A new adaptive algorithm for v-support vector regression with feature selection using Harris hawks optimization algorithm

Omar Mohammed Ismael¹, Omar Saber Qasim¹, and Zakariya Yahya Algamal²
¹ Department of Mathematics, University of Mosul, Mosul, Iraq
² Department of Statistics and Informatics, University of Mosul, Mosul, Iraq

E-mail: omar_al_habash@uomosul.edu.iq

Abstract. Support vector regression, especially, v-support vector regression (v-SVR) has been applied in several real problems. However, it is usually needed to tune manually the hyperparameter. In addition, v-SVR cannot perform feature selection. Nature-inspired algorithms were used as a feature selection and as an estimation for hyperparameter. In this paper, the Harris hawks optimization algorithm (HHOA) is proposed to optimize the hyperparameter of the v-SVR with embedding the feature selection simultaneously. Experimental results, obtained by running on two datasets, show that our proposed algorithm performs better than other methods, in terms of prediction, number of selected features, and running time. In addition, the HHOA’s experimental results confirm the efficiency of the proposed algorithm in improving prediction performance and computational time compared to other nature-inspired algorithms, which show case HHOA’s ability to search for the best hyperparameter values and to select the most informative features for prediction tasks. Therefore the HHOA may likely be ideal for defining the data relationship between input features and the target variable as opposed to other algorithms. In other real applications this is highly effective in making predictions.
1. Introduction

The support vector machine technique has appeared which is written briefly (SVM). Due to its theoretical and practical advantages that explain its improved performance in classification and regression, support vector machine has attracted considerable attention from many researchers and practitioners and statisticians in recent years [1]. Originally, SVM firstly used for solving classification problem. With the introduction of Vapnik’s $\varepsilon$-insensitive loss function, the SVM has been extended to solve the nonlinear regression problem, called the support vector regression (SVR) [1-3].

SVR has three advantages: (1) ensuring convergence to optimal solutions, because quadratic programming is used with linear constraints for learning the data. (2) The kernel mapping is computationally efficient for modeling nonlinear relationships. And, (3) a superior performance in generalization (lower error rates on test set) [4].

The $\nu$-support vector regression is a new class of SVR which was introduced by Schölkopf, et al. [5]. In this class, the parameter $\nu$ is introduced to control the number of support vectors and training error. The computational efficient of the $\nu$-support vector regression is strongly depending on a number of hyperparameter and influences that directly or indirectly affect the finding optimal solution. The comprehensive grid search is usually employing to investigate all the combinations of the hyperparameter and cross-validation are used to check predictive ability of SVR [6]. Despite the excellent characteristics of SVR, there are still several drawbacks, including the selection of features. In other words, SVR cannot perform feature selection [7].
Selecting a minimal number of features in the regression problems reduces computational complexity. An optimal choice of features is essential for building compact regression models and maintaining accuracy [8].

Nature-inspired algorithms, which they developed by drawing inspiration from nature, have attracted considerable interest and achieved competitive results when solving optimization problems including hyperparameter tuning problem [9-11] and feature selection. In the literature, there are numerous studies on tuning the hyperparameter of SVR using nature-inspired algorithms, such as [11-20]. In recent days, researchers are developing several new nature-inspired algorithms for improving and enhancing exploration and exploitation of the existing algorithms. Among these new algorithms, Harris hawks optimization algorithm which has gained popularity due to their high efficiency [21].

To the best of our knowledge, there are no attempts to perform feature selection and to tune the hyperparameter of SVR simultaneously, especially, for v-support vector regression. The main objective of this work is to optimize the hyperparameter of the v-support vector regression with embedding the feature selection simultaneously by employing the Harris hawks algorithm. The efficiency of our proposed algorithm was compared with other previous algorithms.

The reminder of our paper is organized as follows: Section 2 presents the theoretical part of the v-support vector regression. Sections 3 and 4 cover the details of the Harris hawks algorithm and our proposed algorithms. The experimental results are presented in Section 5. Conclusion is provided in Section 6.
2. v-Support vector regression

Support vector machines (SVM) were successfully used to solve the various classification problems. Nonetheless, with Vapnik implementing ε-insensitive loss function [22], the SVM has been expanded to solve the nonlinear regression problems, which is called the support vector regression. The goal in SVR is a quantitative variable, for instance chemical activity [22], spectral analysis [4, 23], and stock price forecasting.

