

Research Article

Probability Distribution and Deviation Information Fusion Driven Support Vector Regression Model and Its Application

Changhao Fan and Xuefeng Yan

Key Laboratory of Advanced Control and Optimization for Chemical Processes of Ministry of Education, East China University of Science and Technology, MeiLong Road No. 130, Shanghai 200237, China

Correspondence should be addressed to Xuefeng Yan; xfyan@ecust.edu.cn

Received 29 June 2017; Revised 25 August 2017; Accepted 30 August 2017; Published 12 October 2017

Academic Editor: Xinkai Chen

Copyright © 2017 Changhao Fan and Xuefeng Yan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In modeling, only information from the deviation between the output of the support vector regression (SVR) model and the training sample is considered, whereas the other prior information of the training sample, such as probability distribution information, is ignored. Probabilistic distribution information describes the overall distribution of sample data in a training sample that contains different degrees of noise and potential outliers, as well as helping develop a high-accuracy model. To mine and use the probability distribution information of a training sample, a new support vector regression model that incorporates probability distribution information weight SVR (PDISVR) is proposed. In the PDISVR model, the probability distribution of each sample is considered as the weight and is then introduced into the error coefficient and slack variables of SVR. Thus, the deviation and probability distribution information of the training sample are both used in the PDISVR model to eliminate the influence of noise and outliers in the training sample and to improve predictive performance. Furthermore, examples with different degrees of noise were employed to demonstrate the performance of PDISVR, which was then compared with those of three SVR-based methods. The results showed that PDISVR performs better than the three other methods.

1. Introduction

Since its proposal by Vapnik, the support vector machine (SVM) has been used in many areas, including both pattern recognition and regression estimation [1, 2]. The original SVM is utilized to provide a pair of parameters as a solution to a quadratic program problem. SVM has some advantages, such as low standard deviation and easy generation, as well as some disadvantages, such as the redundancy of the regression function and the low efficiency of support vector selection. To address these disadvantages, various improvements to the support vector algorithm and its kernel function have been proposed. Suykens proposed least-square support vector regression (LS-SVR) for a regression modeling problem [3, 4]. By transferring inequality constraints to equality constraints, LS-SVR simplifies the solution to quadratic program problems [5]. In the field of regression, Smola proposed the linear programming support vector regression (LP-SVR) model [6, 7]. LP-SVR has numerous strengths, such as the

using of more general kernel functions and fast learning ability. LP-SVR can control the accuracy and sparseness of the original SVR by using the linear kernel combination as a solution approach. In addition, a new kernel function, multi-kernel function (MK), has been introduced into the standard SVM model. MK provides lower fault and requires a shorter training period than the original kernel function. Multiple-kernel SVR (MKSVR) is very popular in some systems. Yeh et al. [8] developed MKSVR for stock market forecasts. Lin and Jhuo [9] discovered a method to generate MKSVR parameters for integration into a system that converts the pixels of a checkpoint into the brightness value. Zhong and Carr [10] used the MKSVR model to estimate pure and impure carbon dioxide-oil matrix metalloproteinases in a CO₂ enhanced oil recovery process.

The SVR model also has been improved by prior knowledge [11, 12]. There are numerous types of prior knowledge, including the average value and monotonicity of the sample data. To appropriately use prior knowledge, three types of

methods are utilized in SVR [13]. Our team previously worked on the monotonous a priori knowledge of sample data. Our monotonous a priori knowledge of the sample data is described by first-order difference inequality constraints of kernel expansion and additive kernels [14]. The constraints are directly added to kernel formulation to acquire a convex optimization problem. For additive kernels, SVMs are conducted through the addition of dissociate kernels for every input dimension. These operations confer higher accuracy to the SVR model in support vector (SV) selection.

Inevitably, even small noise can debase the accuracy of the model. Furthermore, in some situations, part of the noisy information may be ten to even dozens of times larger than the normal data. These outliers introduce bias and inaccuracies to SVR. Nevertheless, the probability distribution of the sample data is a good indicator of noise. From the perspective of the probability distribution of sample data, normal data and data that contain the least amount of noise have the highest probability in the sample data. By contrast, data that contain large amount of noise have relatively small probability. Thus, outliers in the sample data will have the smallest probability. Therefore, the probability distribution is the prior knowledge that helps weaken the influence from noise and outliers in the sample data. We consider this information to modify our SVR model.

This article is structured as follows: Section 2 introduces standard SVR algorithms. Section 3 describes the proposed algorithm that integrates probability distribution information into the SVR framework. Section 4 provides some experimental results that were obtained from comparing the proposed algorithm with other algorithms. Finally, Section 5 presents some conclusions about the proposed algorithm.

2. Review of SVR

To better describe the proposed algorithm, the mathematical clarification of the basic concepts of SVR and the usage of deviation information should be provided.

2.1. Support Vector Regression (SVR). SVR is originally used to solve linear regression problems. For given training samples $\mathbf{X} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R}, i = 1, 2, \dots, n\}$, fitting aims to find the dependency between the independent variable \mathbf{x}_i and the dependent variable y_i . Specifically, it aims to identify an optimal function and minimize prospective risk $R(\boldsymbol{\omega}) = \int L(y, f(\mathbf{x}, \boldsymbol{\omega})) dP(\mathbf{x}, y)$, where $\{f(\mathbf{x}, \boldsymbol{\omega})\}$ is predictive function set, $\boldsymbol{\omega} \in \boldsymbol{\Omega}$ is the generalized parameters of the function, $L(y, f(\mathbf{x}, \boldsymbol{\omega}))$ is the loss function, and

$$f(\mathbf{x}, \boldsymbol{\omega}) = \boldsymbol{\omega} \cdot \mathbf{x} + b \quad (1)$$

is the fitting function [15]. Thus, the solution of the optimal linear function for SVR is expressed as the following constraint optimization problem:

$$\begin{aligned} \min \quad & R(\boldsymbol{\omega}, \xi_i, \xi_i^*) = \frac{1}{2} \langle \boldsymbol{\omega} \cdot \boldsymbol{\omega} \rangle + C \sum_{i=1}^n (\xi_i + \xi_i^*) \\ & \boldsymbol{\omega} \cdot \mathbf{x} - y_i + b \leq \varepsilon + \xi_i^* \end{aligned}$$

$$\begin{aligned} \text{s.t.} \quad & y_i - \boldsymbol{\omega} \cdot \mathbf{x} - b \leq \varepsilon + \xi_i \\ & \xi_i, \xi_i^* \geq 0, \quad i = 1, 2, \dots, n, \end{aligned} \quad (2)$$

where the penalty coefficient C that determines the accuracy of the function fitting and the degree of the error greater than ε is given in advance. Parameter ε is used to control the size of the fitting error, the size of the support vector, and the size of the generalization capability. Taking into account the accuracy of the fitting error, the introduction of slack variables ξ_i, ξ_i^* becomes necessary. Figure 1 in reference [10] illustrates this linear fitting problem.

