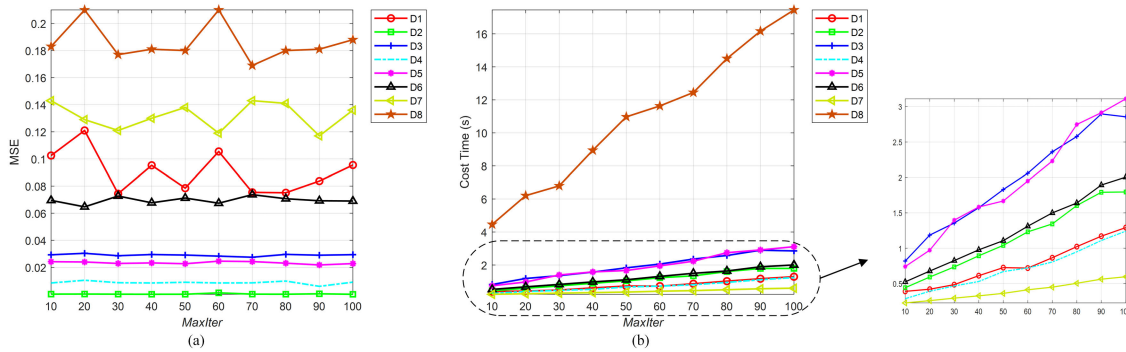


FIGURE 4. The MSE (a) and cost time (b) of KMCM-CSMA-SVR with the increase of $MaxIter$.

- 1) K : The parameter K controls the input of SVR and influences the prediction accuracy and other performance results. The effect of K was studied while the number of search agents ranged within [5,50]. The MSE and time cost are listed in Fig. 3. Considering the metric MSE , the values were very stable with the increase of K on datasets D2 and D3. The value oscillated and declined on D1, D6, D7 and D8. The MSE was stable and exhibited a slight concussion on D4 and D5. With respect to the time cost, the values presented an overall rising posture on datasets D1-D6 and D8. In particular, the time cost suddenly declined on D3 when $K = 50$. The values of time cost exhibited poor stability on D7.
- 2) $MaxIter$: The effect of $MaxIter$ was verified while the iteration number ranged within [10,100]. The outcomes are displayed in Fig. 4. From the figure, the MSE values were relatively stable for D2, D3, D4 and D5. The values fluctuated greatly on the other datasets. With the increase of $MaxIter$, the values of running time all exhibited an upward trend. D3 behaved differently, and the value declined under $MaxIter = 100$.
- 3) nsa : This parameter must be sufficient for detecting the search range. The influence was studied when nsa ranged from 10 to 100. The outcomes are presented in Fig. 5. According to the MSE values of this figure,

there were significant shocks on D1, D7 and D8. D4, D5 and D6 exhibited slight concussions. The data were very stable on D2 and D3. With respect to the data of running time, it presented an overall upward trend. Under the condition of $nsa = 100$, the time cost sharply declined on D3.

- 4) z : The parameter z directly affects the values of SVR parameters. The range is [0,0.09] [13]. Fig. 6 displays the data of MSE and time cost. From this figure, the MSE values oscillated violently on D1, D7 and D8. The small amplitude oscillations occurred on D4, D5 and D6. D2 and D3 performed in relatively stable fashion. According to the time cost data, all datasets exhibit a noisy trend as a whole, with no obvious upward or downward trend.

The normalization operation and Eq. (16) were carried out for selecting appropriate K , $MaxIter$, nsa and z and taking into account two major metrics: MSE and running time. The smaller $paSe$, the better. The prediction accuracy is examined more than running time in this paper. β is set to 0.8, while δ is 0.2. The $paSe$ outcomes are displayed in Table 3, where $ItemOptimum$ indicates the appropriate values of these parameters for datasets. From the table, D1-D8 have different $ItemOptimum$. This makes the prediction result superior.

$$paSe = \beta \cdot N_{MSE} + \delta \cdot N_{RT}. \tag{16}$$

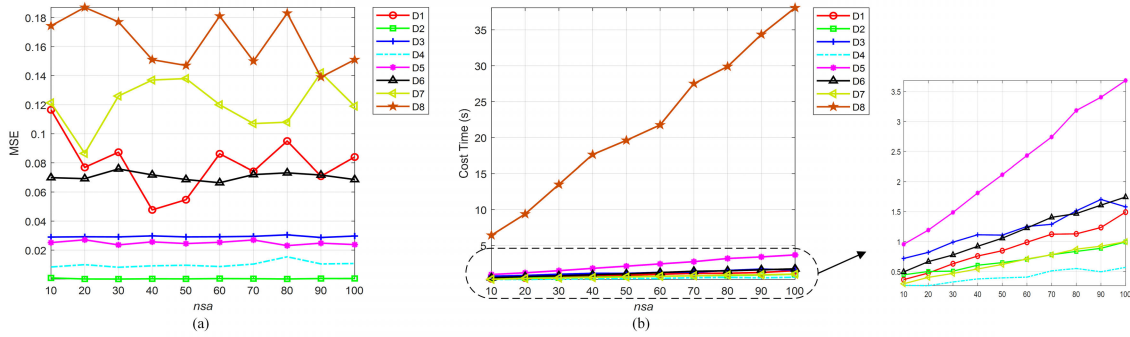


FIGURE 5. The MSE (a) and cost time (b) of KMCM-CSMA-SVR with the increase of nsa .

