Global Nonlinear Kernel Prediction for Large Dataset with a Particle Swarm Optimized Interval Support Vector Regression

Yongsheng Ding, *Senior Member, IEEE*, Lijun Cheng, Witold Pedrycz, *Fellow, IEEE*, and Kuangrong Hao

***Abstract*—A new global nonlinear predictor with a particle swarm optimized interval support vector regression (PSO-ISVR) is proposed to address three issues (viz. kernel selection, model optimization, kernel method speed) encountered when applying support vector regression (SVR) in the presence of large datasets. The novel prediction model can reduce the SVR computing overhead by dividing input space and adaptively selecting the optimized kernel functions to obtain optimal SVR parameter by PSO. To quantify the quality of the predictor, its generalization performance and execution speed are investigated based on statistical learning theory. In addition, experiments using synthetic data as well as the stock volume weighted average price (VWAP) are reported to demonstrate the effectiveness of the developed models. The experimental results show that the proposed PSO-ISVR predictor can improve the computational efficiency and the overall prediction accuracy compared with the results produced by the SVR and other regression methods. The proposed PSO-ISVR provides an important tool for nonlinear regression analysis of big data.**

***Index Terms*—global nonlinear predictor, interval support vector regression, particle swarm optimization, kernel function, sliding adaptive model, large data**

I. Introduction

SUPPORT vector regression (SVR) model is constructed based on statistical learning theory [1], which uses a kernel function to map the data from some input space to a high-dimensional feature space where the problem becomes amenable for handling by linear regression [2]. Owing to its robustness to noise and its generalization abilities, it has been widely employed in various areas such as adaptive flight identification [3], ore grade estimation [4] , and stock market price forecasting [5-7].

Many researchers have pointed out that three crucial problems existing in SVR urgently need to be addressed: (1) How to choose or construct an appropriate kernel to complete forecasting problems [8, 9]; (2) How to optimize parameters of SVR to improve the quality of prediction [10, 11]; (3) How to construct a fast algorithm to operate in presence of large datasets [12, 13]. With unsuitable kernel functions or hyper-parameter settings, SVR may lead to poor prediction results. In fact, a kernel function forms a certain nonlinear transformation function. Due to data uncertainty in practical regression problems, it is difficult to determine which kernel function is the best one for a specific problem without any prior knowledge [14]. If the adjustable kernel parameters in SVR are not properly selected, it will result in the undesirable phenomena of over-fitting or under-fitting [1]. Furthermore, SVR is typically confronted with a heavy computing overhead due to processing large Gram matrices being associated with the kernels [13]. This computing burden becomes an essential barrier when dealing with massive data, such as those encountered in protein structure prediction and time series prediction [15].

During the past several years, various methods have been proposed to address these three problems encountered in SVR applications. (1) For kernel function selection, many researchers integrate multiple-kernels learning [6, 9] or construct some new kernel functions based on some prior knowledge available for the specific problems [16, 17]. With the increasing number of kernels being available, it is not obvious which kernel function is a suitable one for a specific problem at hand. Therefore it becomes necessary to construct a kernel function library so that the kernel functions can be selected depending upon the specificity of the application problem. (2) Since the performance of SVR depends on its kernel and parameters, an effective way is to use a re-sampling

This work was supported in part by the Key Project of the National Nature Science Foundation of China (No. 61134009), the National Nature Science Foundation of China (No. 61473077, 61473078), Cooperative research funds of the National Natural Science Funds Overseas and Hong Kong and Macao scholars (No. 61428302), Program for Changjiang Scholars from the Ministry of Education, Specialized Research Fund for Shanghai Leading Talents, Project of the Shanghai Committee of Science and Technology (Nos. 13JC1407500), and Innovation Program of Shanghai Municipal Education Commission (No. 14ZZ067).

Yongsheng Ding, Lijun Cheng, and Kuangrong Hao are all at the Engineering Research Center of Digitized Textile & Apparel Technology, Ministry of Education, together with the College of Information Science and Technology, Donghua University, Shanghai 201620 China (e-mail: ysding@dhu.edu.cn (corresponding author), lijcheng@iupui.edu, krhao@dhu.edu.cn). Lijun Cheng is also currently at Centers for Computational Biology and Bioinformatics together with Department of Medical and Molecular Genetics, School of Medicine, Indiana University, IN 46202, USA.

Witold Pedrycz is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada, Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah, 21589, Saudi Arabia, and Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland (e-mail: pedrycz@ee.ualberta.ca).

technique such as the cross-validation or leave-one-out [18], or to enumerate all possible combinations of parameters to choose proper values by running methods such as a grid search algorithm [19]. However, for huge datasets and multi-parameter applications, re-sampling and enumerating approaches could become computationally expensive. It is also difficult to set an entire suite of dynamic parameters. Recently some advanced algorithms have been proposed to adjust SVR parameters, such as immune clonal selection algorithms (AIA) [15], genetic algorithms (GA) [20], and particle swarm optimization (PSO) algorithms [21, 22]. These approaches are based on an optimal control design strategy being used to guarantee the SVR performance. The search rules implemented by the PSO algorithm are simpler than those encountered in the GA and the AIA, and PSO algorithm is easy to implement and exhibits quick convergence. In light of these advantages, recently the PSO algorithm has attracted a lot of attention as an effective optimization tool [11, 21, 23]. (3) Given the kernel matrix problem present in SVR, some researchers employ data clustering and data pruning methods to reduce computing time, such as K-Means clustering [12], Fuzzy C-Means [13], and clustering kernel row vectors [24]. These methods are feasible and greatly mitigate the influence of noise and outliers presented in large datasets. However, if we use the weighted clustering average data to conduct a further application, the whole data feature cannot ensure to be unchanged and the important information is intact after the clustering.

