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

ZIYI CHEN, (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.

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 $\mathbf{X}$ consists of individuals
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.

**A. CHAOTIC SLIME MOULD ALGORITHM**

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

\[
\bar{X}(t+1) = \begin{cases} \bar{X}_B(t) + \frac{\bar{b}}{C} \left( \bar{W} \cdot \bar{X}_A(t) - \bar{X}_B(t) \right), & C < p \\ \bar{v} \cdot \bar{X}(t), & C \geq p \end{cases}
\]

(1)

where \(\bar{b}\) is a parameter. Its range is \([-a,a]\). \(a\) is listed in Eq. (2). \(\bar{v}\) reduces linearly from 1 to 0. \(t\) is the current generation. \(\bar{X}_B\) is the individual site where the highest concentration of odour has been found. \(\bar{X}_B\) indicates the slime mould’s location (i.e., a set of SVR parameters). \(\bar{X}_A\) and \(\bar{X}_B\) are 2 individuals randomly elected from the swarm. \(\bar{W}\) 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).

\[
a = \arctan h \left( -\left( t/\text{max}_t \right) + 1 \right)
\]

(2)

where \(\text{max}_t\) represents the maximum number of iterations.

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

(3)
where \textit{condit} is \( S (i) \) in the top half of the population. \( bF \) (\( wF \)) represents the optimal (worst) fitness of the current iterative procedure. \textit{SmellIndex} is the ranked sequence of fitness data.

\[
\text{SmellIndex} = \text{sort} (S) \\
p = \tanh \left| S (i) - DF \right| 
\]

where \( i \in 1, 2, \cdots, n \). \( S (i) \) indicates the fitness of \( X \). \( DF \) is the optimum acquired in all generations.

\[
C_{t+1} = 4C_t (1 - C_t) 
\]

where the range of \( C \) is \([0,1]\)

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

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

where \( U \) and \( L \) are the upper and lower boundaries of the search scope. \textit{rand} represents the randomness in \([0,1]\).

\textbf{Grabble Food:} \( \vec{v} \) floats randomly in \([-a,a] \) and approaches 0 with the increase of iterations. The \( \vec{V} \) 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 \( X \) based on \( X_{opt} \) according to the loop body of the following pseudo-code. The updated \( X \) is used as the input parameters of SVR to make a new prediction and obtain the new \( X_{opt} \). 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 popsize, max_t; Initialize the positions of slime mould \( X_i (i=1,2,\ldots,n) \);

\textbf{While} \((t \leq \text{max}_t)\) \textbf{do}:

- Calculate the fitness of all slime mould;
- Update bestFitness, \( X_b \);
- Calculate the \( W \) by Eq. (3);
- For each search portion
  - Update \( p, v_b, v_c \);
  - Update positions by Eq. (7);
- \textbf{end};
- \( t = t + 1 \);
\textbf{end While};

Return bestFitness, \( X_b \)

\textbf{FIGURE 2.} Pseudocode of CSMA’s operations.

\section*{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 \( Y = \{y_1, y_2, \cdots, y_m\} \) be the set of \( m \) d-dimensional points to be polymerized into a set of \( K \) clusters, \( Cl = \{c_1, c_2, \cdots, c_K\} \). KMCM seeks out a partition to minimize the square error (SE) of the empirical mean and the points in the cluster. Let \( \mu_i \) be the average of cluster \( c_i, i = 1, 2, \cdots, K \). The SE between \( \mu_i \) and the points in cluster \( c_i \) is calculated below.

\[
J (c_i) = \sum_{y_j \in c_i} \left\| y_j - \mu_i \right\| 
\]

The objective of KMCM is minimization of the sum of the SE of all \( K \) clusters.

\[
J (Cl) = \sum_{j=1}^{K} \sum_{y_j \in c_i} \left\| y_j - \mu_i \right\| 
\]

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 \([34]\).