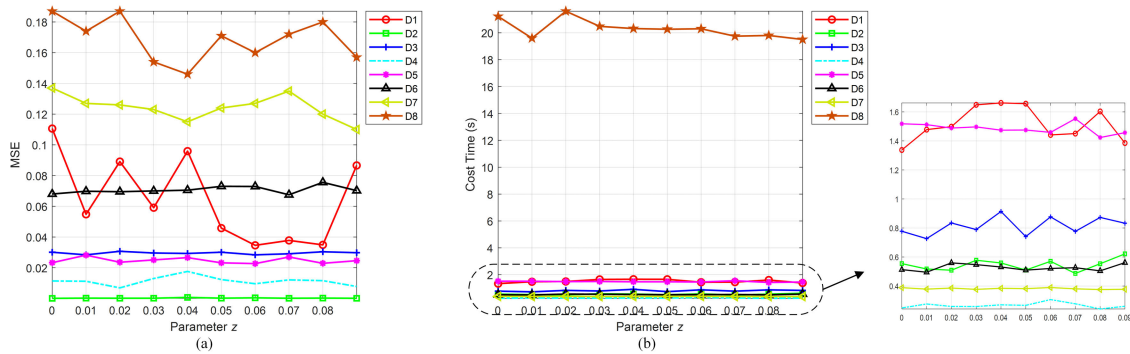


FIGURE 6. The MSE (a) and cost time (b) of KMCM-CSMA-SVR with the increase of z .

where β and δ are combining coefficients. $\beta + \delta = 1$. N_{MSE} and N_{RT} are normalized values of MSE and time cost on each dataset, respectively.

B. EXPERIMENT II: FORECAST ACCURACY AND STABILITY EXPERIMENT

For verifying the forecast accuracy and stability, KMCM-CSMA-SVR was compared with HGSO-SVR, SSA-SVR and other models on D1-D8. The original parameter settings of KMCM-CSMA-SVR and others are displayed in Tables 2 and 3 (*ItemOptimum*). The ten-fold cross-validation training/testing technology was adopted. The ratio of training data to test data was 9:1. The optimal MSE was achieved using the method of Section III supported by the datasets. Each algorithm repeated the process 10 times. The average (Avg) and standard deviation (Std) of forecast accuracies are listed in Table 4. The box-plot charts are shown in Fig. 7. The results revealed that KMCM-CSMA-SVR produced the optimal (joint best) performance on datasets D2, D3, D4, D5, D7 and D8. KMCM-CSMA-SVR’s prediction accuracy was best compared with the other compound algorithms. In addition, KMCM-CSMA-SVR produced the most stable output on D2, D3 and D5 based on the Std data of Table 4 and Fig. 8.

Furthermore, Wilcoxon’s test was applied to verify the statistical significance with respect to the differences of given

forecast results and gain P values with a 5% significance level. The P data of KMCM-CSMA-SVR vs. other models are displayed in Table 5. The P data greater than 0.05 indicate that KMCM-CSMA-SVR has no statistically significant differences compared with other algorithms. The data are underlined in Table 5. From the table, there was no significant difference between the proposed algorithm and HGSO-SVR on 5 datasets (D1, D2, D3, D5 and D6). Compared with SSA-SVR, the number of datasets with no significant difference was 4 (D2-D5). Compared with DA-SVR, there were no significant differences on datasets D1-D3 and D6. Compared with ALO-SVR, the datasets without significant differences were D1, D2 and D4. In particular, compared with PSO-SVR, FA-SVR and KMCM-SVR, this value was 0, which shows that there were significant differences between KMCM-CSMA-SVR and these algorithms. The mentioned outcomes confirmed the ability of CSMA to optimize ϵ -SVR’s parameters regarding forecast accuracy and stability.

C. EXPERIMENT III: COMPUTATION COMPLEXITY EXPERIMENT

The computation complexities of integrated algorithms vary with the instance and attribute number, so calculating them clearly is a challenging job [4]. This experiment set the same

TABLE 3. The *paSe* and *ItemOptimum* results (The minimum *paSe* of each parameter item is underlined).