Considering the above three aspects altogether, a novel sliding adaptive model with particle swarm optimized interval support vector regression (PSO-ISVR) is proposed to address the difficulties of applying the SVR to large datasets. Firstly, the sample space is divided into a number of subspaces, a sliding controller is used to select the optimal kernel function from the kernel function library to fit the nature of the dataset present in each subspace. Secondly, the PSO algorithm optimizes the parameters of the SVR and produces the optimal PSO-ISVR subspace fitting hyper-plane. Finally, the optimal subspace fitting hyper-planes are combined to construct a global nonlinear predictor by using Lagrange interpolation surfaces’ join algorithm. In order to demonstrate the effectiveness of the approach, the PSO-ISVR model is applied to nonlinear numeric functions and the prediction problem of the volume weighted average price (VWAP) on Shanghai stock market exchange index for companies traded in China.

The main contributions of this paper are as follows: A global nonlinear predictor with the PSO-ISVR is proposed for large datasets. The design strategy is highly relevant and exhibits a certain level of originality: the PSO-ISVR simultaneously selects kernel type and optimizes kernel parameters adaptively, which reduces the PSO-SVR complexity, greatly reducing the computing overhead and storage size requirements of SVR. All of these improve the practical usage of the resulting method, especially in the presence of large datasets.

This paper is organized as follows. The basic SVR method and the PSO-SVR model are illustrated in Section 2. The global nonlinear predictor with the PSO-ISVR is presented in Section 3. The performance of the global nonlinear predictor is examined experimentally by using synthetic data and dealing with the VWAP prediction problem in Section 4. Finally, concluding remarks are provided in Section 5.

II. Support Vector Regression with Particle Swarm Optimization

A. Support Vector Regression

Statistical Learning Theory (SLT) provides a very effective framework for SVR. SVR model attempts to minimize the generalization error bound so as to achieve generalized performance. This generalization error bound is the combination of the training error and a regularization term that controls the complexity of the hypothesis space.

Given a dataset T with l examples, T={(x_i, y_i), $x_i \in R^n$, $y_i \in R$, i=1,2,…,l}, where x_i is an input variable, y_i is the target value. The data x are mapped onto high-dimensional feature space F by a nonlinear transformation function $\phi(x)$:

$$\phi: \begin{matrix} R^n \to F \\ x \to X = \phi(x) \end{matrix} \qquad (1)$$

A linear regression function $f(\phi(x))$ is constructed to predict the target y in F as follows:

$$\hat{y} = f(\phi(x)) = w^T \phi(x) + b \qquad (2)$$

where $w \in F$ is a weight vector and b is a coefficient constant. The expression (2) is referred as a regression hyper-plane in F. The structural risk minimization principle provides a theoretical basis to find a suitable approximation hyper-plane from the random samples by using the loss function specified [1] as follows

$$\psi(x) = \frac{1}{2} \| w \|^2 + L \cdot R_{emp}[f] \qquad (3)$$

where L is a regularization constant, $R_{emp}[f]$ is the empirical risk, also referred to as a cost function, which expresses the prediction error loss. Vapnik's ε-insensitive function is often used to describe a tolerable error within the extent of the ε-tube, as shown in (4),

$$R_{emp}[f] = | y - f(x) |_\varepsilon = \begin{cases} 0 & ,| y - f(x) | \le \varepsilon \\ | y - f(x) | - \varepsilon & ,| y - f(x) | > \varepsilon \end{cases} \qquad (4)$$

SVR minimizes the generalized error bound so as to achieve generalized performance. The input data need not lie on or inside the ε-insensitive band strictly. The positive slack variables ξ_i and ξ_i^* are introduced to cope with the infeasible constraint errors. The loss function $R_{emp}[f]$ is defined by the slack variables $| y_i - f(\phi(x_i)) |_\varepsilon = \xi_i + \xi_i^*$. Only the points outside the ε-tube are penalized. The regression problem can be expressed as the following convex optimization problem:

$$\begin{aligned} min \quad & \tfrac{1}{2} \| w \|^2 + C \sum_{i=1}^{l} (\xi_i + \xi_i^*) \\ s.t. \quad & y_i - < w, \phi(x_i) > - b \le \varepsilon + \xi_i \\ & < w, \phi(x_i) > + b - y_i \le \varepsilon + \xi_i^* \\ & \xi_i, \xi_i^* \ge 0 \end{aligned} \qquad (5)$$

where C >0 is the error penalty parameter, which determines a tradeoff between the training error and the complexity of the model; $\varepsilon > 0$ is an accuracy coefficient, which controls the width of the ε-insensitive zone.

The primal problem shown in (5) is solved by handling its dual problem under the Karush-Kuhn-Tucker (KKT) conditions and the Wolfe dual [1] as:

TABLE I
A COLLECTION OF KERNEL FUNCTIONS

Kernel type	Kernel function
Linear kernel	$k(x,y)=(x\cdot y)$
Polynomial kernel	$k(x,y)=(x\cdot y+d)^q$, q>0 integer , d∈R
Radial basis kernel	$k(x,y)=\exp(-\lvert x-y\rvert^2/\delta),\delta>0$
Sigmoid kernel	$k(x,y)=\tanh(u<x,y>+p),u>0,p<0$

TABLE II
POSITION AND VELOCITY VECTORS IN THE PSO-SVR SPACE

Kernel function	Dimension L	Particle position P_i	Velocity V_i
Linear	2	(C,ε)	(V_C, V_ε)
Polynomial	4	(C,ε,q,d)	$(V_C, V_\varepsilon, V_q, V_d)$
Radial basis	3	(C,ε,δ)	$(V_C, V_\varepsilon, V_\delta)$
Sigmoid	4	(C,ε, u, p)	$(V_C, V_\varepsilon, V_u, V_p)$

Note: C - penalty parameter, ε is the insensitivity loss parameter, and kernel parameters depending upon the form of the kernel.

$$\min_{\alpha,\alpha^*}\left\{\frac{1}{2}[\alpha,\alpha^*]\begin{bmatrix}Q & -Q\\ -Q & Q\end{bmatrix}[\alpha,\alpha^*]^T + \begin{bmatrix}\varepsilon I+y & \varepsilon I-y\end{bmatrix}[\alpha,\alpha^*]^T\right\} \tag{6}$$

$$s.t. \quad [I,-I][\alpha,\alpha^*]^T=0$$

where $\alpha=[\alpha_1,\alpha_2,...,\alpha_l]$ and $\alpha^*=[\alpha_1^*,\alpha_2^*,...,\alpha_l^*]$ are defined as vectors of Lagrange multipliers, $\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)=0$, $\alpha_i,\alpha_i^*\in[0,C]$, $Q_{ij}=\phi^T(x_i)\phi(x_j),(i,j=1,2,...,l)$ represents the inner product, which will be substituted by a kernel function value $k(x_i, x_j)$, where I=[1,...,1] is the unit vector, y =[y_1, y_2, ..., y_l].