Given a training dataset of \( n \) observations \( \{(x_i, y_i)\}_{i=1}^{n} \), where \( x_i = (x_{i1}, x_{i2}, \ldots, x_{ip}) \in \mathbb{R}^p \) represents a vector of the \( i \)th feature, \( y_i \in \mathbb{R} \) for \( i = 1, \ldots, n \) is the target variable, which is a quantitative variable, and ε-insensitive loss function, the SVR can be obtained through solving the following optimization problem

\[
\min_{w, b, \zeta, \xi} \left\{ \frac{1}{2} w^T w + C \sum_{i=1}^{n} (\zeta_i + \xi_i) \right\}
\]

\[
\text{S.T.} \quad \begin{cases} y_i - (w^T \phi(x_i) + b) \leq \epsilon + \xi_i \\ (w^T \phi(x_i) + b) - y_i \leq \epsilon + \zeta_i \\ \zeta_i, \xi_i \geq 0, \end{cases}
\]

where \( C > 0 \) is a penalized parameter that controls the tradeoff between the model complexity and training error, \( \zeta_i \) and \( \xi_i \) are slack variables, \( \phi(x_i) \) is a nonlinear mapping which is induced by a kernel function, \( w \) is the weight vector and \( b \) is bias.

Then, Eq. (1) can be solved by the Lagrangian multipliers after reformulated it into its dual problem as
\[
\begin{align*}
\min_{\alpha, \alpha'} \frac{1}{2} \sum_{i,j=1}^{n} (\bar{\alpha}_i - \alpha_i)(\bar{\alpha}_j - \alpha_j)K(x_i, x_j) + \varepsilon \sum_{i=1}^{n} (\bar{\alpha}_i - \alpha_i) - \sum_{i=1}^{n} y_i (\bar{\alpha}_i - \alpha_i) \\
\text{S.T.} & \quad \sum_{i=1}^{n} (\alpha_i - \bar{\alpha}_i) = 0 \\
& \quad 0 \leq \alpha_i, \bar{\alpha}_i \leq C,
\end{align*}
\]

where \( K(x_i, x_j) \) stands for kernel mapping, and \( \alpha_i, \bar{\alpha}_i \) are Lagrangian multipliers. The regression hyperplane for the underlying regression problem is then given by

\[
y_i = f(x_i) = \sum_{x_i \in SV} (\bar{\alpha}_i + \alpha_i)K(x_i, x_j) + b,
\]

where SV is the support vectors set.

\( \nu \)-SVR is a new category of promising nonlinear kernel that goal to find the best regression hyperplane with smallest structural risk in high dimensional feature space [13]. In the SVR with \( \varepsilon \)-insensitive loss function, the number of support vectors cannot be controlled [24]. In order to improve the solution speed of the SVR by controlling the number of support vector, training errors and giving an estimate of the \( \varepsilon \) in the data, an improved version \( \nu \)-SVR was proposed by Schölkopf, et al. [5]. In below there are a figure explain the support vectors and error [25].

![Figure 1 \( \nu \)- Support vector regression](image-url)
We refer to some the differences between SVR and v-SVR. Where the \( v \) parameter \( v \) is used in v-SVR to calculate the proportion of the number of support vectors that you want to retain in your solution with respect to the total number of samples in the dataset. The parameter \( \varepsilon \) is applied to the optimization problem formulation in v-SVR and is automatically (optimally) calculated. While in \( \varepsilon \)-SVR, you have no influence about how many data vectors become support vectors from the dataset, it could be a few, it could be many. The distinction between \( \varepsilon \)-SVR and v-SVR is how the issue of training is parameterized. You may use the \( v \) parameter in v-SVM to control the quantity of support vectors in the resulting model. The exact same problem is solved, given suitable parameters [26].

The original problem in v-SVR leads to convex quadratic programming with inequality constraints as [24, 27-32]

\[
\begin{align*}
\min_{w, b} \left\{ \frac{1}{2} w^T w + C \left[ v \varepsilon + \frac{1}{n} \sum_{i=1}^{n} (\zeta_i + \tilde{\zeta}_i) \right] \right\} \\
\text{S.T.} \quad & y_i - (w \cdot \phi(x_i) + b) \leq \varepsilon + \tilde{\zeta}_i \\
& (w \cdot \phi(x_i) + b) - y_i \leq \varepsilon + \zeta_i \\
& \zeta_i, \tilde{\zeta}_i \geq 0, \varepsilon \geq 0,
\end{align*}
\]

where \( 0 \leq \nu \leq 1 \). It was proved that \( \nu \) is an upper bound on the fraction of margin errors and a lower bound of the fraction of support vectors by Schölkopf, et al. [5]. Furthermore, by introducing \( \nu \) and adding an inequality constraint \( \varepsilon \geq 0 \) in Eq. (4) the value of \( \nu \) can be automatically determined and \( \nu \) is easier to determine than \( \varepsilon \) [5].