ID	Item	<i>paSe</i>										<i>ItemOptimum</i>	
D1	<i>K</i>	0.800	0.357	0.361	0.200	0.191	0.327	<u>0.173</u>	0.422	0.200	0.288	35	
	<i>MaxIter</i>	0.482	0.807	0.021	0.408	0.145	0.607	0.123	0.152	0.333	0.562	30	
	<i>nsa</i>	0.800	0.359	<u>0.507</u>	0.070	0.167	0.558	0.443	0.686	0.424	0.622	40	
	<i>z</i>	0.800	0.299	0.673	<u>0.450</u>	0.845	0.315	<u>0.064</u>	0.103	0.167	0.577	0.06	
D2	<i>K</i>	0.800	0.488	0.127	0.103	<u>0.089</u>	0.134	<u>0.127</u>	0.141	0.153	0.203	25	
	<i>MaxIter</i>	0.095	0.157	0.109	<u>0.083</u>	0.124	0.917	0.216	0.208	0.469	0.200	40	
	<i>nsa</i>	0.800	0.072	<u>0.022</u>	0.243	0.189	0.438	0.370	0.200	0.531	0.608	30	
	<i>z</i>	0.100	0.224	0.123	0.248	0.907	0.181	0.566	<u>0.064</u>	0.311	0.243	0.07	
D3	<i>K</i>	0.667	0.211	0.840	0.626	0.749	0.369	0.359	<u>0.162</u>	0.600	0.405	40	
	<i>MaxIter</i>	0.497	0.835	0.355	0.624	0.539	0.340	<u>0.148</u>	0.749	0.614	0.720	70	
	<i>nsa</i>	<u>0.133</u>	0.243	0.233	0.569	0.257	0.331	0.471	0.961	0.200	0.619	10	
	<i>z</i>	<u>0.646</u>	<u>0.000</u>	0.915	0.484	0.513	0.607	0.160	0.297	0.852	0.600	0.01	
D4	<i>K</i>	0.804	<u>0.682</u>	0.806	0.390	<u>0.075</u>	0.798	0.657	0.426	0.568	0.698	25	
	<i>MaxIter</i>	0.444	0.822	0.518	0.495	0.614	0.535	0.571	0.832	<u>0.172</u>	0.751	90	
	<i>nsa</i>	<u>0.025</u>	0.200	0.040	0.185	0.241	0.149	0.407	0.988	<u>0.407</u>	0.489	10	
	<i>z</i>	0.359	0.428	0.054	0.508	0.888	0.481	0.402	0.499	0.351	0.125	0.02	
D5	<i>K</i>	0.683	0.505	<u>0.843</u>	0.831	0.465	0.503	0.643	0.536	0.481	<u>0.200</u>	50	
	<i>MaxIter</i>	0.686	0.648	0.370	0.500	0.335	0.902	0.840	0.541	<u>0.183</u>	0.486	90	
	<i>nsa</i>	0.420	0.817	<u>0.139</u>	0.583	0.365	0.568	0.911	0.163	<u>0.520</u>	0.340	30	
	<i>z</i>	0.233	0.937	0.231	0.462	0.645	0.152	0.055	0.825	<u>0.029</u>	0.328	0.08	
D6	<i>K</i>	0.800	0.402	0.254	0.230	0.303	0.236	<u>0.127</u>	0.310	<u>0.176</u>	0.326	35	
	<i>MaxIter</i>	0.427	<u>0.021</u>	0.752	0.328	0.656	0.346	0.932	0.684	0.585	0.582	20	
	<i>nsa</i>	0.295	<u>0.263</u>	0.845	0.523	0.275	<u>0.117</u>	0.617	0.728	0.625	0.385	60	
	<i>z</i>	0.104	0.227	0.393	0.406	0.405	<u>0.587</u>	0.612	<u>0.095</u>	0.829	0.467	0.07	
D7	<i>K</i>	0.721	0.344	0.804	0.454	0.636	0.184	0.376	<u>0.899</u>	0.632	0.390	30	
	<i>MaxIter</i>	0.800	0.399	<u>0.145</u>	0.464	0.709	0.156	0.914	0.901	0.182	0.798	30	
	<i>nsa</i>	0.501	<u>0.030</u>	0.619	0.801	0.831	0.597	0.425	0.475	0.978	0.666	20	
	<i>z</i>	0.988	0.558	0.626	0.417	0.272	0.517	0.712	0.836	0.312	<u>0.028</u>	0.09	
D8	<i>K</i>	0.800	0.709	0.741	0.626	0.811	0.462	0.839	0.738	0.731	<u>0.200</u>	50	
	<i>MaxIter</i>	0.257	0.811	0.180	0.289	0.312	0.910	<u>0.123</u>	0.361	0.412	0.564	70	
	<i>nsa</i>	0.592	0.819	0.689	0.272	0.224	0.796	<u>0.316</u>	0.885	<u>0.177</u>	0.398	90	
	<i>z</i>	0.957	0.544	1.000	0.247	<u>0.077</u>	0.549	0.356	0.525	0.676	0.208	0.04	

TABLE 4. Forecast accuracy outcomes of each model.