Once the SVR has been trained, some α_i and α_i^* become equal to zero. Only those examples with nonzero values for α_i or α_i^* are called support vectors and will enter into the expansion of $f(x)$ in (2). The regression hyper-plane is expressed as:

$$\hat{y}=f(x)=\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)k(x_i,x)+b \tag{7}$$

where $k(\cdot)$ is kernel function, b is the average constant over all non-zero support vectors.

Note that different kernel functions implicitly define the form of mapping and the feature space, thus they determine how well the nonlinearity of a system can be captured. The most widely used kernel functions are shown in Table I. All these kernels compose a kernel function library. In Table I, q,d,δ,u,p denotes the kernel parameters, which reflect the characteristics of the training data, affect the performance of the SVR, and determine a location of the hyper-plane. We note that it is necessary to optimize all parameters in SVR, including the error penalty parameter C, the insensitivity loss parameter ε, as well as these kernel parameters.

B. Parameters Optimization of SVR by PSO Algorithm

The solution to the SVR problem is not a simple indicator function and there are more complicated parameters to design. Here, a novel integrated framework based on PSO algorithm for SVR parameter optimization is proposed to obtain better prediction performance.

PSO algorithm is an evolutionary computation technique that has been used successfully in optimization [25]. The method uses a swarm of particles to represent the potential solutions to a problem. Each particle in the swarm is characterized by its position and a velocity. In an iterative fashion, the particles adjust their velocity so that they start moving towards the optimal solution. There are two important characteristics, the best previous position *pbest* and the overall best position *gbest*. The search process determines the best *gbest* so that the corresponding particle's fitness reaches its best value [22, 25]. Here, a particle swarm optimization-based SVR (PSO-SVR) model is designed, in which PSO algorithm is used to determine the parameters of SVR. These parameters consist of the error penalty parameter C, the tube parameter ε, and the kernel parameters. Each particle position P_i optimizes a suite of these parameters. The dimensionality (L) of the particle depends upon the number of the parameters being optimized. Table II lists the definition of the particles' position and velocity depending on different kernel functions used in the model.

The PSO-SVR can automatically determine the parameters of SVR and control the predictive accuracy and generalization ability simultaneously. The overall framework of the PSO-SVR is depicted while the optimization proceeds as follows:

Step 1: Collect the training samples and select a kernel function from kernel function library for the SVR predictor;

Step 2: Search SVR optimal parameters by engaging a PSO algorithm:

Step 2.1: Initialize n particles in a population $U=\{P_1,P_2,...,P_i,...,P_n\}$, each particle has the same dimension size L according to the kernel function type, for example, if the radial basis kernel function is selected, its particle is $P_i=(C_i,\varepsilon_i,\delta_i)$, L=3.

Step 2.2: For each particle, determine a value of its fitness function expressed in the form

$$fitness(t)=\sum_{i=1}^{n}\frac{(\hat{y}_i-y_i)^2}{n} \tag{8}$$

where the fitness expresses the deviation from the particle predictive value $\hat{y}$ and the true value y. The particles move towards a direction where low fitness values are reported;

Step 2.3: Select the particle with the lowest fitness value in overall particles in all generations as the best overall position (*gbest*) P_{gd} and the smallest fitness value in current generation as the best current generation position (*pbest*) P_{bd};

Step 2.4: Update the velocity and position for each particle P_i in t-th generation:

$$v_{id}(t+1)=\omega v_{id}(t)+C_1r_1(P_{bd}-P_{id}(t))+C_2r_1(P_{gd}-P_{id}(t)) \tag{9}$$

$$P_{id}(t+1) = P_{id}(t) + v_{id}(t) \qquad (10)$$

where $1 \le i \le n$, $1 \le d \le L$, C_1 and C_2 are acceleration coefficients, quite commonly both of them are set to 2; r_1 and r_2 are two random variables in the range [0, 1]; $\omega = 1/t$, is an inertia weight, whose values decline linearly over successive generations (t);

Step 2.5: Check the termination condition. If the maximum number of iterations is reached or the required minimal fitness error accuracy is obtained, return the *gbest* P_{gd} ; otherwise go back to Step 2.2;

Step 3: Obtain the fitting hyper-plane function (7) of SVR for the optimal parameters already obtained.

III. Global Nonlinear Kernel Adaptive Prediction Approach

A. The PSO-ISVR Predictor

The idea of the PSO-ISVR predictor originates from the piecewise support vector machine (SVM) [26] and the Takagi-Sugeno (TS) fuzzy model [27]. The TS fuzzy model comprises local input-output relations of a nonlinear system. In the model, local controllers are designed for each subsystem by fuzzy approximation rules to obtain the local optimum. In the sequel such local controllers are aggregated by considering their weights to produce a control value. The underlying concept of PSO-ISVR is derived from TS fuzzy model, and the only difference is that the PSO-ISVR is looking for an optimization regression model by the PSO algorithm in subsystem not by considering a fuzzy rule. On the other hand, the idea of splitting the data space originates from the piecewise SVM [26]. But the piecewise SVM does not consider the kernel selection and the parameter optimization realized in a given subspace as well as the generation of the regression model. The overall process of ISVR-PSO based sliding adaptive control predictor is displayed in Fig. 1. First of all, the sample space is divided into m subspaces, as shown in Part I of Fig. 1. In each subspace, a sliding adaptive switcher is used to select the optimal kernel and form its optimal parameters set so as to obtain the local optimal input-output relations of a nonlinear system to fit the data. Hence, a group of kernel candidate units C_1, C_2, …, C_k are designed to select the kernel function from the kernel function library, as illustrated in Part II of Fig. 1. In fact, each kernel candidate unit C_i corresponds to a certain kernel function. The goal of C_i is to gain an optimal subspace SVR fitting hyper-plane of its kernel function. Here, the PSO algorithm is used to adjust the SVR parameters and form the optimal subspace hyper-plane, as shown in Part IV of Fig. 1. At last, by using Lagrange interpolation surfaces' join algorithm [28], the optimal fitting hyper-planes for the subspace segments are aggregated to create a global nonlinear predictor for new data, as shown in Part III of Fig.1. In the following, the ISVR-PSO predictor is illustrated in details and its generalization performance and execution speed are analyzed.