However, the previous solution is only for a linear regression problem. Nonlinear regression necessitates the kernel function in the SVR model [16]. The kernel function can be expressed as follows:

$$\mathbf{K}(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}) \cdot \phi(\mathbf{z}) \rangle, \quad (3)$$

where ϕ is the mapping from a low-dimensional space to a high-dimensional space. The independent variable \mathbf{x}_i becomes a vector that should be mapped to a feature space so that a nonlinear problem could be changed into a linear problem. After introducing the kernel function, the new fitting function becomes

$$f(\mathbf{x}) = \sum_{i=1}^n \omega_i k(\mathbf{x}, \mathbf{x}_i) + b = \mathbf{K}(\mathbf{x}, \mathbf{X}^T) \boldsymbol{\omega} + b, \quad (4)$$

where the symbol \mathbf{X}^T indicates the transpose of the matrix \mathbf{X} .

The changing of the fitting function leads to the following constraint optimization problem:

$$\begin{aligned} \min \quad & R(\boldsymbol{\omega}, \xi_i, \xi_i^*) = \frac{1}{2} \langle \boldsymbol{\omega} \cdot \boldsymbol{\omega} \rangle + C \sum_{i=1}^n (\xi_i + \xi_i^*) \\ & \mathbf{K}(\mathbf{x}, \mathbf{X}^T) \boldsymbol{\omega} - y_i + b \leq \varepsilon + \xi_i^* \\ \text{s.t.} \quad & y_i - \mathbf{K}(\mathbf{x}, \mathbf{X}^T) \boldsymbol{\omega} - b \leq \varepsilon + \xi_i \\ & \xi_i, \xi_i^* \geq 0, \quad i = 1, 2, \dots, n. \end{aligned} \quad (5)$$

In this constraint optimization problem, the length of $\boldsymbol{\omega}$ and \mathbf{x} is n . The notion $\mathbf{K}(\cdot, \cdot)$ is the kernel function that fulfills Mercer's requirements.

The standard SVR is a compromise between structural risk minimization and empirical risk minimization. In particular, for the support vector regression learning algorithm, the structural risk term is $(1/2)\langle \boldsymbol{\omega} \cdot \boldsymbol{\omega} \rangle$ and the empirical risk term is $\sum_{i=1}^n (\xi_i + \xi_i^*)$. However, calculating the structural risk term $(1/2)\langle \boldsymbol{\omega} \cdot \boldsymbol{\omega} \rangle$ requires enormous time and resources [17]. Researchers found counting the minimization of the 1-norm of the parameter $\boldsymbol{\omega}$ will reduce the time and resources spent

on calculation. Then, the optimization formula turns into the following form:

$$\begin{aligned} \min \quad & R(\boldsymbol{\omega}, \xi_i, \xi_i^*) = \|\boldsymbol{\omega}\|_1 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \\ & \mathbf{K}(\mathbf{x}, \mathbf{X}^T) \boldsymbol{\omega} - y_i + b \leq \varepsilon + \xi_i^* \\ \text{s.t.} \quad & y_i - \mathbf{K}(\mathbf{x}, \mathbf{X}^T) \boldsymbol{\omega} - b \leq \varepsilon + \xi_i \\ & \xi_i, \xi_i^* \geq 0, \quad i = 1, 2, \dots, n. \end{aligned} \quad (6)$$

Although the time and resource spent on modeling are reduced, there is no considerable difference in the final accuracy.

2.2. Support Vector Regression with Deviation Information as a Consideration. Traditional SVR does not possess a special method for addressing noise in sample data. An efficient way to weaken noise is to adjust parameters in the SVR model. These parameters are called hyperparameters in SVRs. Hyperparameters exert a considerable impact on algorithm performance. The general way to test the performance of hyperparameters is via the deviation between the model output and the sample data [18]. The obtained deviation is then compared with other deviations to select the minimum deviation as the final result. The parameters that correspond to the minimum deviation are the best parameters in the optimization process. Usually, this process is conducted using an intelligent optimization algorithm, such as particle swarm optimization (PSO) [19] and genetic algorithm (GA) [20]. The deviation is set as the fitness function in an intelligent optimization algorithm. In this section, we refer to this method as deviation-minimized SVR (DM-SVR).

In most of the circumstances, the deviation between the model output and sample data is represented by the correlation coefficient r or the mean square error (MSE). Given vector $\hat{\mathbf{y}}$ as the model output and vector \mathbf{y} as the sample output, the correlation coefficient r can be expressed as

$$r = \frac{\sum_{i=1}^N \hat{y}_i y_i - \sum_{i=1}^N \hat{y}_i \sum_{i=1}^N y_i / N}{\sqrt{\left(\sum_{i=1}^N \hat{y}_i^2 - \left(\sum_{i=1}^N \hat{y}_i\right)^2 / N\right) \left(\sum_{i=1}^N y_i^2 - \left(\sum_{i=1}^N y_i\right)^2 / N\right)}}. \quad (7)$$

The formula for mean square error (MSE) is as follows:

$$\text{MSE} = \frac{\sum_{i=1}^N (|\hat{y}_i - y_i|)^2}{N}. \quad (8)$$

In short, if the value of MSE is close to zero and the value of r is close to one, that group of parameters will produce the best performance.

3. Probability Distribution Information Weighted Support Vector Regression

Although DM-SVR can reduce influence from the noise, it also has some weaknesses. The main disadvantage of this method is the time it spends on training. There are many parameters that need to be optimized in SVR. If there are extra parameters to optimize, these works would make the train process inefficient. To solve the uncertainty of error parameter ε , we introduce the probability distribution information (PDI) into SVR and designate it as PDISVR.

3.1. Probability Distribution of the Output. The probability distribution information is the same as the probability distribution function and describes the likelihood that the output value of a continuous random variable is near a certain point. Integrating the probability density function is the proper way to calculate the probability value of the random variable in that certain region. From the sample data, we could set the frequency of output to appear as different values. Then, we set frequency as $N(\mathbf{y})$, where \mathbf{y} is the output value vector. Let $P(\mathbf{y})$ be the probability of the sample's output. Therefore, the relationship between $P(\mathbf{y})$ and $N(\mathbf{y})$ can be expressed as

$$P(\mathbf{y}) = \frac{N(\mathbf{y})}{\sum_{\mathbf{y} \in \mathbf{Y}} N(\mathbf{y})}, \quad (9)$$

where \mathbf{Y} is the range of \mathbf{y} . Then, we can easily obtain the probability distribution function. The next step is the identification of the probability of every point.

3.2. Optimization Formula with Probability Distribution Information Weight. Once we have obtained the probability distribution of output, it should be integrated into the basic SVR model. In the basic SVR model, the error parameter ε indicates the accuracy of model fitting by providing an area that does not have any loss for the objective function. However, due to the influence of noise, some sample data contain excessive noise information. If the same parameters ε are adapted, the performance of the model is reduced. To prevent this situation, SVR should be adjusted in accordance with noise information. We propose illustrating noise information through the probability distribution of the output. Samples in the regions with low probability distributions have a relatively large proportion of noise. For this reason, in modeling, the region with higher probability should have a smaller error parameter than the lower probability region. Thus, the probability distribution function increases the accuracy of the SVR model in the area with the high probability of output.