Equation (4) can be solved by the Lagrangian multipliers after reformulated it into its dual problem as follows:
\[ L(w, b, \varepsilon, \zeta, \xi) = \frac{1}{2} w^T w + C (\varepsilon + \frac{1}{n} \sum_{i=1}^{n} (\zeta_i + \xi_i) - \sum_{i=1}^{n} \theta_i \zeta_i - \sum_{i=1}^{n} \theta_i \xi_i - \gamma \varepsilon) \]
\[ + \sum_{i=1}^{n} \alpha_i (w^T \phi(x_i) + b - y_i - \varepsilon - \zeta_i) + \sum_{i=1}^{n} \alpha_i (w^T \phi(x_i) + b - y_i - \varepsilon - \zeta_i), \]

where \( \alpha_i, \tilde{\alpha}_i, \theta_i, \tilde{\theta}_i, \gamma \geq 0 \) are Lagrange multipliers. The solution of Eq. (5) can be achieved by partially differentiating with respect to \( \zeta_i, w, b, \varepsilon \), and \( \xi_i \) as

\[
\begin{align*}
\frac{\partial L}{\partial w} &= w + \sum_{i=1}^{n} \alpha_i x_i - \sum_{i=1}^{n} \tilde{\alpha}_i x_i = 0 \\
\frac{\partial L}{\partial b} &= \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \tilde{\alpha}_i = 0 \\
\frac{\partial L}{\partial \varepsilon} &= C \frac{n}{n} - \gamma - \sum_{i=1}^{n} (\alpha_i + \tilde{\alpha}_i) = 0 \\
\frac{\partial L}{\partial \xi} &= \sum_{i=1}^{n} \frac{C}{n} - \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \theta_i = 0 \\
\frac{\partial L}{\partial \zeta} &= \sum_{i=1}^{n} \frac{C}{n} - \sum_{i=1}^{n} \tilde{\alpha}_i - \sum_{i=1}^{n} \tilde{\theta}_i = 0
\end{align*}
\]

Substituting Eq. (6) into Eq. (5), the Lagrange function can be rewritten as follows:

\[ L = -\frac{1}{2} \sum_{i,j=1}^{n} (\tilde{\alpha}_i - \alpha_i)(\tilde{\alpha}_j - \alpha_j)K(x_i, x_j) + \sum_{i=1}^{n} (\tilde{\alpha}_i - \alpha_i) y_i, \]

According to Karush-Kuhn-Tucker conditions, the optimization problem of the objective function expressed by Eq. (7) can be achieved via the optimal solution of the corresponding dual problem. Then, the decision function of the v-SVR can be expressed by

\[ y_i = f(x_i) = \sum_{i=1}^{n} (\tilde{\alpha}_i + \alpha_i) K(x_i, x_j) + b. \]
3. **Harris hawks optimization algorithm**

Harris hawks optimization algorithm (HHOA), which was introduced by Heidari, et al. [21], is designed based on simulation of the behaviors of Harris Hawks during the process of search and catching the rabbit in natural space. The HHOA process of optimization is explained by three phases in order to find the optimal solution for any given problem. Exploration, transition from exploration to exploitation, and exploitation are those phases. Where it has been used HHO and DA optimization methods to analyze the bearing capacity of footing over two-layer foundation soils by Moayedi et al [33]. In another study, the effect of the HHO optimization algorithm was studied to improve the accuracy of the traditional multilayer perception technique in order to analyze the stability of the soil slopes more accurately [34]. In addition, Mehta et al. [35] showed that the HHO algorithm can be regarded as an effective optimization algorithm to solve the problems of optimum load dispatch.

3.1 **Exploration phase**

The exploration phase mimics the process where a Harris hawk is no able to duly trace the prey. The hawks take a break when this happens to track and locate new preys. The candidate solutions in the HHOA are the hawks and the best solution so far is the prey at every step. The hawks then perch randomly in a different location and wait for a prey using two operators that are selected based on a probability $q$ [36, 37].

Mathematically, this process is modeled as

$$
x^{(t+1)} = \begin{cases} 
  x'_{\text{rand}} - r_1 x'_{\text{rand}} - 2r_2 x' & q \geq 0.5 \\
  (x'_{\text{prey}} - x'_m) - r_3 (L_b + r_4 (U_b - L_b)) & q < 0.5,
\end{cases}
$$

(9)
where $x^{(t+1)}$ is the position vector of hawks in the next iteration, $x^t_{\text{prey}}$ represents the position of intended rabbit, and $x^t_{\text{rand}}$ is the position of a hawk which is chosen randomly from current team. $r_1$, $r_2$, $r_3$ and $r_4$ are random numbers. $L_b$ and $U_b$ are the upper and lower bounds of search space. $x^t_m$ is the average position of the current population of hawks which is calculated by the following equation

$$x^t_m = \frac{1}{nh} \sum_{i=1}^{nh} x^t_i,$$

where $x^t_i$ is the position of each hawk in team and $nh$ indicates the total number of team members.