(1) *Sample space division*

The training data need to be normalized firstly. The maximum and minimum values of each variable are determined and the samples space Ω is divided into m consecutive subspaces Ω_i of equal size $\Delta x = (x_{\max} - x_{\min})/m$, where $\Omega_i \cap \Omega_j = \phi, i \ne j, i, j = 1, 2, \ldots, m$.

(2) *Subspace adaptive switching and hyper-plane fitting*

In each subspace Ω_i , the PSO-ISVR sliding adaptive switch is used to select the optimal kernel function from the kernel function library to fit the input data. The switch mechanism belongs to a class of control selecting methods, refer to Part II of Fig. 1. The switch consists of three elements: (1) The parallel kernel candidate units set $\{C_1, C_2, \ldots C_k\}$, which controls the kernel function selection; (2) The controlling decision-making device D, which evaluates the performance of kernel candidate units; (3) A switch unit P which switches among different kernel candidate units. P switches to the optimal kernel candidate unit C_k in candidate kernels set, outputs and saves its fitting hyper-plane into the storage 1.

Given the training sample set j-th as ($x^{(j)}$, $y^{(j)}$)={($x_i^{(j)}$, $y_i^{(j)}$), i=1, 2,... ,l_j} in the j-th subspace, j is an subspace index, not an exponent, where $x^{(j)}$ is the input data, $y^{(j)}$ is the actual value, $\hat{f}_j(x)$ is the predicted output in Fig.1, l_j is the number of samples for the training set in j-th subspace. The switching unit P is used to find an optimal kernel candidate unit C_k and produces its optimal parameters configuration set $\hat{\theta}_{jk}$, which will be used in the kernel function as well as the optimal hyper-plane $\hat{f}_{jk}(x)$ in SVR. The optimal SVR model in set $\hat{\theta}_{jk}$ makes the mean square error (*MSE*) J minimal, namely

$$J(\hat{\theta}_{jk}) = min(\hat{f}_{jk}(x^{(j)} \mid \hat{\theta}_{jk}) - y^{(j)})^2 \qquad (11)$$

where k denotes the k-th kernel candidate unit, $\hat{f}_{jk}(x^{(j)} \mid \hat{\theta}_{jk})$ is the predicted value of $x^{(j)}$ obtained for the optimal parameters produced by the PSO-SVR. The optimal prediction is corresponding to Part IV in Fig. 1. Each kernel candidate has its $J(\hat{\theta}_{jk})$ in the j-th subspace. The decision D evaluates the performance of the kernel candidate units by their $J(\hat{\theta}_{ji})$ (i=1, 2,…,k). Switching unit P switches to the optimal kernel candidate unit C_k with the *min*($J(\hat{\theta}_{j1})$, $J(\hat{\theta}_{j2})$ ... $J(\hat{\theta}_{jk})$), and outputs its fitting hyper-plane $\hat{f}_j$ in the j-th subspace j=1, 2, …, m.

(3) *Subspaces link and fitting hyper-planes connect*

In every subspace, it will produce an optimal fitting hyper-plane, but each fitting hyper-plane $\hat{f}_j(x)$ of subspace Ω_j is relatively independent from the others. Therefore, the overall fitting hyper-planes at the subspace boundaries of the entire space are not continuous and sometimes may exhibit jumps, as shown in Fig. 2. In order to eliminate this discontinuity, we add a connection decision function to construct a buffer zone near the borders of each subspace, where the endpoints between the optimal fitting hyper-planes $\hat{f}_j(x)$ and $\hat{f}_{j+1}(x)$ are connected by Lagrangian three points interpolation [28].

The decision function $\hat{f}_j(x)$ of subspace Ω_j consists of part

A and part B. Part A is the subspace fitting hyper-plane, while part B is the buffer zone between the subspaces. Given that the buffer zone set is $[x_j-\Delta x, x_j]$ in subspace Ω_j , we select three points (u_i, v_i), i =1, 2, 3 which form a quadratic fitting function by Lagrangian three points interpolation method completed in the buffer zone.

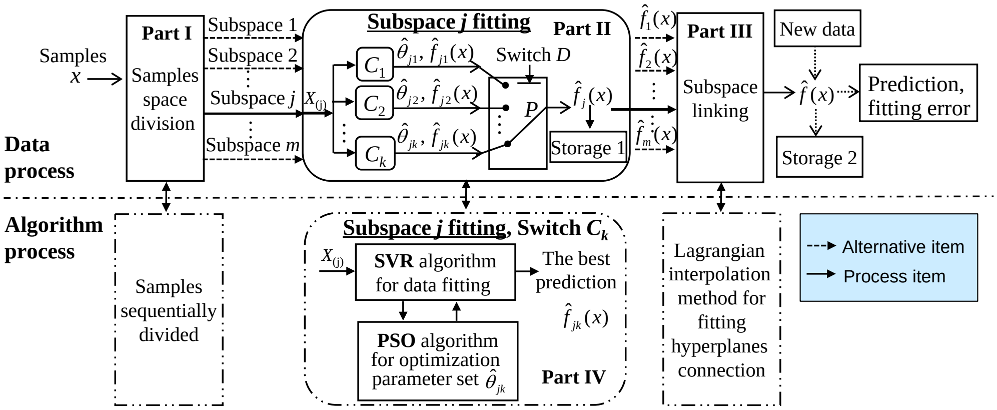

Fig. 1. The PSO-ISVR predictor.