ID	HGSO-SVR	SSA-SVR	DA-SVR	ALO-SVR	PSO-SVR	FA-SVR	KMCM-SVR	KMCM-CSMA-SVR
	Avg±Std	Avg±Std	Avg±Std	Avg±Std	Avg±Std	Avg±Std	Avg±Std	Avg±Std
D1	0.036±0	0.034±0.002	0.037±0.007	0.034±0.002	0.086±0.004	0.086±0.020	0.129±0.007	0.038±0.005
D2	0.000±0.000	0.000±0.000	0.000±0.000	0.000±0.000	0.003±0.000	0.003±0.004	0.007±0.001	0.000±0.000
D3	0.028±0.008	0.094±0.209	0.095±0.196	0.07±0.002	0.054±0.016	0.129±0.192	0.031±0.000	0.028±0.000
D4	0.008±0.000	0.01±0.005	0.013±0.000	0.008±0.003	0.32±0.047	0.32±0.004	0.022±0.006	0.007±0.001
D5	0.025±0.009	0.026±0.013	0.041±0.038	0.043±0.007	0.039±0.010	0.044±0.043	0.026±0.001	0.021±0.000
D6	0.062±0.000	0.058±0.001	0.061±0.029	0.061±0.029	0.173±0.084	0.173±0.037	0.16±0.008	0.063±0.001
D7	0.138±0.003	0.138±0.003	0.141±0.002	0.139±0.003	0.139±0.220	0.139±0.195	0.142±0.015	0.102±0.011
D8	0.102±0.000	0.1±0.002	0.099±0.002	0.1±0.002	0.102±0.000	0.102±0.000	0.232±0.004	0.011±0.008

TABLE 5. *P* values of KMCM-CSMA-SVR vs. other models based on Wilcoxon's test (*P* > 0.05 is underlined).

ID	vs. HGSO-SVR	vs. SSA-SVR	vs. DA-SVR	vs. ALO-SVR	vs. PSO-SVR	vs. FA-SVR	vs. KMCM-SVR
D1	<u>4.73E-01</u>	4.52E-02	<u>6.78E-01</u>	<u>6.40E-02</u>	1.83E-04	1.83E-04	1.83E-04
D2	<u>3.68E-01</u>	<u>3.68E-01</u>	<u>3.68E-01</u>	<u>3.68E-01</u>	1.59E-05	4.17E-05	5.31E-05
D3	<u>4.73E-01</u>	<u>7.34E-01</u>	<u>3.85E-01</u>	1.83E-04	1.63E-04	1.83E-04	1.83E-04
D4	1.42E-03	<u>4.67E-01</u>	1.29E-04	<u>4.67E-01</u>	8.74E-05	8.74E-05	1.83E-04
D5	<u>4.73E-01</u>	<u>1.00E+00</u>	2.57E-02	1.83E-04	2.28E-03	1.83E-04	1.83E-04
D6	<u>4.43E-01</u>	1.74E-04	<u>4.70E-01</u>	2.58E-03	1.54E-04	8.74E-05	1.83E-04
D7	1.83E-04	1.82E-04	1.73E-04	1.83E-04	2.70E-03	2.12E-02	1.83E-04
D8	6.39E-05	1.48E-04	1.45E-04	1.45E-04	6.39E-05	6.39E-05	1.83E-04

conditions for these models and achieved the homologous time cost of D1-D8 for quantifying the complexity. The experiment criteria were identical to those of Experiment II. The cut-off condition was when the forecast accuracy reached the mean of *MSE* for all models in each dataset of Table 4. The time cost was reported. This process was repeated ten

times, and the mean was computed. The shorter the time cost, the lower the computation complexity. The outcomes in seconds are shown in Table 6. PSO-SVR obtained the optimal complexity on dataset D4. KMCM-SVR achieved the minimum on D1-D3 and D5-D8. Eq. (17) is utilized to quantify the general evaluation (*CCR*_{rank}) of the computation