### 3.2 Transition phase

According to the energy level of the prey (escape energy), $E$, the HHO algorithm goes from the exploration phase to the exploitation phase. The energy reduction of the prey is defined as

$$E = 2E_0 (1 - \frac{t}{t_{\text{max}}}),$$

where $t_{\text{max}}$ indicates the maximum number of iterations and $E_0$ is the initial energy which is randomly changing inside $(-1,1)$ at each iteration. This value is used to indicate either the physically flagging of the prey for $-1 \leq E_0 < 0$ or its strengthening $0 \leq E_0 < 1$. Furthermore, in the case $|E| \geq 1$ then HHOA will explore the search space otherwise, HHOA will change its status to the exploitation phase [38].
3.3 Exploitation phase

During the exploitation phase, the \(|E|\) is considered to choose the type of besiege to catch the prey. Accordingly, a soft one is taken when \(|E| \geq 0.5\), and the hard one is taken when \(|E| < 0.5\) [39-41]. This process is stimulated by the following two strategies: Soft besiege and hard besiege.

In soft besiege strategy, the \(r \geq 0.5\) and \(|E| \geq 0.5\) (where \(r\) represents the prey’s ability to escape). This means that the prey still has enough escape energy, so the Harris hawks select the best solution from the population to update the solution. This can be formulated using the following equation:

\[
\Delta x^{(t+1)} = \Delta x^t - E \left| J x^t_{pury} - x^t \right|,
\]

where \(\Delta x^t = x^t_{pury} - x^t\), \(J = 2(1 - r_j)\), which is standing for jump severity of the prey in the stage of escaping, and \(r_j\) is a random number in the range \([0,1]\).

On the other hand, in hard besiege strategy, the \(r \geq 0.5\) and \(|E| < 0.5\), which means that the prey is tired and does not have sufficient energy to escape. The updated position of the Harris’ hawk is defined as

\[
x^{(t+1)} = x^t_{pury} - E \left| \Delta x^t \right|,
\]

In case of \(r < 0.5\) and \(|E| \geq 0.5\), which is called soft besiege with progressive rapid dives, the Harris’ hawk progressively selects the best possible dive to catch the prey competitively [42, 43]. Then, the new position of the hawk is mathematically modeled as
\[
Y = x_{pury}' - E \left| J x_{pury}' - x' \right|
\]  
(14)

The Harris’ hawk can dive by

\[
Z = Y + S \times \text{Levy}(D),
\]  
(15)

where \( D \) is the dimension of problem, \( S \) is a random vector by size \( 1 \times D \), and \( \text{Levy} \) is the levy flight function, which is calculated as

\[
\text{Levy}(D) = 0.01 \times \frac{\mu}{\delta} \left[ \frac{\Gamma(1 + \beta) \times \sin \left( \frac{\pi \beta}{2} \right)}{\Gamma(1 + \beta) / 2 \times \beta \times 2^{1 + \beta / 2}} \right]^{1/\beta},
\]  
(16)

where \( \mu \) and \( \delta \) are random values from \((0,1)\) and \( \beta \) is a constant ant its value is 1.5 [42]. In this phase, the position of the Harris’ hawk is updated as

\[
x^{(i+1)} = \begin{cases} 
Y & \text{if } \text{Fitness}(Y) < \text{Fitness}(x') \\
Z & \text{if } \text{Fitness}(Z) < \text{Fitness}(x') 
\end{cases}
\]  
(17)

where \( \text{Fitness}(\cdot) \) is the fitness function.

4. The proposed algorithm

In SVR there are several parameters that are needed to be fixing. These parameters are called hyperparameters, such as penalized parameter, \( C \), \( \epsilon \)-insensitive loss function, \( \epsilon \), and the kernel parameter. The SVR performance is very sensitive to the selection of these hyperparameters and there is no mathematical based procedure for deriving the exact desired values [44]. As a result, the selection of those hyperparameters is a crucial part of the research on SVR.
In the literature, there several attempts with different procedures to improve the SVR performance by appropriated choosing of these hyperparameters [9-11, 44-46]. Nature-inspired algorithms are among these different procedures that were employed to select the hyperparameters of SVR [11-20, 47]. However, in all these existence procedures regarding the selection of hyperparameters, there is no attempt to perform feature selection simultaneously. On the other hand, there is no attempt to tune the hyperparameters of v-SVR.

Our contribution of this work is to optimize the hyperparameter (v) of the v-support vector regression with embedding the feature selection simultaneously by employing the Harris hawks algorithm. In our proposed procedure, the type of kernel function is Gaussian kernel with parameter $\sigma > 0$. The illustrative of the solution representation is depicted in Figure 1. The flowchart of our proposed framework is presented in Figure 2.

![Figure 1: Representation of the proposed solution](image-url)
Figure 2: The flowchart of our proposed framework.