Define the ε -insensitive loss function as

$$|\mathbf{y} - f(\mathbf{x})|_\varepsilon = \begin{cases} 0, & \text{if } |\mathbf{y} - f(\mathbf{x})| \leq \varepsilon \\ |\mathbf{y} - f(\mathbf{x})| - \varepsilon, & \text{otherwise,} \end{cases} \quad (10)$$

where $f(\mathbf{x})$ is a regression estimation function constructed by learning the sample and \mathbf{y} is the target output value that corresponds to \mathbf{x} . By defining the ε -insensitive loss function, the SVR model specifies the error requirement of the estimated

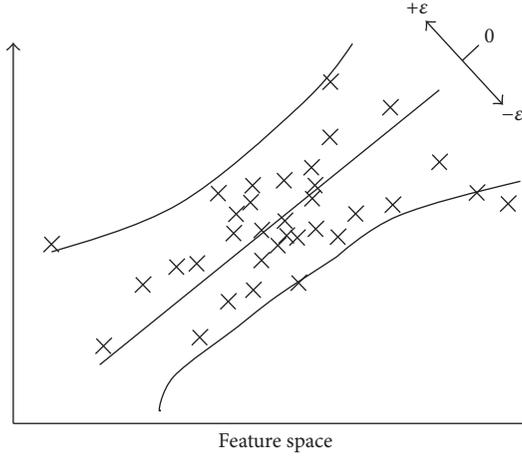


FIGURE 1: Linear problem illustration of a PDISVR model.

function on the sample data. This requirement is the same for every sample point. To avoid this situation, the artificially set error parameter ε is divided by the probability distribution vector $P(y)$. Figure 1 illustrates the change from a constant ε to a vector $\boldsymbol{\varepsilon}$. The distance between two hyperplanes has been modified in some area where the density of the points becomes different. Furthermore, in the high-density area, the model has a smaller error parameter. By contrast, in the low-density area, the model has a large error parameter. The density of the output points is directly related to the probability of the sample's output \mathbf{y} . Therefore, the division of PDI would make the SVR model emphasize the area with a high density of points. This technique can improve overall accuracy despite sacrificing the accuracy of low-density areas. According to (9) and (10), the PDISVR can be expressed as

$$\begin{aligned}
 \min \quad & R(\boldsymbol{\omega}, \xi_i, \xi_i^*) = \|\boldsymbol{\omega}\|_1 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \\
 & \mathbf{K}(\mathbf{x}, \mathbf{X}^T) \boldsymbol{\omega} - y_i + b \leq \frac{\varepsilon}{P(y_i)} + \xi_i^* \\
 \text{s.t.} \quad & y_i - \mathbf{K}(\mathbf{x}, \mathbf{X}^T) \boldsymbol{\omega} - b \leq \frac{\varepsilon}{P(y_i)} + \xi_i \\
 & \xi_i, \xi_i^* \geq 0, \quad i = 1, 2, \dots, n.
 \end{aligned} \tag{11}$$

By comparing (11) with the standard form of SVR, we can see that the error parameter ε changes in accordance with y_i . Then, the PDISVR model will have low error tolerance for the high density of points.

To further improve the performance of the SVR model, we consider adding an extra fragment to the PDISVR framework. The PDISVR model only has a unique error parameter $\boldsymbol{\varepsilon}$. However, $\boldsymbol{\varepsilon}$ is too small to have an obvious impact on the accuracy of the model. Hence, we propose an additional method to introduce PDI; that is, we apply the same operation as that on error parameter ε on the slack variable ξ . Given the treated error parameter $\boldsymbol{\varepsilon}$ in the PDISVR model, we divided

TABLE 1: Typical parameters for PSO algorithms.

Parameter	Value
Maximum generation	200
Population size	20
Cognitive efficient ($u1$)	1.5
maximum velocity (V_{\max})	1
Social efficient ($u2$)	1.7
Initial inertia weights	0.9
Final inertia weights	0.4
minimum velocity (V_{\min})	10

the slack variable ξ by probability distribution information $P(y)$. Subsequently, the final PDISVR model is

$$\begin{aligned}
 \min \quad & R(\boldsymbol{\omega}, \xi_i, \xi_i^*) = \|\boldsymbol{\omega}\|_1 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \\
 & \mathbf{K}(\mathbf{x}, \mathbf{X}^T) \boldsymbol{\omega} - y_i + b \leq \frac{1}{P(y_i)} (\varepsilon + \xi_i^*) \\
 \text{s.t.} \quad & y_i - \mathbf{K}(\mathbf{x}, \mathbf{X}^T) \boldsymbol{\omega} - b \leq \frac{1}{P(y_i)} (\varepsilon + \xi_i) \\
 & \xi_i, \xi_i^* \geq 0, \quad i = 1, 2, \dots, n.
 \end{aligned} \tag{12}$$

3.3. Parameters Optimization Based on PSO. Normally, the performance of the different SVRs is intensively dependent on the parameters selection. A PSO based hyperparameters selecting method in [10] is used in this paper. After dataset was normalized, the control parameters of PSO including maximum velocity (V_{\max}), minimum velocity (V_{\min}), initial inertia weight (ω_{\min}), final inertia weight (ω_{\max}), cognitive coefficient (ϕ_1), social coefficient (ϕ_2), maximum generation, and population size should be initialized according to the experience of operators. In our experiment, we set the control parameters of PSO based on Table 1. In the following experiments, there are at most five hyperparameters of different SVRs that need to be optimized in PSO. These parameters include penalizing coefficient (C), radial basis function kernel parameter (γ), and polynomial degree (d), (ε) in ε -insensitivity function and mixing coefficient (m) in multikernel function. In our experiments, different comparative methods adapt different groups of parameters and, in order to search the global optimum reasonably, the parameter m is limited in $[0, 1]$, γ in $[10^{-2}, 10]$, d in $[1, 10]$, C in $[10, 1000]$, and ε in $[10^{-4}, 10^{-1}]$. During the process of searching best parameters by PSO, particles update their positions by changing velocity and converge finally at a global optimum within the searching space. In this study, the particles' positions are the combination of m, γ, d, C , and ε , which are denominated as \mathbf{P} . Then v -fold ($v = 5$) cross-validation resampling method is applied to validate the performance of searched best parameters until criteria are met. To evaluate the performance of training process, mean square error (MSE) is chosen as the fitness function, which is formulated as (8). Figure 2 shows the workflow to find the optimum values of each parameter in models.