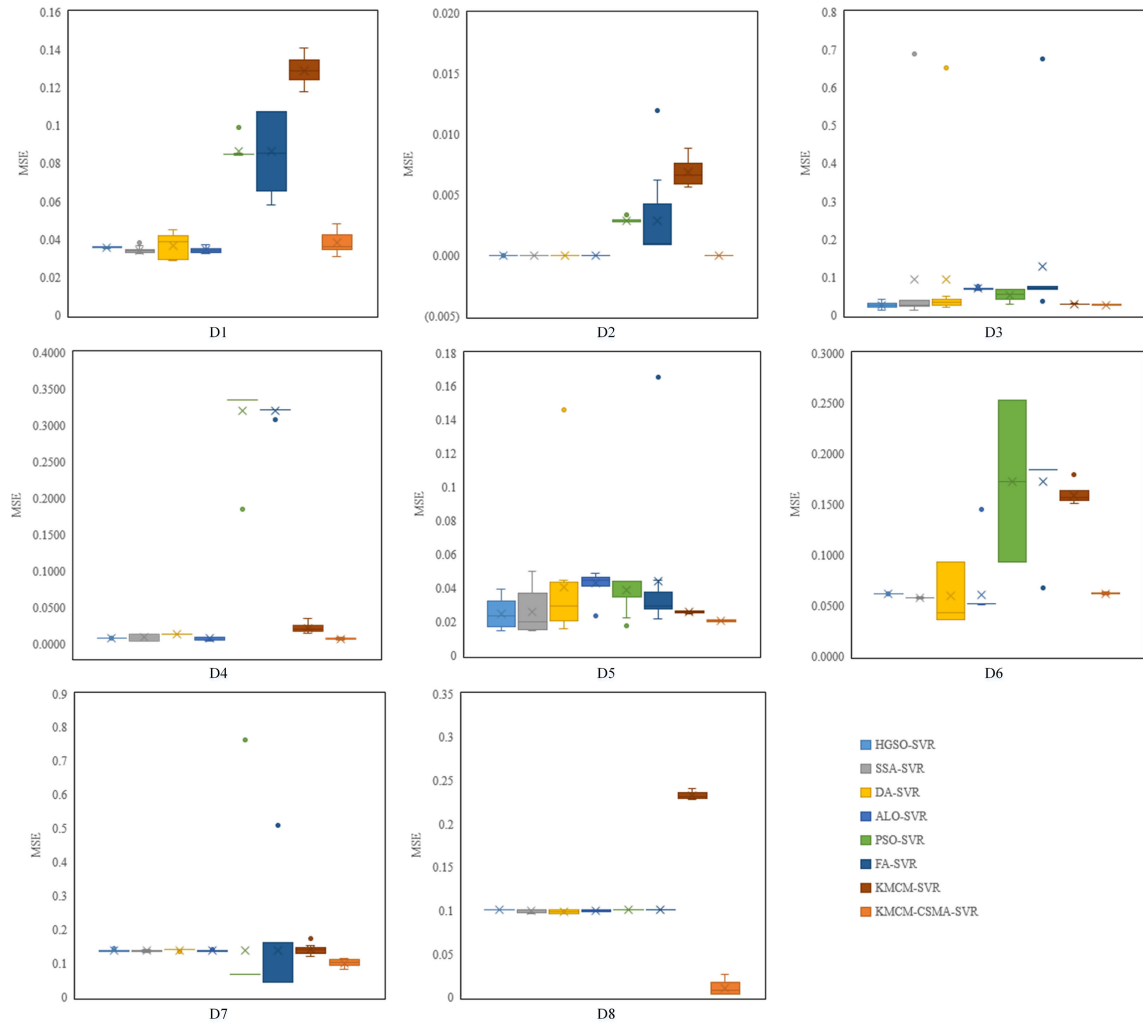


FIGURE 7. Box-plot charts of forecast accuracy for KMCM-CSMA-SVR and other models based on 8 datasets.

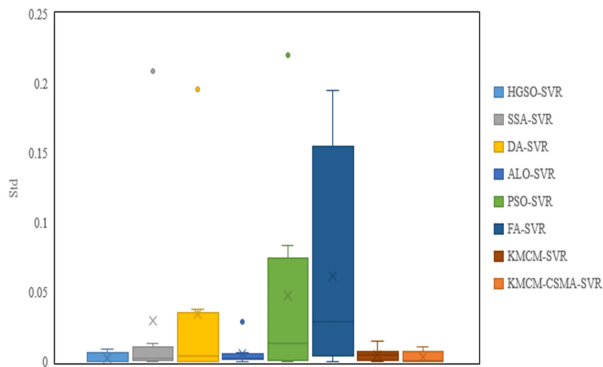


FIGURE 8. The Std data of KMCM-CSMA-SVR vs. other algorithms.

complexity.

$$\begin{aligned}
 CCRank &= \frac{1}{nd} \sum_{i=1}^{nd} CCrate_i \\
 CCrate &= \frac{CCold - CCMin}{CCMax - CCMin}. \tag{17}
 \end{aligned}$$

where $CCrate$ is the normalized value of time cost. $CCold$ is the time cost. $CCMax$ ($CCMin$) denotes the maximum (minimum) time cost for each dataset.

The outcome is displayed in Table 9 ($CCRank$). From the table, KMCM-CSMA-SVR won second place with a score of 0.068 (lower is better). KMCM-SVR had an optimum at 0.000. Other models performed less effectively without KMCM. The abovementioned results confirmed the substantial role of KMCM in reducing computation complexity.

D. EXPERIMENT IV: COMPARISON WITH HIGHLY SIMILAR APPROACHES

In this section, KMCM was fused with HGSO-SVR, SSA-SVR, DA-SVR, ALO-SVR, PSO-SVR and FA-SVR to form KMCM-HGSO-SVR, KMCM-SSA-SVR, KMCM-DA-SVR, KMCM-ALO-SVR, KMCM-PSO-SVR and KMCM-FA-SVR for better comparing the parameter optimization performance with KMCM-CSMA-SVR. The init setting parameters referred to the original independent algorithm, as shown in Table 2 and 3. The ten-fold cross-validation

TABLE 6. The mean cost time of KMCM-CSMA-SVR vs. other models based on 8 datasets.