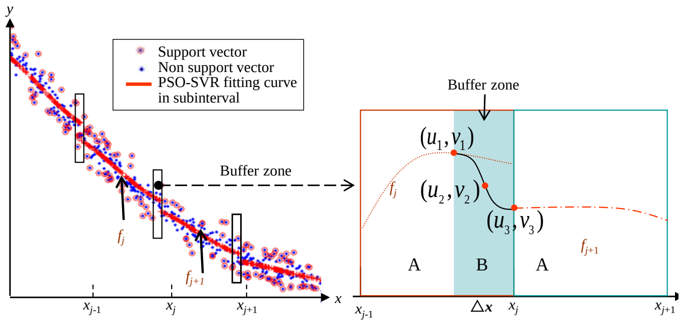

Fig. 2. Fitting hyper-plane subspace link scheme.

Considering that the space Ω is separated into m subspaces, the overall decision function becomes

$$\hat{f}(x)=\sum_{j=1}^{m} I_{\Omega_j}(x)\hat{f}_j(x) \qquad (12)$$

where $\hat{f}_j(x)$ is the optimal fitting hyper-plane function in the subspace Ω_j , $I_{\Omega_j}(x)$ is an indicator function,

$$I_{\Omega_j}=\begin{cases}1 & x\in\Omega_j\\ 0 & x\notin\Omega_j\end{cases} \qquad (13)$$

The decision function $\hat{f}_j(x)$ of subspace Ω_j consists of A and B parts

$$\hat{f}_j(x)=\hat{f}_{jA}(x)+\hat{f}_{jB}(x) \qquad (14)$$

where

$$\hat{f}_{jA}(x)=\sum_{i=1}^{l_j}(\alpha_{ji}-\alpha_{ji}^{*})k(x_{ji},x)+b_j\,,\ \ x\in\Omega_j \qquad (15)$$

Here, α_{ji} , α_{ji}^{*} are non-negative Langrange multipliers, b_j is the translation component, $k(\cdot)$ is the kernel function, l_j stands for the number of training samples in subspace Ω_j , x_{ji} is the i-th training sample located in subspace Ω_j .

$$\hat{f}_{jB}(x)=\sum_{k=1}^{3}\Big(\prod_{\substack{i=1\\ i\neq k}}^{3}\frac{(x-u_i)}{(u_k-u_i)}\Big)v_k \qquad (16)$$

where $\Delta x=2(x_j-x_{j-1})/l_j$, $u_1=x_j-\Delta x$, $v_1=\hat{y}(x_j-\Delta x)$, $u_2=x_j-\Delta x/2$, $v_2=[\hat{y}(x_j-\Delta x)+\hat{y}(x_j)]/2$, $u_3=x_j$, $v_3=\hat{y}(x_i)$, $\Delta x=(x_j-x_{j-1})$.

B. *Generalization Abilities of the PSO-ISVR Predictor*

Given a function set $f(x,\alpha)$, $\alpha\in\Omega$, there are l observation samples $(x_1,y_1),(x_2,y_2)...(x_l,y_l)$, where the samples are independent, identically distributed, and follow some unknown joint probability $F(x, y)$. The PSO-ISVR estimation seeks an optimal prediction function $y=f(x,\alpha_0)$ in function set $f(x,\alpha)$ so that the prediction comes with a minimum expected risk in (17):

$$R(\alpha)=\int L(y,f(x,\alpha))dF(x,y)=\int (y-f(x,\alpha))^2 dF(x,y) \ (17)$$

where $R(\alpha)$ is the expected risk function, $\alpha\in\Omega$ is a generalized parameter of function, $L(y,f(x,\alpha))=(y-f(x,\alpha))^2$ is the loss function, a prediction loss of implied y by the use of $f(x,\alpha)$.

However, in practice, the loss function $L(y,f(x,\alpha))$ and the joint probability distribution $F(x,y)$ cannot be obtained directly. According to the law of large numbers, an arithmetic average can approximate the joint function $F(x,y)$ in a large dataset. Therefore, the empirical risk $R_{emp}(\alpha)$ can replace the expected risk $R(\alpha)$:

$$R_{emp}(\alpha)=\frac{1}{l}\sum_{i=1}^{l}(y_i-f(x_i,\alpha))^2\ ,\ \alpha\in\Omega \qquad (18)$$

The principle, which uses the empirical risk function to approach the expected risk function, is referred to the empirical risk minimization principle [1]. We refer to the correct prediction capacity between the input and the prediction as generalization performance in machine learning, and the relationship between empirical risk and expected risk is generally referred to the generalization bound in statistical learning theory. The goal of machine learning is to find a function from a function set of $f(x,\alpha)$ so that the expected risk $R(\alpha)$ approaches the lowest upper bound on the actual risk under a fixed, sufficiently small η when the empirical risk gradually converges to the expected risk with the number of samples *l* increasing. They form an important basis for the analysis of performance of machine learning schemes and help in the development of a new algorithm.

To explore the PSO-ISVR generalization, the upper bound of converging speed in empirical risk $R_{emp}(\alpha)$ close to expected risk $R(\alpha)$ and the minimum value difference between the expected risk $R(\alpha)$ and the empirical risk $R_{emp}(\alpha)$ [1] in bound of function $L(y, f(x,\alpha))$ in compact space Ω . This relationship is captured in terms of **Theorem 1** and **Theorem 2**. Here, $L(y, f(x,\alpha))$ is defined in a finite domain for SVR, which forms a difference to [26] where SVM is unbounded set.