Each hawk in the family members has a position contains three quantitative values which are representing the $v$, and $p$ binary values which are representing the features.
The relevant feature will take the value of 1 and 0 otherwise. In other words, each hawk has 1 + \( p \) positions. The steps of our proposed algorithm are presented as follows.

**Step 1:** The number of hawks is \( nh = 10 \) and the maximum number of iterations is \( t_{\text{max}} = 100 \).

**Step 2:** The first positions represent the hypermeter is randomly generated from uniform distribution as \( \nu \sim U(0,1) \). The rest positions which are representing the feature are generated as \( U(0,1) \).

**Step 3:** The fitness function is defined as

\[
\text{fitness} = \min \left[ \frac{1}{nh} \sum_{i=1}^{nh} (y_{i, \text{test}} - \hat{y}_{i, \text{test}})^2 \right],
\]

where the fitness is calculated for the testing dataset.

**Step 4:** The positions the hawks are updated using Eqs. (9), (12), (13), and Eq. (17). To deal with feature selection, the binary HHOA (BHHOA) is utilized. Here, each hawk is represented by the \( p \)-bit binary string. To update the position, the transfer function is usually used to force hawk to be in a binary space. In order to build this binary vector, a transfer function can be used, in which the new solution is constrained to only binary values as [43].

\[
x^{t+1} = \begin{cases} 
1 & \text{if } T(\Delta x^{t+1}) > \text{rand} \\
0 & \text{otherwise},
\end{cases}
\]

where \( \text{rand} \in [0,1] \) is a random number, \( T(x) = (1/1 + \exp(-x)) \) is the sigmoid transfer function.
Step 5: Steps 3 and 4 are repeated until a $t_{\text{max}}$ is reached.

5. Experimental results

To test the predicting performance of our proposed algorithm, HHOA, comprehensive comparative experiments with grid search approach (GS) and cross-validation approach with ten folds (CV) are utilized. Three different sets of chemical datasets were used in this research: antidiabetic activity of dipeptidyl peptidase-IV inhibitors (Dataset 1) [48], anticancer potency of imidazo[4,5-b]pyridine derivatives (Dataset 2) [49], diverse series of antifungal agents (Dataset 3) [50]. All these datasets include thousands of descriptors as features. Table 1 presents a summary of these datasets. Each dataset was divided into two groups: a training dataset consisting of 70% of the total samples, and a test dataset consisting of 30%. This splitting is repeated 20 times. Two evaluation criteria were used:

Mean-squared error of the training dataset ($MSE_{\text{train}} = \frac{1}{n_{\text{train}}} \sum_{i=1}^{n_{\text{train}}} (y_{i,\text{train}} - \hat{y}_{i,\text{train}})^2 / n_{\text{train}}$) and mean-squared error of the testing dataset ($MSE_{\text{test}} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} (y_{i,\text{test}} - \hat{y}_{i,\text{test}})^2 / n_{\text{test}}$).

Table 1: Description of the datasets used.

| Dataset | #Samples | #features |
|---------|----------|-----------|
| Dataset 1 | 134 | 1048 |
| Dataset 2 | 65 | 2540 |
| Dataset 3 | 212 | 1048 |
Table 2 shows the preference for HHOA over two other methods, which are the CV and the GS. This preference based on the averaged mean square error, for instance in Dataset 1 HHOA has \( \text{MSE}_{\text{train}} \) lowest from CV and GS.

| Datasets  | GS  | CV  | HHOA |
|----------|-----|-----|------|
|          | MSE\(_{\text{train}}\) | MSE\(_{\text{train}}\) | MSE\(_{\text{train}}\) |
| Dataset 1 | 1.5411 | 1.3377 | 1.06E-07 |
| Dataset 2 | 1.0838 | 0.8296 | 0.306926 |
| Dataset 3 | 1.8951 | 1.6492 | 1.19E-07 |

Once again, based on the test set results in Table 3, the proposed method, HHOA, yields significantly better predictive ability as compared to CV and GS. The predictive performance achieved by the HHOA in Dataset 1, for example, was 0.0519, which was better than 1.5946 and 1.8472 obtained by CV and GS, respectively. Among the methods used, it is clearly seen that the GS is the worst predictive method.

| Datasets  | GS  | CV  | HHOA |
|----------|-----|-----|------|
|          | MSE\(_{\text{test}}\) | MSE\(_{\text{test}}\) | MSE\(_{\text{test}}\) |
| Dataset 1 | 1.8472 | 1.5946 | 1.01E-07 |
| Dataset 2 | 1.2068 | 1.0557 | 0.160816 |
| Dataset 3 | 1.9844 | 1.8069 | 1.18E-07 |

To further highlight the computational efficiency, Table 4 shows the CPU time of the proposed algorithm, HHOA, GS, and CV. As can be seen, in terms of computational efficiency, the HHOA has less time than CV and GS. The p-values (*) from Wilcoxon’s rank sum test (nonparametric statistical test) with 5% significance level are adopted. The statistical test is needed to indicate that the HHOA provides a significant improvement.
compared to the other methods. It can be seen that there is a statistical difference between HHOA and all the others for all datasets. This is not surprising because the GA and CV are computationally consuming time.