ID	HGSO-SVR	SSA-SVR	DA-SVR	ALO-SVR	PSO-SVR	FA-SVR	KMCM-SVR	KMCM-CSMA-SVR
D1	108.59	194.06	253.01	94.54	278.03	359.59	0.19	3.48
D2	25031.35	25428.92	26051.45	26150.79	15.43	24.94	0.21	6.11
D3	17873.13	17349.47	16042.59	20586.70	13395.81	12799.56	0.64	17.32
D4	1.65	1.64	3.65	1.48	0.44	1.17	0.45	2.53
D5	18745.95	24135.65	27662.51	25528.37	21728.26	28109.13	0.76	54.43
D6	515.52	210.40	356.85	356.65	144.00	428.76	0.38	9.60
D7	1.75	2.16	3.62	2.42	0.75	6.05	0.05	1.33
D8	104720.71	115192.78	1087392.52	110585.06	20779.18	22025.93	12.03	884.76

TABLE 7. Forecast accuracy results of KMCM-HGSO-SVR and other models.

ID	KMCM-HGSO-SVR Avg±Std	KMCM-SSA-SVR Avg±Std	KMCM-DA-SVR Avg±Std	KMCM-ALO-SVR Avg±Std	KMCM-PSO-SVR Avg±Std	KMCM-FA-SVR Avg±Std
D1	0.112±0.038	0.054±0.012	0.058±0.014	0.054±0.014	0.170±0.075	0.192±0.035
D2	0.004±0.001	0.000±0.000	0.000±0.001	0.000±0.001	0.098±0.310	0.107±0.306
D3	0.077±0.038	0.028±0.001	0.052±0.046	0.028±0.001	0.031±0.002	0.047±0.023
D4	0.145±0.067	0.009±0.001	0.076±0.086	0.015±0.015	0.160±0.000	0.145±0.081
D5	0.039±0.036	0.022±0	0.024±0.003	0.021±0.001	0.098±0.047	0.032±0.005
D6	0.256±0.068	0.065±0.003	0.066±0.003	0.065±0.002	0.220±0.057	0.206±0.048
D7	0.156±0.069	0.102±0.013	0.135±0.062	0.107±0.009	0.134±0	0.288±0.266
D8	0.177±0.022	0.152±0.005	0.148±0.006	0.170±0.016	0.210±0.004	0.183±0.007

TABLE 8. The mean cost time of KMCM-HGSO-SVR and other models.

ID	KMCM-HGSO-SVR	KMCM-SSA-SVR	KMCM-DA-SVR	KMCM-ALO-SVR	KMCM-PSO-SVR	KMCM-FA-SVR
D1	2.22	3.83	0.81	4.57	1.95	2.18
D2	4.62	6.11	11.13	6.68	3.64	5.71
D3	7.99	15.85	14.85	16.15	3.67	10.59
D4	1.89	2.10	3.72	2.12	1.52	1.40
D5	20.47	39.78	58.13	48.17	14.63	36.57
D6	4.40	7.36	10.27	9.42	4.57	8.03
D7	1.41	1.34	2.82	1.30	1.09	1.53
D8	92.68	184.39	264.82	146.40	81.71	98.91

training/testing technology was also adopted to access the forecast accuracy and computation complexity. The test process was similar to those of Experiment II and III. The forecast accuracy results are revealed in Table 7. The computation complexity results are displayed in Table 8. From the tables, the time consumption was greatly reduced for all algorithms. However, compared with the original algorithm, the prediction accuracy exhibited different degrees of attenuation, especially for KMCM-HGSO-SVR, KMCM-PSO-SVR and KMCM-FA-SVR. It was likely to fall into local optima. Considering the prediction accuracy, the proposed algorithm was superior to these approaches on datasets D1 and D3-D8. In terms of computational complexity, the proposed algorithm and these methods offered their own merits. More comparison is shown in the discussion section.

E. DISCUSSION

As a summary for all results of Experiment I-IV, Table 9 denotes the statistical data, which are the numbers of datasets each algorithm defeated/tied/lost to with respect to forecast accuracy, associated P values, etc. It discloses that KMCM-CSMA-SVR is superior to others on 6 datasets in MSE. Concerning significance, the results show that KMCM-CSMA-SVR is better on 3, 2, 4, 4, 8, 8, 8, 8, 5, 8, 6, 8 and 8 datasets compared with HGSO-SVR, SSA-SVR,

DA-SVR, ALO-SVR, PSO-SVR, FA-SVR, KMCM-SVR, KMCM-HGSO-SVR, KMCM-SSA-SVR, KMCM-DA-SVR, KMCM-ALO-SVR, KMCM-PSO-SVR and KMCM-FA-SVR, respectively.

For fully evaluating the algorithms, Eq. (18) was applied to verify extensive performance (Rank) regarding the forecast accuracy, stability and computation complexity. ARank is the normalized win rate of MSE. StdRank is the normalized stability ratio on the Std Data. The lower the Rank, the better the overall performance. The outcomes are shown in Table 9.

$$\begin{aligned}
 Rank &= \frac{1}{3}ARank + \frac{1}{3}StdRank + \frac{1}{3}CCRank \\
 ARank &= \frac{WMax - Wold}{WMax - WMin} \\
 StdRank &= \frac{Sold - SMin}{SMax - SMin}
 \end{aligned}
 \tag{18}$$

where Wold is the win rate concerning MSE. Wold = $\frac{NumW}{NumT}$. NumW indicates the number of datasets each approach defeated on MSE. NumT denotes the total number of datasets. WMax is the maximum of Wold. WMin is the minimum of Wold. Sold indicates the value of each approach on the Std average data. SMax denotes the maximum of Sold. SMin denotes the minimum of Sold [4].