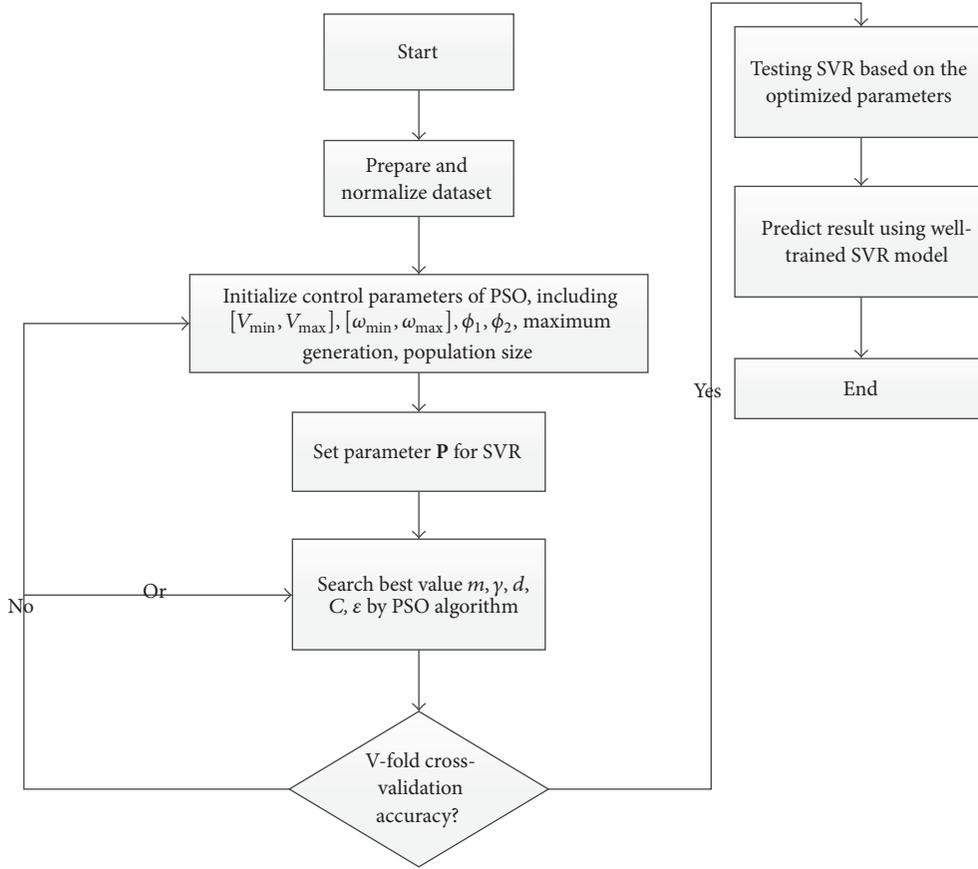


FIGURE 2: Workflow on hyperparameters' selection.

4. Experimental Results

To verify the effect of the probability distribution information on the standard SVR model, we employed three kinds of numerical experiments with real datasets. In these experiments, we considered three kernels including linear kernel, polynomial kernel, and Gaussian kernel as SVRs' kernel functions. All of the experiments were operated on MATLAB with Intel i5 CPU and 6 GB internal storage.

Experimental studies have mainly compared different SVR models, including basic SVR, MKSVR and heuristic weighted SVR [21]. The correlation coefficient r and mean square error (MSE) are used to evaluate generalization performance. The formulas of these two criteria are listed in Section 2.2.

4.1. Example 1. Example 1 tested the previous four methods with one-high-dimensional functions, as shown in [22]

$$\begin{aligned}
 f(\mathbf{x}) = & \sum_{i=1}^n \{ [\ln(x_i - 2)]^2 + [\ln(10 - x_i)]^2 \} \\
 & - \left(\prod_{i=1}^n x_i \right)^{0.2} + e_i(k), \tag{13} \\
 & 2.1 \leq x_i \leq 9.9 \quad (n = 3).
 \end{aligned}$$

In the above function, the symbol $e_i(k)$ indicates the random fluctuation variable between $-k$ and k . From the range of [2.1, 9.9] for x_i above, we generated 100 data at random. Through (13), we obtained the output of these 100 data. We evenly divided the 100 data into five parts, which comprised four training parts and one testing part. After the cross-validation method introduced in Section 3.3, optimal hyperparameters for different SVR algorithms are selected.

After obtaining the optimal hyperparameters, we then determined the influence of noise on different SVR algorithms. The range of the magnitude of the noise k , which was set as 0.1, 0.5, 1, 3, 6, and 10, was set in accordance with the output range. To obtain objective comparisons, 10 groups of noise were added to each training sample of the algorithm using the MATLAB toolbox, which completely generated 10 training datasets. Moreover, testing data were directly generated from the objective function equation (13). The results for the criterion of these ten experiments are recorded by their average and standard deviation values, as shown in Figures 3–5.

In these three figures, the average criterion values indicated that the general performance of the algorithm and the standard deviation are representative of the algorithm's stability. From Figures 3–5, we can see that the performance of the proposed PDISVR is less affected by noisy information than those of the other three SVR algorithms. In Figure 3, the result