Theorem 1: For the bounded real function set, $A \le L(y, f(x,\alpha)) \le B$, $\alpha \in \Omega$, the following inequality establishes in probability $1-2m\eta$:

$$\begin{aligned} R(a) \le R_{emp}(\alpha)(1+\sqrt{\frac{-l\ln\eta}{2}}\sum_{j=1}^{m}\frac{1}{l_j}) \\ +\sum_{j=1}^{m}(\frac{l_j}{l}+\sqrt{\frac{-\ln\eta}{2l}})(B-A)\sqrt{\varepsilon_j(l)} \end{aligned} \tag{19}$$

where $\varepsilon_j(l)=\frac{h_j(\ln(2l/h_j)+1)-\ln(\eta/4)}{l}$, η is the subspace number, $0\le\eta\le 1$, h_j is a non-negative integer, called the Vapnik Chervonenk is (VC) dimension in Ω_j , l_j and *l* are the number of samples in subspace and the entire space respectively, $l=\sum_{j=1}^{m} l_j$.

The proof is given in **Appendix A**. **Theorem 1** offers the upper bound of the expected risk when empirical risk $R_{emp}(\alpha)$ converges to expected risk $R(\alpha)$ in probability η . The minimum risk difference between the expected risk and the empirical risk is provided in **Theorem 2**.

Suppose that $R(\alpha_0)$ is the minimization expected risk on function $L(y, f(x,\alpha))$ and $R_{emp}(\alpha_n)$ is the minimization empirical risk on function $L(y, f(x,\alpha))$, the upper bound $\Delta(\alpha)=R_{emp}(\alpha_n)-R(\alpha_0)$ is described in **Theorem 2**.

Theorem2: Assume that parameters $\alpha=(\alpha_1,\alpha_2,...,\alpha_l)\in\Omega$ and the expected risk $R(\alpha)$ has a solution set in subspaces which are decoupled, for the functions set $A\le L(y, f(x,\alpha))\le B$, the following inequality establishes in probability $1-3m\eta$:

$$\begin{aligned} \Delta(\alpha)=R_{emp}(\alpha_n)-R(\alpha_0)\le \\ \sum_{j=1}^{m}(\frac{l_j}{l}+\sqrt{\frac{-\ln\eta}{2l}})(B-A)\left[\sqrt{\varepsilon_j(l)}+\sqrt{\frac{-\ln\eta}{2l}}\right] \end{aligned} \tag{20}$$

The proof is given in **Appendix B**. **Theorem 2** describes the proximity degree of minimal empirical risk $R_{emp}(\alpha_n)$ close to the minimal expected risk $R(\alpha_0)$ in the empirical risk minimization principle, its risk bound is related to the VC dimension in $\varepsilon_j(l)$.**Theorem 1** describes the converge speed whereas **Theorem 2** describes the distance between the empirical risk $R_{emp}(\alpha_n)$ and the expected risk $R(\alpha_0)$. In virtue of **Theorem 1** and **Theorem 2**, the PSO-ISVR in joint subspaces can find the control risk bound, which provides a sound theoretical basis for the PSO-ISVR predictor.

C. *Speed Analysis of the PSO-ISVR Predictor*

Given that the number of PSO generations to *k*, there are *n* particles in a population. For the *j*-th subspace the training time of the SVR satisfies the relationship [26, 29]:

$$t_j=\alpha(l_j)^u nk \tag{21}$$

where *n*, *k* and α are constants; l_j is the number of training samples in *j*-th subspace. The size of *u* is related to the training algorithm, for example, if the sequential minimal optimization algorithm is used, only two examples are analytically optimized at every step, then $u=2$. But in Osuna's algorithm[29], a fixed number of examples are optimized and the same number of examples is discarded from the problem at every step, then *u* is the fixed number of examples in handling.

If the PSO-ISVR model splits the input space into *m* subspaces and there is the same number of samples $l_j=l/m$ in each subspace $j=1,2,...m$,where *l* is the number of training samples, then the overall training time *t* becomes

$$t=mt_j=ma\left(\frac{l}{m}\right)^u nk=am^{1-u}l^u nk \tag{22}$$

When *m*=1, viz. the data space is not split into regions, the total training time is $t=\alpha l^u nk$, which is m^{u-1} times of the estimate provided by (22) in the *m* interval spaces for the PSO-ISVR model. Thus the larger the number of subspaces *m* is, the higher the speedup is. It can be noted that the training time of the PSO-SVR rises faster than that of the PSO-ISVR when the number of the training samples increases.

It is interesting to search for the value of *m* for which the PSO-ISVR model can reach the optimal generalization performance. The piecewise SVM provides a sound view at the problem [26]. According to **Theorem 1**, the speed of $R(a)$ is proportional to $R_{emp}(\alpha)$ by $\sqrt{\frac{-l\ln\eta}{2}}\sum_{j=1}^{m}\frac{1}{l_j}$, and $l_j=l/m$ with the equal number samples in each subspace as follows:

$$\sqrt{\frac{-l\ln\eta}{2}}\sum_{j=1}^{m}\frac{1}{l_j}=m^2\sqrt{\frac{-\ln\eta}{2l}} \tag{23}$$

TABLE III
Initialization parameters and particles position search scope in PSO

Initialization parameters		Particles position search scope	
Parameters name	Value	Parameter name	Search space
Population size	n =100	SVR parameter	$C \in$[0.001,500] $\varepsilon \in$[0.001,1]
Acceleration coefficients	C_1=1.8 C_2=2.1	Polynomial kernel parameter	$d \in$[-1000,1000] $q \in$[1,10]
Constriction coefficient	ω = 1	RBF kernel parameter	$\delta \in$[0.01,500]
Maximal iteration generation	t =1000	Sigmoid kernel parameter	$\mu \in$[0.01,500] $p \in$[-1000,0]