TABLE 9. Statistical results summary of KMCM-CSMA-SVR and other models.

	HGSO-SVR			SSA-SVR			DA-SVR			ALO-SVR			PSO-SVR			FA-SVR			KMCM-SVR		
	W	T	L	W	T	L	W	T	L	W	T	L	W	T	L	W	T	L	W	T	L
MSE	1	0	7	3	0	4	1	0	7	1	0	7	0	0	8	0	0	8	0	0	8
P value	0	5	3	2	4	2	0	4	4	1	3	4	0	0	8	0	0	8	0	0	8
ARank	0.833			0.500			0.833			0.833			1.000			1.000			1.000		
StdRank	0.000			0.183			0.216			0.024			0.307			0.404			0.019		
CCRank	0.543			0.531			0.776			0.563			0.326			0.572			0.000		
Rank	0.459			0.405			0.608			0.474			0.545			0.659			0.340		
	KMCM-HGSO-SVR			KMCM-SSA-SVR			KMCM-DA-SVR			KMCM-ALO-SVR			KMCM-PSO-SVR			KMCM-FA-SVR			KMCM-CSMA-SVR		
	W	T	L	W	T	L	W	T	L	W	T	L	W	T	L	W	T	L	W	T	L
MSE	0	0	8	1	0	7	0	0	8	0	0	8	0	0	8	0	0	8	6	0	2
P value	0	0	8	0	3	5	0	0	8	0	2	6	0	0	8	0	0	8	N/A	N/A	N/A
ARank	1.000			0.833			1.000			1.000			1.000			1.000			0.000		
StdRank	0.227			0.024			0.220			0.038			0.715			1.000			0.005		
CCRank	0.090			0.097			0.186			0.098			0.070			0.076			0.068		
Rank	0.439			0.318			0.469			0.379			0.595			0.692			0.024		

From the table, KMCM helped SSA-SVR, DA-SVR and ALO-SVR to improve the total score. However, it reduced the scores of HGSO-SVR, PSO-SVR and FA-SVR. It can be found that KMCM is not beneficial to all algorithms. KMCM-CSMA-SVR obtained the optimum of 0.024. This indicates that the proposed algorithm offers the optimal comprehensive performance. Due to CSMA, KMCM-CSMA-SVR has the optimum prediction accuracy on 6 datasets. It has the optimal stability on datasets D2, D3 and D5, and low computation complexity on almost all datasets. This results from the KMCM. It is proven that KMCM-CSMA-SVR exhibits high effectiveness and efficiency and is superior in handling high-dimensional data. Considering prediction accuracy, convergence stability and computation complexity, KMCM-CSMA-SVR shows very strong competitiveness. Thus, KMCM and CSMA are worthy of application in the field of SVR parameter optimization.

V. CONCLUSION AND FUTURE DIRECTIONS

The study proposes a new hybrid algorithm for optimizing SVR parameters using KMCM and CSMA. KMCM is applied to speed up the whole algorithm. CSMA is utilized to improve prediction accuracy. The proposed KMCM-CSMA-SVR achieves the adaptive optimization of SVR parameters. The experiments of key parameter settings, forecast accuracy and computation complexity were performed to compare HGSO-SVR, SSA-SVR, DA-SVR, ALO-SVR, PSO-SVR, FA-SVR and KMCM-SVR. The comparison experiment with KMCM-HGSO-SVR, KMCM-SSA-SVR, KMCM-DA-SVR, KMCM-ALO-SVR, KMCM-PSO-SVR and KMCM-FA-SVR was also implemented. The results on the benchmark datasets revealed KMCM-CSMA-SVR’s best prediction performance compared with other famous algorithms. It can also effectively reduce the running cost of this kind of algorithm and allow it to no longer rely on high-performance computers. This work validated the tremendous potential of the KMCM-CSMA-SVR in the field of prediction.

For future study, more big data experiments and real-world engineering optimization problems should be added to test the proposed approach. Moreover, key parameters of KMCM and CSMA exert important effects on clustering and prediction performance. Their adaptive adjustment is a field worth studying. The structure of the whole approach is complicated and can be appropriately simplified in the future.