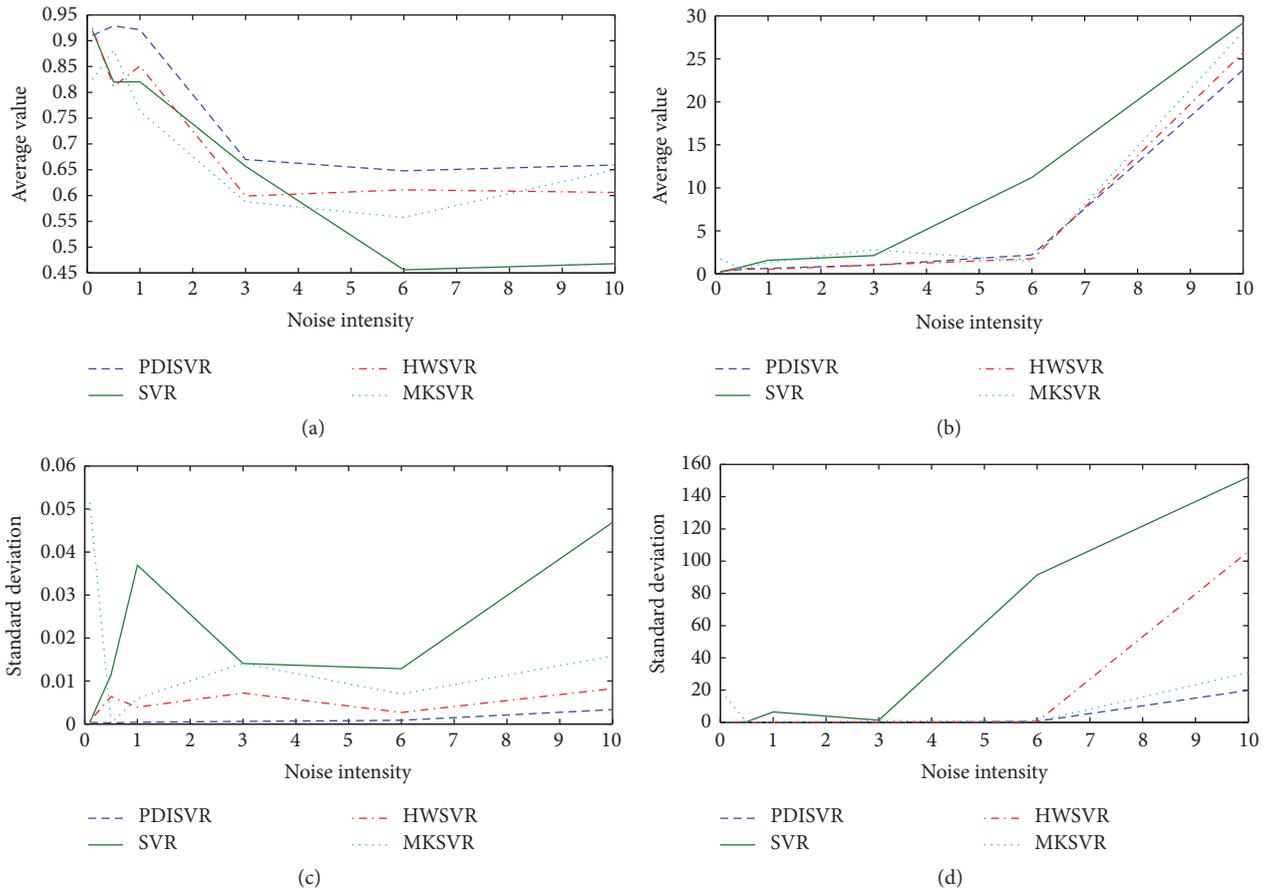


FIGURE 3: Four models' prediction results in linear kernel with different noise: (a) is the average value of correlation coefficient r , (b) is the average value of MSE, (c) is the standard deviation of correlation coefficient r , and (d) is the standard deviation of MSE.

line of PDISVR is more stable than other three methods. And it achieves best prediction performance when adding larger noise in samples among all models. Compared to Figure 3, the PDISVR's ability to predict is not always the best in Figures 4 and 5. That means the PDISVR with linear kernel is suitable for this dataset. Besides, in the area with large intensity of noise, the basic SVR and MKSVR poorly handled the effects of noise, although HW-LSSVR resisted some of the effects of noise, its performance slightly worsened with a high intensity of noisy samples. The average of the prediction accuracy and standard deviation of PDISVR were relatively better in fitting models with noise of 1, 3, 6, and 10. With noise of 0.1 and 0.5, although the differences among the average values were small, the PDISVR was more stable than other algorithms in some certain circumstances.

4.2. Example 2. The effects from rough error cannot be ignored in real production processes. To better simulate real production conditions and reveal the robustness of the proposed PDISVR when the training samples involved outliers, the rough error term should be added to the function in the previous model. A total of 80 data with a noise intensity of 1 were haphazardly generated by (13) as a fundamental training sample. Test samples containing 20 data were also generated

by (13). Then, the dependent variables of the 17th and 48th data in the fundamental training sample were attached $4.5 \times e$ and $3 \times e$, respectively, as two trivial outliers. The dependent variable of the 50th datum in the fundamental training sample was attached $10 \times e$ as one strong outlier. Thus, the new training sample that contained one strong outlier, that is, the 50th datum, and two trivial outliers, that is, the 17th and 48th data, was constructed.

To better compare the predictive performance of the different SVR algorithms, the same four algorithms were trained ten times in samples with three outliers. The average values and standard deviation values of r and MSE represented the performance of these algorithms.

As indicated in Tables 2–4, the PDISVR algorithm performed better in the testing experiments than the other algorithms. The unweighted SVR and MKSVR were influenced by noise and produced biased estimates in predicting results, whereas PDISVR dramatically reduced this secondary action. Given the misjudgments on the outliers in this complicated system, the HWSVR algorithm could not obtain satisfactory results even when it adapted weighted error parameters.

In order to illustrate the quality of PID-SVR's weight element, we compared the weighting results in Table 2. The weight values of the PDISVR algorithm in training samples

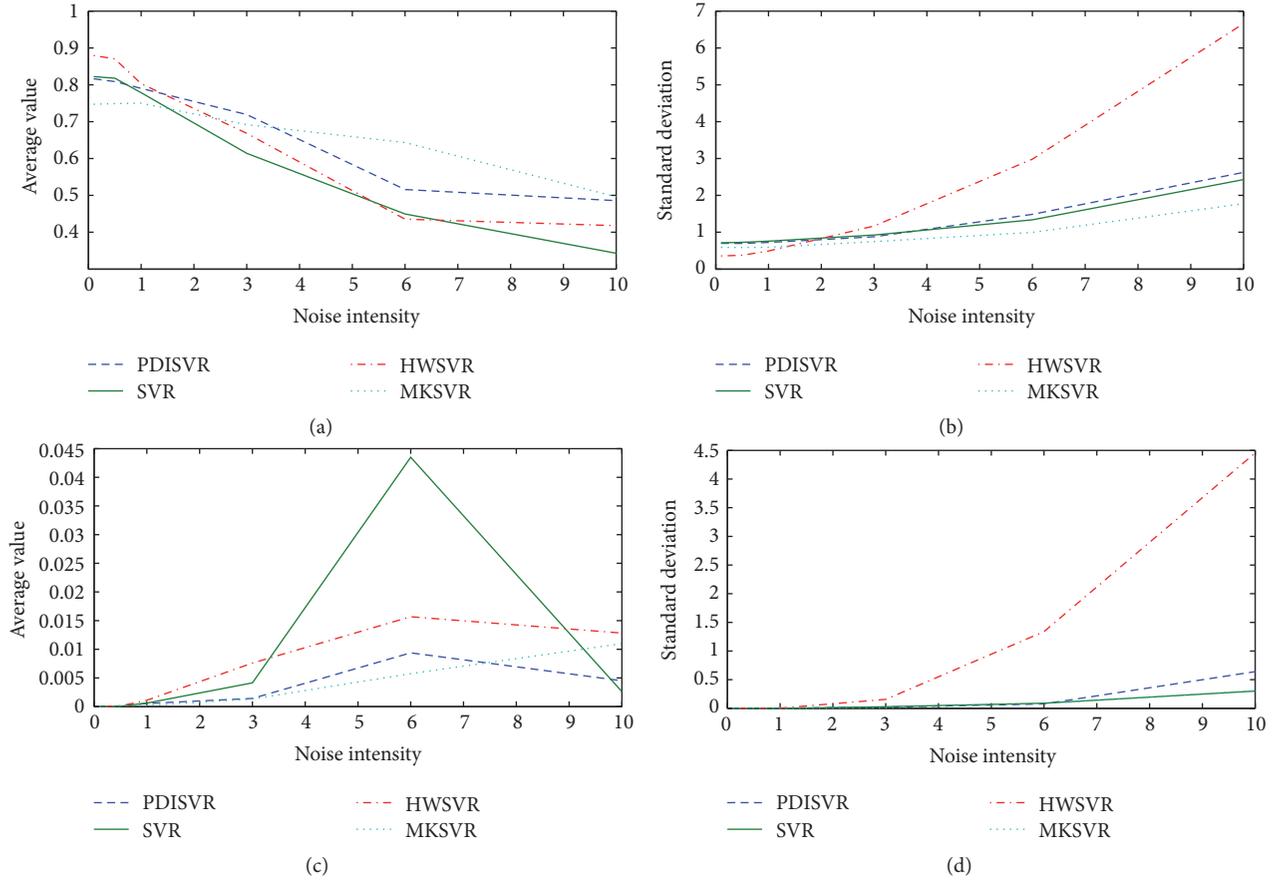


FIGURE 4: Four models' prediction results in radial basis function kernel with different noise: (a) is the average value of correlation coefficient r , (b) is the average value of MSE, (c) is the standard deviation of correlation coefficient r , and (d) is the standard deviation of MSE.