Table 4: The averaged computational time in seconds.

| Datasets | GS      | CV      | HHOA |
|----------|---------|---------|------|
| Dataset 1| 812.28 *| 757.13 *| 6.11 |
| Dataset 2| 773.57 *| 697.22 *| 10.87|
| Dataset 3| 884.72 *| 815.95 *| 5.56 |

In order to verify the feasibility and effectiveness of the proposed HHOA in optimizing the v-SVR hyperparameter and feature selection, the predictive performance of the HHOA is compared with other widely algorithms applied to solve this problem. These algorithms are: Particle swarm optimization (PSO), and whale optimization algorithm (WOA). The parameters of these algorithms are setting as basic. For the population size and the iteration numbers, they are assumed as same as in HHOA. The average $\text{MSE}_{\text{test}}$ and the computational time of the comparative algorithms are depicted in Table 5 and Figure 3, respectively.

From Table 5, it is clear that, HHOA performs better than the other algorithms in all the datasets which clearly demonstrates the predictive performance strength of the HHOA. WOA is in second place and PSO in the last place. In general, it is evident from Table 5 that HHOA has shown competitive performance in comparison to WOA and it has shown superior performance in comparison to PSO.
Table 5: Comparative experimental results (on average) for algorithms used based on testing dataset.

| Algorithms | Dataset 1  | Dataset 2  | Dataset 3  |
|------------|------------|------------|------------|
| PSO        | 0.0711     | 0.0326     | 0.3351     |
| WOA        | 0.0541     | 0.0246     | 0.2607     |
| HHOA       | 1.96E-07   | 0.306926   | 1.91E-07   |
6. Conclusion

Optimizing hyperparameter of v-SVR and performing feature selection are playing a crucial role in developing a successful study for any prediction problem. In this paper, HHOA was proposed to optimize the hyperparameter of the v-support vector regression with embedding the feature selection simultaneously. The experimental results and statistical analysis on four datasets have demonstrated that the performance of our proposed algorithm compared with the other methods and algorithms leads to a better performance in terms of prediction, number of selected features, and running time. Thus, the HHOA is more suitable than the others for describing the data relationship between input features and target variable. This highly effective predicting framework can be applied to other real applications.

References

[1] C.-C. Chuang and Z.-J. Lee, "Hybrid robust support vector machines for regression with outliers," Applied Soft Computing, vol. 11, pp. 64-72, 2011.
[2] Y.-F. Ye, Y.-H. Shao, N.-Y. Deng, C.-N. Li, and X.-Y. Hua, "Robust Lp -norm least squares support vector regression with feature selection," Applied Mathematics and Computation, vol. 305, pp. 32-52, 2017.
[3] Y.-P. Zhao and J.-G. Sun, "Robust truncated support vector regression," Expert Systems with Applications, vol. 37, pp. 5126-5133, 2010.

[4] S. Xu, B. Lu, M. Baldea, T. F. Edgar, and M. Nixon, "An improved variable selection method for support vector regression in NIR spectral modeling," Journal of Process Control, vol. 67, pp. 83-93, 2018.

[5] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, "New support vector algorithms," Neural computation, vol. 12, pp. 1207-1245, 2000.

[6] H. Kaneko and K. Funatsu, "Fast optimization of hyperparameters for support vector regression models with highly predictive ability," Chemometrics and Intelligent Laboratory Systems, vol. 142, pp. 64-69, 2015.

[7] N. A. Al-Thanoon, O. S. Qasim, and Z. Y. Algamal, "Tuning parameter estimation in SCAD-support vector machine using firefly algorithm with application in gene selection and cancer classification," Comput Biol Med, vol. 103, pp. 262-268, Dec 1 2018.

[8] N. A. Al-Thanoon, O. S. Qasim, and Z. Y. Algamal, "A new hybrid firefly algorithm and particle swarm optimization for tuning parameter estimation in penalized support vector machine with application in chemometrics," Chemometrics and Intelligent Laboratory Systems, vol. 184, pp. 142-152, 2019.

[9] J.-S. Chou and A.-D. Pham, "Nature-inspired metaheuristic optimization in least squares support vector regression for obtaining bridge scour information," Information Sciences, vol. 399, pp. 64-80, 2017.