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] L. Shen, H. Chen, Z. Yu, W. Kang, B. Zhang, H. Li, B. Yang, and D. Liu, “Evolving support vector machines using fruit fly optimization for medical data classification,” *Knowl.-Based Syst.*, vol. 96, pp. 61–75, Mar. 2016.
- [3] M. Wang and H. Chen, “Chaotic multi-swarm whale optimizer boosted support vector machine for medical diagnosis,” *Appl. Soft Comput.*, vol. 88, Mar. 2020, Art. no. 105946.
- [4] W. Cao, X. Liu, and J. Ni, “Parameter optimization of support vector regression using Henry gas solubility optimization algorithm,” *IEEE Access*, vol. 8, pp. 88633–88642, May 2020.
- [5] 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.
- [6] 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,” *Energy Sources A, Recovery, Utilization, Environ. Effects*, pp. 1–14, Jun. 2019.
- [7] M. Amroune, I. Musirin, T. Bouktir, and M. M. Othman, “The amalgamation of SVR and ANFIS models with synchronized phasor measurements for on-line voltage stability assessment,” *Energies*, vol. 10, no. 11, pp. 1–18, Oct. 2017.
- [8] 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.
- [9] 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.
- [10] Z. Hao, W. Wen, X. Yang, J. Lu, and G. Zhang, “A fast data preprocessing procedure for support vector regression,” in *Intelligent Data Engineering and Automated Learning (Lecture Notes in Computer Science)*, vol. 4224. Berlin, Germany: Springer, 2006, pp. 48–56.
- [11] C.-J. Lu and C.-C. Chang, “A hybrid sales forecasting scheme by combining independent component analysis with K-Means clustering and support vector regression,” *Sci. World J.*, vol. 2014, Jun. 2014, Art. no. 624017.

- [12] R. J. Kuo and P. S. Li, "Taiwanese export trade forecasting using firefly algorithm based K-means algorithm and SVR with wavelet transform," *Comput. Ind. Eng.*, vol. 99, pp. 153–161, Sep. 2016.
- [13] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Gener. Comput. Syst.*, vol. 111, pp. 300–323, Apr. 2020.
- [14] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Syst. Mag.*, vol. 22, no. 3, pp. 52–67, Jun. 2002.
- [15] X. Xu and H.-L. Chen, "Adaptive computational chemotaxis based on field in bacterial foraging optimization," *Soft Comput.*, vol. 18, no. 4, pp. 797–807, Apr. 2014.
- [16] H. Chen, Q. Zhang, J. Luo, Y. Xu, and X. Zhang, "An enhanced bacterial foraging optimization and its application for training kernel extreme learning machine," *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art. no. 105884.
- [17] M. Dorigo and C. Blum, "Ant colony optimization theory: A survey," *Theor. Comput. Sci.*, vol. 344, nos. 2–3, pp. 243–278, Nov. 2005.
- [18] X. Zhao, D. Li, B. Yang, C. Ma, Y. Zhu, and H. Chen, "Feature selection based on improved ant colony optimization for online detection of foreign fiber in cotton," *Appl. Soft Comput.*, vol. 24, pp. 585–596, Nov. 2014.
- [19] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016.
- [20] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017.
- [21] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
- [22] X. Zhao, X. Zhang, Z. Cai, X. Tian, X. Wang, Y. Huang, H. Chen, and L. Hu, "Chaos enhanced grey wolf optimization wrapped ELM for diagnosis of paraquat-poisoned patients," *Comput. Biol. Chem.*, vol. 78, pp. 481–490, Feb. 2019.
- [23] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015.
- [24] M. Wang, H. Chen, B. Yang, X. Zhao, L. Hu, Z. Cai, H. Huang, and C. Tong, "Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses," *Neurocomputing*, vol. 267, pp. 69–84, Dec. 2017.
- [25] Y. Xu, H. Chen, J. Luo, Q. Zhang, S. Jiao, and X. Zhang, "Enhanced moth-flame optimizer with mutation strategy for global optimization," *Inf. Sci.*, vol. 492, pp. 181–203, Aug. 2019.
- [26] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.
- [27] 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.
- [28] 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, Sep. 2010.
- [29] S. Li, H. Fang, and X. Liu, "Parameter optimization of support vector regression based on sine cosine algorithm," *Expert Syst. Appl.*, vol. 91, pp. 63–77, Jan. 2018.
- [30] 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.
- [31] H. Li, C. Li, and Y. Liu, "Maximum frequency deviation assessment with clustering based on metric learning," *Int. J. Electr. Power Energy Syst.*, vol. 120, Sep. 2020, Art. no. 105980.
- [32] 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.
- [33] S. Mirjalili and A. H. Gandomi, "Chaotic gravitational constants for the gravitational search algorithm," *Appl. Soft Comput.*, vol. 53, pp. 407–419, Apr. 2017.
- [34] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010.
- [35] M. Lichman. (2013). *UCI Machine Learning Repository*. [Online]. Available: <http://archive.ics.uci.edu/ml>



ZIYI CHEN (Graduate Student Member, IEEE) received the master's degree from Shanghai Maritime University, in 2015, where he is currently pursuing the Ph.D. degree.

His major research interests include machine learning and civil economics and management.



WENBAI LIU received the Ph.D. degree in geotechnical engineering from Tongji University, Shanghai, China, in 2003.

He is currently working as a Professor with Shanghai Maritime University. His research interests include economy and management of transportation engineering.

• • •