TABLE 2: Testing results of SVR algorithms with rough error (linear kernel).

Criterion	PDISVR	SVR	HWSVR	MKSVR
r				
Average value	0.89278	0.86844	0.84402	0.85828
Standard deviation	0.000258	0.002867	0.002707	0.001998
MSE				
Average value	0.30614	0.52311	0.58221	0.45612
Standard deviation	0.003953	0.111161	0.045897	0.028321

TABLE 3: Testing results of SVR algorithms with rough error (radial basis function kernel).

Criterion	PDISVR	SVR	HWSVR	MKSVR
r				
Average value	0.78424	0.78126	0.75976	0.64346
Standard deviation	0.000297	0.000169	0.000774	0.000244
MSE				
Average value	0.70686	0.73748	0.62362	0.84296
Standard deviation	0.00316	0.00150	0.00506	0.00136

are listed in Figure 6. The weight values of the HWSVR algorithm are listed in Figure 7. As shown in Figure 6, the weights of two trivial outliers were 0 and 0.10531 for the 17th and 48th data, respectively, and the weight of one strong outlier was 0.00036, which indicated that the PDISVR precisely

detected the outliers. As shown in Figure 7, the HWSVR did not perform as well as PDISVR. One strong outlier had a weight of 0.0751 and two trivial outliers had weights of 0.3143 and 0.2729, which were unsuitable for modeling given that smaller weights, such as that of the 23rd datum (0.0126),

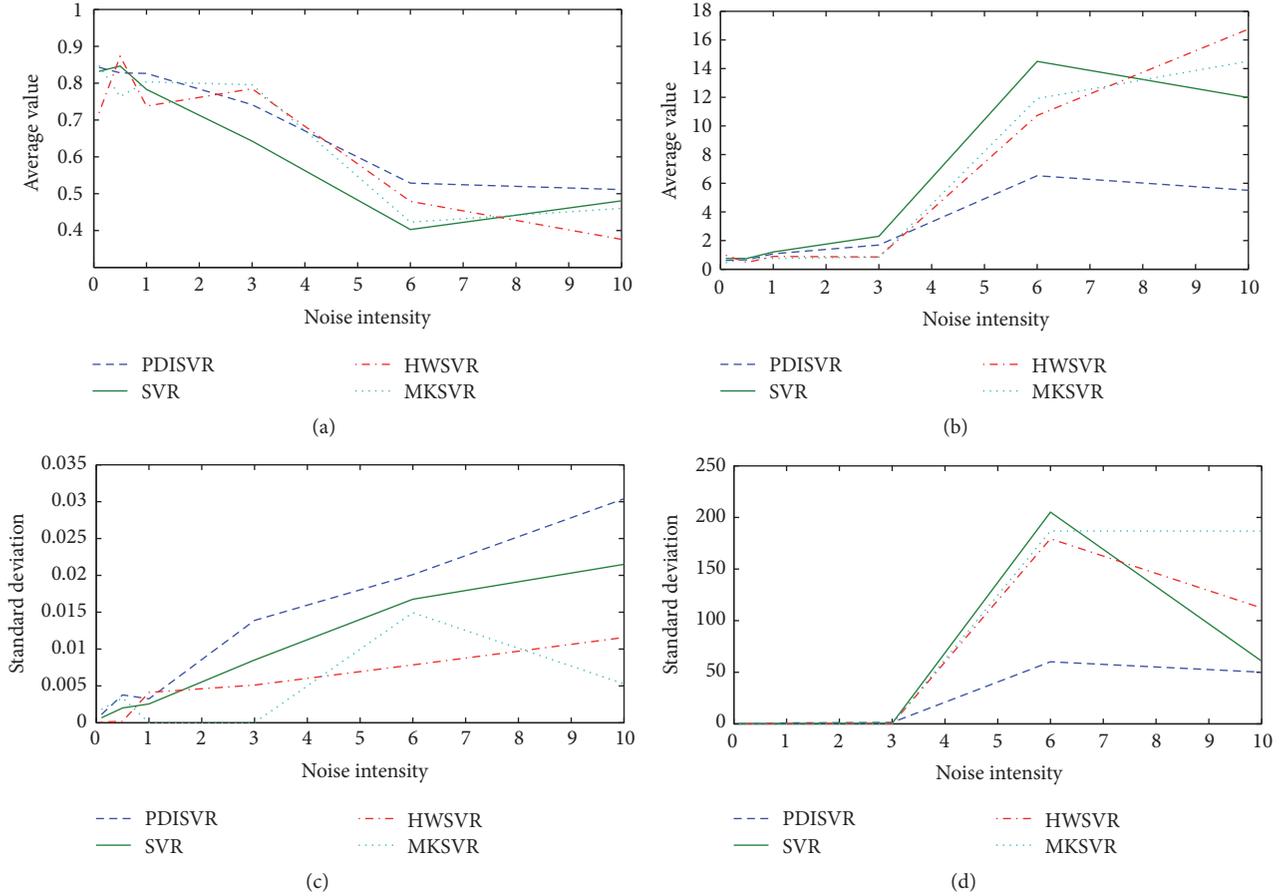


FIGURE 5: Four models' prediction results in polynomial kernel with different noise: (a) is the average value of correlation coefficient r , (b) is the average value of MSE, (c) is the standard deviation of correlation coefficient r , and (d) is the standard deviation of MSE.

TABLE 4: Testing results of SVR algorithms with rough error (polynomial kernel).

Criterion	PDISVR	SVR	HWSVR	MKSVR
r				
Average value	0.83278	0.79399	0.75874	0.8137
Standard deviation	0.000377	0.004657	0.009121	0.003568
MSE				
Average value	0.4473	0.727	1.48208	0.62433
Standard deviation	0.003168	0.076291	1.462867	0.028654

could affect outlier detection. Thus, the influence from two trivial outliers on the predictability of the PDISVR algorithm was reduced and the influence from one strong outlier was eliminated. By contrast, the effect of outliers remained in the HWSVR algorithm.

4.3. Example 3. To test our regression model in a more realistic way, we imported six more realistic datasets from the UCI Machine Learning Repository [23–25], Department of Food Science, University of Copenhagen database [26], and some real chemical industrial process [14]. See Table 5 for more detailed information. In these datasets, four out of five data were used for training and one-fifth of the data was used for

testing. The hyperparameters used in this example are also obtained by the process introduced in Section 3.3.