From (23) we note that the higher the subspace number m is, the larger the distance between empirical risk $R_{emp}(\alpha)$ and expected risk $R(\alpha)$ in probability η. Then the generalization performance decreases even the training speed increases. Therefore, we have to consider both the speed and the performance of the predictor. As a matter of fact, this issue can be quantified as follows:

$$min\ f(m)=(m^2\sqrt{\frac{-\ln\eta}{2l}}+m^{1-u}) \qquad (24)$$

It can be used to decide upon the optimal number of subspaces, where η is a certain probability. By derivative extremum analysis $\frac{df(m)}{dm}=0$, then

$$2m\sqrt{\frac{-\ln\eta}{2l}}+(1-\mu)m^{-u}=0\,.$$

Hence, the optimal subspace number m is

$$m=[(u-1)\sqrt{\frac{l}{-2\ln\eta}}]^{\frac{1}{u+1}}$$

when u=2 in sequential minimal optimization algorithms, we obtain the optimal subspace number m to be

$$m=(\frac{l}{-2\ln\eta})^{\frac{1}{6}} \qquad (25)$$

The division of the input space is essential to the successful performance of the PSO-SVR. This split exhibits a direct impact on the final prediction performance and the generalization abilities of the predictor. The number of subspaces not only affects the training speed of the SVR, but also impacts the prediction performance of the PSO-ISVR. Therefore, the selection of the number of subspaces requires a thorough attention.

IV. Experimental Results

In this section, we conducted a series of experiments to illustrate how the PSO-ISVR functions and show its performance. We use synthetic data and the stock volume weighted average price (VWAP) of real-world large data coming from Shanghai stock market exchange index in China.

A. Noise Functions

1) *Noise function* 1

We consider a single-variable function affected by noise and described as follows:

$$f(x)=\begin{cases} x^2 & x\le 0 \\ 3+x & x>0 \end{cases} \quad +noise\sim N(0,1), \qquad -5\le x\le 5$$

where $N(0,1)$ is a normal distribution with zero mean and variance of 1. Here we illustrate how the PSO-ISVR functions and quantify the training time and accuracy of the PSO-ISVR model. Also, we compare the obtained results with those produced by the PSO-SVR algorithm. In the experiment, we randomly generate 500 points treated as the training data and subsequently 500 points as the testing data. Here, the kernel function library includes the linear kernel, the Polynomial kernel, the RBF kernel and the Sigmoid kernel function as shown in Table I. For all the experiments, the PSO initialization and the particle search space are listed in Table III. In this table, d, q, δ, u, p are kernel parameters, C is penalty parameter, and ε is the SVR parameter. An integrated tool for SVR and classification, libSVM [30], is improved by combining with PSO algorithm. The detailed computer setting concerns AMD 2.2GHz and 2.0GB RAM.

In the PSO-SVR experiment, the entire training set (x, y) is input to the PSO-SVR algorithm. Then the sliding switch selects the RBF kernel function as the optimal kernel function because it has the minimum MSE = 66.4144 in 4 kernel functions, the optimal parameters C=18.7632, ε = 0.32975, δ=0.121 is obtained by the PSO algorithm. The optimal RBF fitting optimal hyper-plane to observe the fitting result is shown in Fig. 3. It can be seen that the deviation of fitting in middle, the beginning and the terminal points is larger than the other points in the PSO-SVR testing.

In PSO-ISVR experiment, the training samples are divided into 11 successive sub-intervals which are used by the adaptive PSO-ISVR model. Table IV contains the optimal training result of PSO-ISVR model in each sub-space and the overall comparing result with the PSO-SVR model to the noise function 1, including the kernel function, the optimal parameter, the MSE in training model and the algorithm time consumption in each subspace. It can be seen that the time consumption of PSO-ISVR algorithm is far less than that of PSO-SVR algorithm and the PSO-ISVR algorithm has higher fitting accuracy than that of the PSO-SVR in spatial prediction. Fig. 4 shows the fitting result of the test sample points in each sub-interval in noise function1. It can be seen that there are discontinuous phenomenon in adjacent sub-spaces, the subspace are connected by using Lagrange three-point interpolation and output the overall decision-making function. Fig. 5 shows the final approximation result of the test data. It has a good fitness in the entire space.

On the other hand, the generalization abilities of the PSO-ISVR are examined in experiments. Owing to the speed of the empirical risk $R_{emp}(\alpha)$ approximating to the expected risk $R(a)$ by (23), the approach speed from the empirical to expected risk is $11^2\times\sqrt{\frac{-\ln 0.01}{2\times 500}}=8.21$ in the PSO-ISVR and is $1^2\times\sqrt{\frac{-\ln 0.01}{2\times 500}}=0.06$ in the PSO-SVR under the assumption of $\eta=0.01$. The distance between the $R(\alpha)$ and the $R_{emp}(\alpha)$ increases $8.21/0.06\approx 121$ times for the PSO-ISVR comparing with the PSO-SVR. It means the generalization abilities of the PSO-ISVR decrease from 8.21 to 0.06, and the over-fitting phenomenon occurred although its MSE 6.096 and computing overhead of 0.5299s for m=11 is less than the whole PSO-SVR's MSE 66.4144 and 91.4051s in m=1. The ideal division of the input space is round $(\frac{500}{-2\ln 0.01})^{\frac{1}{6}}\approx 2$ by (24) and (25). Here, the generalization ability and algorithm velocity is tradeoff. In 2 division experiments for the PSO-ISVR, the average MSE and the consumption time is 20.218 and 2.565s, respectively. It is far less than those of the PSO-SVR, which is 66.4144 and 91.4051s, respectively, as shown in Table IV. That is to say, the overall performance of the PSO-ISVR is better.

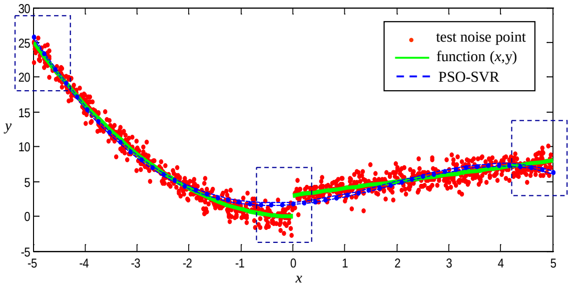

Fig. 3. Original noise function 1 and the PSO-SVR prediction - testing data.