\textbf{Step 1:} Elect an original segmentation using \( K \) clusters; iterate steps 2 and 3 until cluster membership stabilizes.

\textbf{Step 2:} An updated segmentation is generated by assigning each pattern to its nearest cluster centre.

\textbf{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).

\section*{C. \( \epsilon \)-SUPPORT VECTOR REGRESSION}

The \( \epsilon \)-SVR prediction method was developed by Vapnik \([1]\).

Consider training samples \( \{(x_{t1}, z_{t1}), \cdots, (x_{tl}, z_{tl})\} \), where \( x_{ti} \in R^d \) is a feature vector and \( z_{ti} \in R \) is the objective. \( \epsilon > 0 \) and the penalty factor \( PF > 0 \). \( \epsilon \)-SVR is described as

\[
\begin{aligned}
&\min_{w, b, \xi, \xi^*} \frac{1}{2} w^T w + PF \sum_{i=1}^{l} \xi_i + \sum_{i=1}^{l} \xi_i^* \\
&s.t. \quad \phi(x_{ti}) + b - z_{ti} \leq \epsilon + \xi_i^* \\
&\quad \phi(x_{ti}) - b \leq \epsilon + \xi_i^* \\
&\quad \xi_i, \xi_i^* \geq 0, i = 1, \cdots, l. 
\end{aligned}
\]

The dual form is exhibited below.

\[
\begin{aligned}
&\min_{\alpha, \alpha^*} \frac{1}{2} \sum_{i,j=1}^{l} (\alpha_i - \alpha^*_{i})^T Q (\alpha - \alpha^*) + \sum_{i=1}^{l} (\alpha_i + \alpha_i^*) \\
&s.t. \quad \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) = 0 \\
&\quad 0 \leq \alpha_i, \alpha_i^* \leq PF \\
&\quad i = 1, \cdots, l. 
\end{aligned}
\]
where $Q_{ij} = K_T(x_t_i, x_t_j) = \phi(x_t_i)^T \phi(x_t_j)$. $e = [1, 1, \cdots, 1]^T$.

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

1) Linear kernel:

\[
K_T(\cdot) = u^T \cdot v. \tag{12}
\]

2) Polynomial kernel:

\[
K_T(\cdot) = [\gamma (u^T \cdot v) + \text{coef}]^d. \tag{13}
\]

3) RBF kernel:

\[
K_T(\cdot) = \exp(-\gamma |u - v|^2). \tag{14}
\]

4) Sigmoid kernel:

\[
K_T(\cdot) = \tanh(\gamma (u^T \cdot v) + \text{coef}). \tag{15}
\]

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 $K_T, PF, \gamma$ and $e$, 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.

| Algorithm | Item | Value |
|-----------|------|-------|
| HSRO-SVR  | Number of gas types | 5     |
|            | $K$ | 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) |
| DA-SVR    | Separation coefficient | Random number from 0 to 0,192 |
|            | Alignment coefficient | Random number from 0 to 0,192 |
|            | Cohesion coefficient | Random number from 0 to 0,192 |
|            | Food attraction coefficient | Random number from 0 to 2 |
|            | Energy dissipation coefficient | 0,096 |
|            | Parameter minimum of SVR | (2,0,0001,0,0) |
|            | Parameter maximum of SVR | (3,50,50,50) |
| ALO-SVR   | Selection mechanism | Roulette wheel |
|            | Parameter minimum of SVR | (2,0,0001,0,0) |
|            | Parameter maximum of SVR | (3,50,50,50) |
| PSO-SVR   | Personal learning coefficient | 1,5 |
|            | Global learning coefficient | 2,0 |
|            | 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) |
| FA-SVR    | Coefficient $\alpha$ | 0,5 |
|            | Attractiveness coefficient | 0,2 |
|            | Absorption coefficient | 1 |
|            | Parameter minimum of SVR | (2,0,0001,0,0) |
|            | Parameter maximum of SVR | (3,50,50,50) |
| KMCM-SVR  | Cluster initialization | Randomly generated |
|            | Distance metric | Euclidean metric |
|            | SVR parameters | (2,1,1/Number of features, 0,1) |
| KMCM-CSMA-SVR | Cluster initialization | Randomly generated |
|            | Distance metric | Euclidean metric |
|            | Chaotic map | Logistic map |
|            | Parameter minimum of SVR | (2,0,0001,0,0) |
|            | Parameter maximum of SVR | (3,50,50,50) |