[10] R. Laref, E. Losson, A. Sava, and M. Siadat, "On the optimization of the support vector machine regression hyperparameters setting for gas sensors array applications," Chemometrics and Intelligent Laboratory Systems, vol. 184, pp. 22-27, 2019.

[11] S. Li, H. Fang, and X. Liu, "Parameter optimization of support vector regression based on sine cosine algorithm," Expert Systems with Applications, vol. 91, pp. 63-77, 2018.

[12] C.-H. Wu, G.-H. Tzeng, and R.-H. Lin, "A Novel hybrid genetic algorithm for kernel function and parameter optimization in support vector regression," Expert Systems with Applications, vol. 36, pp. 4725-4735, 2009.

[13] 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," Applied Soft Computing, vol. 13, pp. 947-958, 2013.

[14] M. Nait Amar and N. Zeraibi, "Application of hybrid support vector regression artificial bee colony for prediction of MMP in CO2-EOR process," Petroleum, 2018.

[15] C.-F. Huang, "A hybrid stock selection model using genetic algorithms and support vector regression," Applied Soft Computing, vol. 12, pp. 807-818, 2012.

[16] C.-T. Cheng, W.-C. Wang, D.-M. Xu, and K. W. Chau, "Optimizing Hydropower Reservoir Operation Using Hybrid Genetic Algorithm and Chaos," Water Resources Management, vol. 22, pp. 895-909, 2007.

[17] W.-C. Hong, Y. Dong, L.-Y. Chen, and S.-Y. Wei, "SVR with hybrid chaotic genetic algorithms for tourism demand forecasting," Applied Soft Computing, vol. 11, pp. 1881-1890, 2011.
[18] J. Cheng, J. Qian, and Y.-n. Guo, "Adaptive chaotic cultural algorithm for hyperparameters selection of support vector regression," in *International Conference on Intelligent Computing*, 2009, pp. 286-293.

[19] B. Üstün, W. Melssen, M. Oudenhuijzen, and L. Buydens, "Determination of optimal support vector regression parameters by genetic algorithms and simplex optimization," *Analytica Chimica Acta*, vol. 544, pp. 292-305, 2005.

[20] J. Zhang, C.-H. Zheng, Y. Xia, B. Wang, and P. Chen, "Optimization enhanced genetic algorithm-support vector regression for the prediction of compound retention indices in gas chromatography," *Neurocomputing*, vol. 240, pp. 183-190, 2017.

[21] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849-872, 2019.

[22] G. H. Fu, D. S. Cao, Q. S. Xu, H. D. Li, and Y. Z. Liang, "Combination of kernel PCA and linear support vector machine for modeling a nonlinear relationship between bioactivity and molecular descriptors," *Journal of chemometrics*, vol. 25, pp. 92-99, 2011.

[23] B. Malik, K. Chaitanya, and M. Benaisa, "Support vector regression with digital band pass filtering for the quantitative analysis of near-infrared spectra," *Journal of Chemometrics*, vol. 28, pp. 116-122, 2014.

[24] D. Kong, Y. Chen, N. Li, and S. Tan, "Tool wear monitoring based on kernel principal component analysis and v-support vector regression," *The International Journal of Advanced Manufacturing Technology*, vol. 89, pp. 175-190, 2016.

[25] P. Samui, V. R. Mandla, A. Krishna, and T. Teja, "Prediction of rainfall using support vector machine and relevance vector machine," *Earth Science India*, vol. 4, 2011.

[26] C.-C. Chang and C.-J. Lin, "Training v-support vector regression: theory and algorithms," *Neural computation*, vol. 14, pp. 1959-1977, 2002.

[27] P. Y. Hao, "Pair- $\$ v \$ -SVR: A Novel and Efficient Pairing nu-Support Vector Regression Algorithm," *IEEE Trans Neural Netw Learn Syst*, vol. 28, pp. 2503-2515, Nov 2017.

[28] N. Li, Y. Chen, D. Kong, and S. Tan, "Force-based tool condition monitoring for turning process using v-support vector regression," *The International Journal of Advanced Manufacturing Technology*, vol. 91, pp. 351-361, 2016.

[29] Y. Liu and G. Pender, "A flood inundation modelling using v-support vector machine regression model," *Engineering Applications of Artificial Intelligence*, vol. 46, pp. 223-231, 2015.

[30] X. Teng, H. Dong, and X. Zhou, "Adaptive feature selection using v-shaped binary particle swarm optimization," *PLoS One*, vol. 12, p. e0173907, 2017.

[31] J. Wang, R. Hou, C. Wang, and L. Shen, "Improved v-support vector regression model based on variable selection and brain storm optimization for stock price forecasting," *Applied Soft Computing*, vol. 49, pp. 164-178, 2016.

[32] Y. Zhang and Y. Xie, "Forecasting of short-term freeway volume with v-support vector machines," *Transportation Research Record*, vol. 2024, pp. 92-99, 2007.