As shown in Tables 6–8, the proposed PDISVR obtained the best predictive ability in the majority of the criterion. For example, in the case of Auto-MPG, the proposed PDISVR was best achieved with both standards. Thus, the proposed PDISVR is appropriate for the Auto-MPG dataset. In the datasets for crude oil distillation and computer hardware, the proposed PDISVR only obtained the best correlation coefficient r and could not establish a suitable model at a point where the probability distribution was low, thus increasing the MSE. Therefore, the use of PDISVR requires validation

TABLE 5: Details of the experimental datasets.

Data	Number of instances	Number of attributes	Dataset information
Combustion side reaction	144	6	The combustion side reaction is the process by which the solvent and reactants are oxidized to carbon dioxide, carbon monoxide, and water
Auto-MPG dataset	398	8	Data concerning city-cycle fuel consumption
Computer hardware dataset	209	9	Relative CPU performance in terms of cycling time, memory size, and other parameters
Soil samples measured by Near-Infrared Spectroscopy (NIR)	108	1050	This data is about the soil organic matter (SOM) from northern Sweden (681210N, 181490E)
Energy efficiency	768	9	This study looked into assessing the heating load and cooling load requirements of buildings (that is, energy efficiency) as a function of building parameters
Crude oil distillation process	30	1	Data related to different components with different boiling points

TABLE 6: Comparative results of previous SVR models in real datasets (linear kernel).

	Criterion	PDISVR	SVR	HWSVR	MKSVR
Combustion side reaction	r	0.7141	0.6790	0.6797	0.6939
	MSE	0.0051	0.0050	0.0046	0.0044
Auto-MPG	r	0.9055	0.8578	0.8992	0.8554
	MSE	19.0023	25.8051	23.7999	20.4802
Computer hardware	r	0.9533	0.8750	0.8748	0.9012
	MSE	7987.2	8373.7	8381.5	6577.2
NIRSOIL	r	0.8033	0.7335	0.7712	0.7657
	MSE	1806.1	2230.3	2085.1	2385.8
Energy efficiency	r	1	1	1	0.9999
	MSE	0.0047	1.1092	0.0221	0.0818
Crude oil distillation process	r	0.9372	0.7684	0.9372	0.9927
	MSE	1016.2	972.4	14510	8162.8

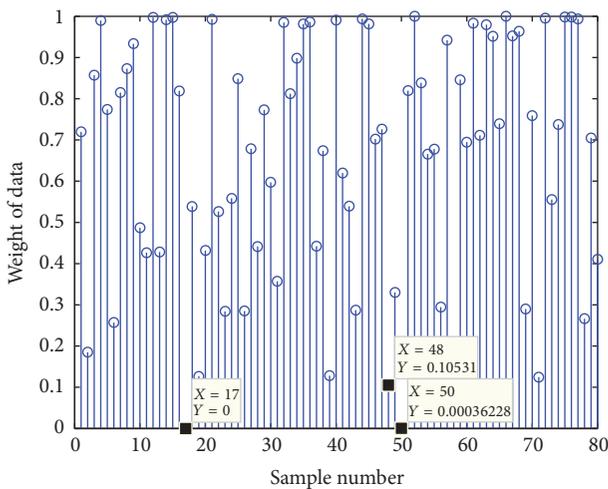


FIGURE 6: Weights of the training sample for PDISVR.

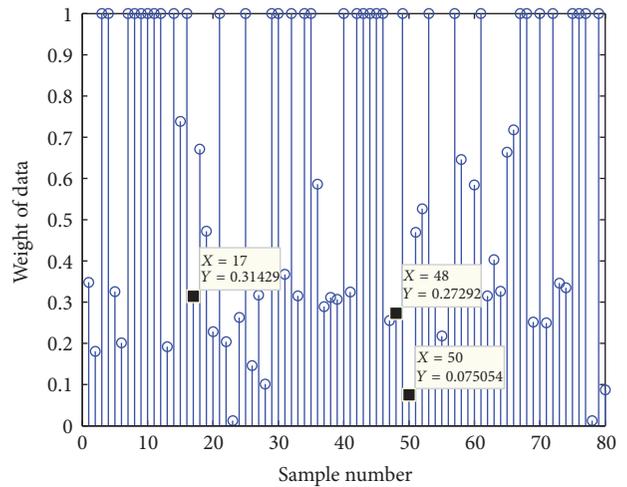


FIGURE 7: Weights of the training sample for HWSVR.

through additional research and dataset message. Moreover, in the sample of crude oil boiling point, there are far less quantity of data, and the PDISVR cannot differentiate the

noise according to its probability distribution. Thus, the proposed PDISVR is applied to improve the SVR in the case of large datasets.

TABLE 7: Comparative results of previous SVR models in real datasets (radial basis function kernel).

	Criterion	PDISVR	SVR	HWSVR	MKSVR
Combustion side reaction	r	0.6664	0.425	0.6031	0.5827
	MSE	0.0038	0.0095	0.6031	0.0578
Auto-MPG	r	0.5594	0.614	0.6261	0.6563
	MSE	20.4277	26.1598	169.7247	22.0482
Computer hardware	r	0.7815	0.8597	0.8647	0.8851
	MSE	27961	19986	15355	13160
NIRSOIL	r	0.7278	0.6707	0.5565	0.5666
	MSE	1113	1302.3	1601.1	3523.5
Energy efficiency	r	1	0.9796	0.9901	0.9997
	MSE	0.000269	4.1875	3.019	0.0474
Crude oil distillation process	r	0.9899	0.9739	0.9913	0.9573
	MSE	47.7425	78.7604	1119.8	317.6248

TABLE 8: Comparative results of previous SVR models in real datasets (polynomial kernel).

	Criterion	PDISVR	SVR	HWSVR	MKSVR
Combustion side reaction	r	0.5796	0.6853	0.8078	0.5518
	MSE	0.0081	0.0106	0.3546	0.1237
Auto-MPG	r	0.9124	0.9338	0.9341	0.8604
	MSE	1821.3	6793.3	6814.4	4855.9
Computer hardware	r	0.8337	0.8571	0.8519	0.8423
	MSE	12.5509	19.5298	19.8641	13.9781
NIRSOIL	r	0.6576	0.6342	0.6468	0.6429
	MSE	2751.3	3104.7	2918.9	3007.3
Energy efficiency	r	1	0.9964	0.9988	0.9867
	MSE	0.0221	0.4598	0.1676	4.1448
Crude oil distillation process	r	0.9571	0.8071	0.9807	0.9913
	MSE	321.8016	112.9353	54.3637	1105.4000

5. Conclusion

In traditional SVR modeling, the deviation between the model outputs and the real data is the only way to represent the influence of noise. Other information, such as possibility distribution information, is not emphasized. Therefore, we proposed a special method that uses the possibility distribution information to modify the basic error parameter and slack variables of SVR. Given that these parameters are weighted by the probability distribution of the model output points, they can be adjusted by SVR itself and no longer require optimization by any intelligent optimization algorithm. The proposed algorithm is superior to other SVR-based algorithms in dealing with noisy data and outliers in simulation and actual datasets.