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 | 72000          | 49              | Real             |
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) \(\text{MaxIter}\): The effect of \(\text{MaxIter}\) 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 \(\text{MaxIter}\), the values of running time all exhibited an upward trend. D3 behaved differently, and the value declined under \(\text{MaxIter} = 100\).

3) \(\text{nsa}\): This parameter must be sufficient for detecting the search range. The influence was studied when \(\text{nsa}\) 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 \(\text{nsa} = 100\), 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 \(K\), \(\text{MaxIter}\), \(\text{nsa}\) and \(z\) and taking into account two major metrics: \(MSE\) and running time. The smaller \(\text{paSe}\), the better. The prediction accuracy is examined more than running time in this paper. \(\beta\) is set to 0.8, while \(\delta\) is 0.2. The \(\text{paSe}\) outcomes are displayed in Table 3, where \(\text{ItemOptimum}\) indicates the appropriate values of these parameters for datasets. From the table, D1-D8 have different \(\text{ItemOptimum}\). This makes the prediction result superior.

\[
\text{paSe} = \beta \cdot N_{MSE} + \delta \cdot N_{RT}. 
\]
where $\beta$ and $\delta$ are combining coefficients. $\beta + \delta = 1$. $N_{\text{MSE}}$ and $N_{\text{RT}}$ 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 $\epsilon$-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       | $paSe$ | $ItemOptimum$ |
|----|------------|--------|---------------|
| D1 | $MaxIter$  | 0.800  | 0.361         |
|    |            | 0.200  | 0.191         |
|    | $nsa$      | 0.800  | 0.037         |
|    |            | 0.372  | 0.327         |
| D2 | $MaxIter$  | 0.800  | 0.488         |
|    |            | 0.022  | 0.123         |
|    | $nsa$      | 0.800  | 0.022         |
|    |            | 0.488  | 0.123         |
| D3 | $K$        | 0.667  | 0.211         |
|    |            | 0.840  | 0.626         |
|    | $MaxIter$  | 0.497  | 0.835         |
|    |            | 0.355  | 0.624         |
| D4 | $K$        | 0.645  | 0.000         |
|    |            | 0.915  | 0.484         |
|    | $MaxIter$  | 0.484  | 0.000         |
|    |            | 0.915  | 0.484         |

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 |
|-----|----------|---------|--------|---------|---------|--------|----------|----------------|
| D1  | 0.364 ± 0.002 | 0.037 ± 0.002 | 0.034 ± 0.002 | 0.038 ± 0.004 | 0.031 ± 0.003 | 0.042 ± 0.010 | 0.042 ± 0.010 | 0.038 ± 0.004 |
| D2  | 0.000 ± 0.000 | 0.000 ± 0.000 | 0.000 ± 0.000 | 0.000 ± 0.000 | 0.000 ± 0.000 | 0.000 ± 0.000 | 0.000 ± 0.000 | 0.000 ± 0.000 |
| D3  | 0.028 ± 0.008 | 0.094 ± 0.209 | 0.095 ± 0.196 | 0.070 ± 0.002 | 0.054 ± 0.016 | 0.129 ± 0.192 | 0.129 ± 0.192 | 0.030 ± 0.008 |
| D4  | 0.000 ± 0.000 | 0.000 ± 0.000 | 0.000 ± 0.000 | 0.000 ± 0.000 | 0.000 ± 0.000 | 0.000 ± 0.000 | 0.000 ± 0.000 | 0.000 ± 0.000 |
| 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 | 0.044 ± 0.043 | 0.026 ± 0.001 |
| D6  | 0.062 ± 0.007 | 0.058 ± 0.001 | 0.061 ± 0.029 | 0.061 ± 0.029 | 0.173 ± 0.084 | 0.173 ± 0.084 | 0.173 ± 0.084 | 0.063 ± 0.001 |
| D7  | 0.138 ± 0.003 | 0.138 ± 0.003 | 0.138 ± 0.003 | 0.138 ± 0.003 | 0.138 ± 0.003 | 0.138 ± 0.003 | 0.138 ± 0.003 | 0.138 ± 0.003 |
| D8  | 0.102 ± 0.002 | 0.102 ± 0.002 | 0.102 ± 0.002 | 0.102 ± 0.002 | 0.102 ± 0.002 | 0.102 ± 0.002 | 0.102 ± 0.002 | 0.102 ± 0.002 |