[33] H. Moayedi, H. Nguyen, and A. S. A. Rashid, "Comparison of dragonfly algorithm and Harris hawks optimization evolutionary data mining techniques for
the assessment of bearing capacity of footings over two-layer foundation soils," *Engineering with Computers*, pp. 1-11, 2019.

[34] H. Moayedi, A. Osouli, H. Nguyen, and A. S. A. Rashid, "A novel Harris hawks’ optimization and k-fold cross-validation predicting slope stability," *Engineering with Computers*, pp. 1-11, 2019.

[35] M. S. Mehta, M. B. Singh, and M. Gagandeep, "Harris Hawks optimization for solving optimum load dispatch problem in power system," *Int J Eng Res Technol*, vol. 8, pp. 962-968, 2019.

[36] E. H. Houssein, M. E. Hosney, D. Oliva, W. M. Mohamed, and M. Hassaballah, "A novel hybrid Harris hawks optimization and support vector machines for drug design and discovery," *Computers & Chemical Engineering*, vol. 133, p. 106656, 2020.

[37] E. H. Houssein, M. R. Saad, K. Hussain, W. Zhu, H. Shaban, and M. Hassaballah, "Optimal Sink Node Placement in Large Scale Wireless Sensor Networks Based on Harris’ Hawk Optimization Algorithm," *IEEE Access*, vol. 8, pp. 19381-19397, 2020.

[38] T. A. Shehabeldeen, M. A. Elaziz, A. H. Elsheikh, and J. Zhou, "Modeling of friction stir welding process using adaptive neuro-fuzzy inference system integrated with harris hawks optimizer," *Journal of Materials Research and Technology*, vol. 8, pp. 5882-5892, 2019.

[39] C. Qu, W. He, X. Peng, and X. Peng, "Harris Hawks optimization with information exchange," *Applied Mathematical Modelling*, vol. 84, pp. 52-75, 2020.

[40] A. S. Menesy, H. M. Sultan, A. Selim, M. G. Ashmawy, and S. Kamel, "Developing and Applying Chaotic Harris Hawks Optimization Technique for Extracting Parameters of Several Proton Exchange Membrane Fuel Cell Stacks," *IEEE Access*, vol. 8, pp. 1146-1159, 2020.

[41] N. A. Golilarz, A. Addeh, H. Gao, L. Ali, A. M. Roshandeh, H. Mudassir Munir, *et al.*, "A New Automatic Method for Control Chart Patterns Recognition Based on ConvNet and Harris Hawks Meta Heuristic Optimization Algorithm," *IEEE Access*, vol. 7, pp. 149398-149405, 2019.

[42] Q. Fan, Z. Chen, and Z. Xia, "A novel quasi-reflected Harris hawks optimization algorithm for global optimization problems," *Soft Computing*, 2020.

[43] Too, Abdullah, and S. Mohd, "A New Quadratic Binary Harris Hawk Optimization for Feature Selection," *Electronics*, vol. 8, p. 1130, 2019.

[44] P. Tsirikoglou, S. Abraham, F. Contino, C. Lacor, and G. Ghorbaniasl, "A hyperparameters selection technique for support vector regression models," *Applied Soft Computing*, vol. 61, pp. 139-148, 2017.

[45] V. Cherkassky and Y. Ma, "Practical selection of SVM parameters and noise estimation for SVM regression," *Neural networks*, vol. 17, pp. 113-126, 2004.

[46] K. Ito and R. Nakano, "Optimizing support vector regression hyperparameters based on cross-validation," in *Proceedings of the International Joint Conference on Neural Networks*, 2003, pp. 2077-2082.

[47] G. Cao and L. Wu, "Support vector regression with fruit fly optimization algorithm for seasonal electricity consumption forecasting," *Energy*, vol. 115, pp. 734-745, 2016.
[48] A. Al-Fakih, Z. Algamal, M. Lee, M. Aziz, and H. Ali, "A QSAR model for predicting antidiabetic activity of dipeptidyl peptidase-IV inhibitors by enhanced binary gravitational search algorithm," SAR and QSAR in Environmental Research, vol. 30, pp. 403-416, 2019.

[49] Z. Y. Algamal, M. H. Lee, A. M. Al-Fakih, and M. Aziz, "High-dimensional QSAR prediction of anticancer potency of imidazo [4, 5-b] pyridine derivatives using adjusted adaptive LASSO," Journal of Chemometrics, vol. 29, pp. 547-556, 2015.

[50] A. Al-Fakih, Z. Algamal, M. Lee, M. Aziz, and H. Ali, "QSAR classification model for diverse series of antifungal agents based on improved binary differential search algorithm," SAR and QSAR in Environmental Research, vol. 30, pp. 131-143, 2019.