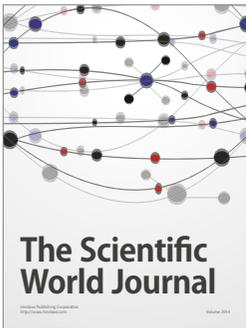
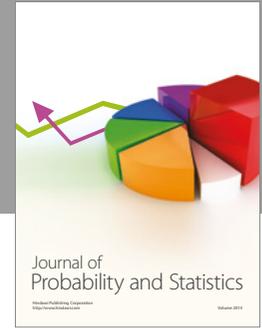
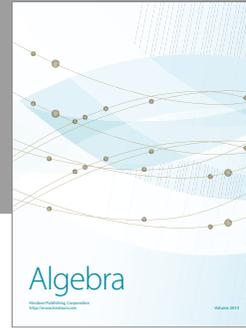
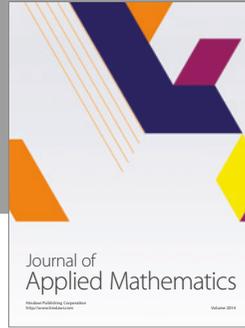
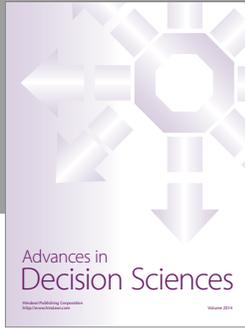
Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, 2000.
- [2] S. Yin and J. Yin, "Tuning kernel parameters for SVM based on expected square distance ratio," *Information Sciences*, vol. 370-371, pp. 92-102, 2016.
- [3] J. A. K. Suykens, J. Vandewalle, and B. de Moor, "Optimal control by least squares support vector machines," *Neural Networks*, vol. 14, no. 1, pp. 23-35, 2001.
- [4] J. A. K. Suykens, J. De Brabanter, L. Lukas, and J. Vandewalle, "Weighted least squares support vector machines: robustness and sparse approximation," *Neurocomputing*, vol. 48, pp. 85-105, 2002.
- [5] H.-S. Tang, S.-T. Xue, R. Chen, and T. Sato, "Online weighted LS-SVM for hysteretic structural system identification," *Engineering Structures*, vol. 28, no. 12, pp. 1728-1735, 2006.
- [6] A. Smola, B. Schoelkopf, and G. Raetsch, "Linear programs for automatic accuracy control in regression," in *Proceedings of 9th International Conference on Artificial Neural Networks (ICANN99)*, pp. 575-580, IEE, UK, September 1999.
- [7] A. Smola and B. Scholkopf, *Learning with Kernels*, MIT Press, Cambridge, UK, 2002.
- [8] C. Y. Yeh, C. W. Huang, and S. J. Lee, "A multiple-kernel support vector regression approach for stock market price forecasting," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2177-2186, 2011.
- [9] W.-J. Lin and S.-S. Jhuo, "A fast luminance inspector for backlight modules based on multiple kernel support vector

- regression,” *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 4, no. 8, pp. 1391–1401, 2014.
- [10] Z. Zhong and T. R. Carr, “Application of mixed kernels function (MKF) based support vector regression model (SVR) for CO₂ – Reservoir oil minimum miscibility pressure prediction,” *Fuel*, vol. 184, pp. 590–603, 2016.
- [11] G. Bloch, F. Lauer, G. Colin, and Y. Chamaillard, “Support vector regression from simulation data and few experimental samples,” *Information Sciences*, vol. 178, no. 20, pp. 3813–3827, 2008.
- [12] J. Zhou, B. Duan, J. Huang, and N. Li, “Incorporating prior knowledge and multi-kernel into linear programming support vector regression,” *Soft Computing*, vol. 19, no. 7, pp. 2047–2061, 2015.
- [13] F. Lauer and G. Bloch, “Incorporating prior knowledge in support vector machines for classification: A review,” *Neurocomputing*, vol. 71, no. 7-9, pp. 1578–1594, 2008.
- [14] C. Pan, Y. Dong, X. Yan, and W. Zhao, “Hybrid model for main and side reactions of p-xylene oxidation with factor influence based monotone additive SVR,” *Chemometrics and Intelligent Laboratory Systems*, vol. 136, pp. 36–46, 2014.
- [15] O. Naghash-Almasi and M. H. Khooban, “PI adaptive LS-SVR control scheme with disturbance rejection for a class of uncertain nonlinear systems,” *Engineering Applications of Artificial Intelligence*, vol. 52, pp. 135–144, 2016.
- [16] Y. Wang and L. Zhang, “A combined fault diagnosis method for power transformer in big data environment,” *Mathematical Problems in Engineering*, vol. 2017, Article ID 9670290, 6 pages, 2017.
- [17] H. Dai, B. Zhang, and W. Wang, “A multiwavelet support vector regression method for efficient reliability assessment,” *Reliability Engineering and System Safety*, vol. 136, pp. 132–139, 2015.
- [18] P.-Y. Hao, “Pairing support vector algorithm for data regression,” *Neurocomputing*, vol. 225, pp. 174–187, 2017.
- [19] W.-C. Hong, “Chaotic particle swarm optimization algorithm in a support vector regression electric load forecasting model,” *Energy Conversion and Management*, vol. 50, no. 1, pp. 105–117, 2009.
- [20] G. Wang, T. R. Carr, Y. Ju, and C. Li, “Identifying organic-rich Marcellus Shale lithofacies by support vector machine classifier in the Appalachian basin,” *Computers and Geosciences*, vol. 64, pp. 52–60, 2014.
- [21] W. Wen, Z. Hao, and X. Yang, “A heuristic weight-setting strategy and iteratively updating algorithm for weighted least-squares support vector regression,” *Neurocomputing*, vol. 71, no. 16-18, pp. 3096–3103, 2008.
- [22] H. Xiong, Z. Chen, H. Qiu, H. Hao, and H. Xu, “Adaptive SVR-HDMMR metamodeling technique for high dimensional problems,” *International Conference on Modelling, Identification and Control: AASRI Proceediapp*, vol. 3, pp. 95–100, 2012.
- [23] M. Lichman, Auto MPG Data Set. UCI Machine Learning Repository. <http://archive.ics.uci.edu/ml/datasets/Auto+MPG>, 2013.
- [24] M. Lichman, Computer Hardware Data Set. UCI Machine Learning Repository. <http://archive.ics.uci.edu/ml/datasets/Computer+Hardware>, 2013.
- [25] A. Tsanas and A. Xifara, “Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools,” *Energy and Buildings*, vol. 49, pp. 560–567, 2012.
- [26] R. Rinnan and Å. Rinnan, “Application of near infrared reflectance (NIR) and fluorescence spectroscopy to analysis of microbiological and chemical properties of arctic soil,” *Soil Biology and Biochemistry*, vol. 39, no. 7, pp. 1664–1673, 2007.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