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

| ID  | HGSO-SVR vs. SSA-SVR | SSA-SVR vs. DA-SVR | DA-SVR vs. ALO-SVR | ALO-SVR vs. PSO-SVR | PSO-SVR vs. FA-SVR | FA-SVR vs. KMCM-SVR | KMCM-SVR vs. KMCM-CSMA-SVR |
|-----|-----------------------|---------------------|--------------------|----------------------|-------------------|---------------------|----------------------------|
| D1  | 0.036 ± 0.002         | 0.037 ± 0.002       | 0.034 ± 0.002      | 0.034 ± 0.002        | 0.036 ± 0.004     | 0.036 ± 0.004       | 0.038 ± 0.005             |
| D2  | 0.000 ± 0.000         | 0.000 ± 0.000       | 0.000 ± 0.000      | 0.000 ± 0.000        | 0.000 ± 0.000     | 0.000 ± 0.000       | 0.000 ± 0.000             |
| D3  | 0.028 ± 0.008         | 0.094 ± 0.209       | 0.095 ± 0.196      | 0.070 ± 0.002        | 0.054 ± 0.016     | 0.129 ± 0.192       | 0.129 ± 0.192             |
| D4  | 0.000 ± 0.000         | 0.000 ± 0.000       | 0.000 ± 0.000      | 0.000 ± 0.000        | 0.000 ± 0.000     | 0.000 ± 0.000       | 0.000 ± 0.000             |
| 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       | 0.044 ± 0.043             |
| D6  | 0.062 ± 0.007         | 0.058 ± 0.001       | 0.061 ± 0.029      | 0.061 ± 0.029        | 0.173 ± 0.084     | 0.173 ± 0.084       | 0.173 ± 0.084             |
| D7  | 0.138 ± 0.003         | 0.138 ± 0.003       | 0.138 ± 0.003      | 0.138 ± 0.003        | 0.138 ± 0.003     | 0.138 ± 0.003       | 0.138 ± 0.003             |
| D8  | 0.102 ± 0.002         | 0.102 ± 0.002       | 0.102 ± 0.002      | 0.102 ± 0.002        | 0.102 ± 0.002     | 0.102 ± 0.002       | 0.102 ± 0.002             |

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 ($CCRank$) of the computation.
complexity.

\[
CCRank = \frac{1}{nd} \sum_{i=1}^{nd} CCrate_i
\]

\[
CCrate = \frac{CCold - CCMin}{CCMax - CCMin}
\]

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
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.

\[
ARank = \frac{WMax - Wold}{WMax - WMin},
\]

\[
StdRank = \frac{Sold - SMin}{SMax - SMin},
\]

\[
Rank = \frac{1}{3}ARank + \frac{1}{3}StdRank + \frac{1}{3}CCRank
\]

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].
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.

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.

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.
[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. Saadah, “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.