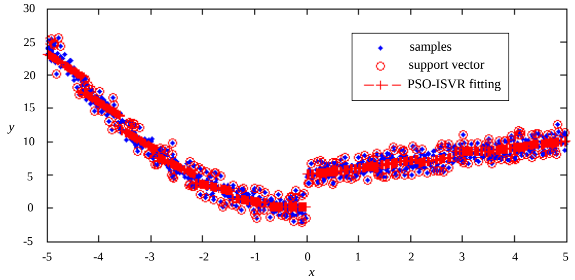

Fig. 4. The prediction results for noise function 1 by PSO-ISVR in testing data.

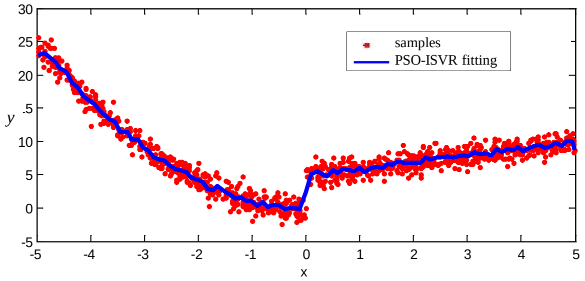

Fig. 5. The fitting result of the test sample points in PSO-ISVR by using Lagrange three interpolation connection.

TABLE IV
PSO-SVR AND PSO-ISVR RESULTS FOR NOISY FUNCTION 1 (TRAINING SAMPLES)

NO	The best kernel function	Optimal parameters	*MSE*	Consumption time (*s*)
1	Linear	C = 100 ε= 0.3088 q =1， d =0	0.1946	0.0210
2	Linear	C = 17.7253 ε= 0.4951 q =1， d =0	0	0.0435
3	RBF	C = 9.8055 ε= 1 δ= 0.0271	0	0.0797
4	RBF	C = 3.0164 ε= 0.5334 δ= 0.821	0.0787	0.0735
5	RBF	C = 62.5812 ε= 0.0359 δ= 0.01	0	0.1043
6	RBF	C = 30.5344 ε= 0.3177 δ= 0.021	0.0987	0.0406
7	RBF	C = 2.53117 ε= 1 δ= 0.101	0.0717	0.0271
8	RBF	C = 51.1719 ε= 0.1808 δ= 0.352	0	0.1544
9	RBF	C = 81.2194 ε= 0.1787 δ= 0.891	0.0127	0.0159
10	RBF	C = 51.1671 ε= 0.2732 δ= 0.01	4.1896	0.0837
11	RBF	C = 18.3878 ε= 1 δ= 0.211	0	0.0406
	PSO-ISVR	**sum**	6.0960	0.5299
	PSO-SVR	C = 18.7632 ε= 0.3298 δ= 0.121	66.4144	91.4051

than that of the PSO-SVR.

2) *Noise function* 2

In order to further evaluate the PSO-ISVR model's performance, another noise function is considered.

$$f(x)=\begin{cases}\sin(x^2)+noise\sim N(0,0.1), 0<x\le 5\\ \cos(x^2-1)+noise\sim N(0,0.1), -5\le x\le 0\end{cases}$$

where N(0, 0.1) has a normal distribution with zero mean and variance of 0.1. For comparative purposes, we consider two other prediction methods, Back-Propagation Neural Network (BPNN) and Cubic Spline Curve Fitting (CSCF) model [31]. A standard three-layer BPNN network has a single node in the input layer and a single node in output layer, and the number of hidden nodes varying from 2 to 10. In the CSCF, the interpolation interval is divided into subintervals, and over each subinterval we carry out interpolation by using the cubic polynomial. The polynomial satisfies the condition of continuity at each endpoint. Since the CSCF doesn't pass through all given points, there are many random noise points in the experiment, which are overlapped, repeated, and even inconsistent. We use subinterval average as the curve fitting input. Here, the CSCF has the same interval division as the PSO-ISVR. In each experiment, we randomly generate 2,000 points as the training data and then 2,000 points to form the testing data. According to (25), it is appropriate to set the subinterval number in the PSO-ISVR to around $(\frac{2000}{-2\ln 0.01})^{\frac{1}{6}}\approx 3$.

In total, 10 experiments are conducted for each model repeatedly. We compare evaluation process of the PSO-ISVR with that of the CSCF and the BPNN in 3 continuous subspaces, as shown in Fig.6. The performance of all models is evaluated by calculating the root mean square error (*MSE*) and inequality coefficient *U*.

$$MSE=\frac{1}{l}\sum_{i=1}^{l}(y_i-\hat{y}_i)^2 \qquad (26)$$

$$U=\frac{\sqrt{\frac{1}{l}\sum_{i=1}^{l}(y_i-\hat{y}_i)^2}}{\sqrt{\frac{1}{l}\sum_{i=1}^{l}y_i^2}+\sqrt{\frac{1}{l}\sum_{i=1}^{l}\hat{y}_i^2}},\ 0<U<1 \qquad (27)$$

where $\hat{y}_i$ is the predicted value, y_i is the actual value, l is the

TABLE V
MSE AND U VALUES OF CSCF, BPNN AND PSO-ISVR MODELS IN NOISE FUNCTION 2

Experiments	CSCF		BPNN		PSO-ISVR	
times	*MSE*	*U*	*MSE*	*U*	*MSE*	*U*
1	3.45	0.36	1.63	0.27	0.83	0.21
2	4.53	0.44	1.51	0.36	0.96	0.19
3	4.22	0.28	1.51	0.22	1.03	0.17
4	4.87	0.39	1.81	0.24	1.11	0.20
5	5.53	0.55	1.64	0.21	1.16	0.15
6	4.67	0.47	2.27	0.41	0.72	0.20
7	4.32	0.48	1.54	0.28	1.08	0.16
8	5.21	0.51	1.64	0.32	1.39	0.13
9	3.55	0.39	1.67	0.22	1.21	0.16
10	5.56	0.36	1.64	0.19	1.42	0.14
Average	4.591	0.423	1.686	0.272	1.091	0.171

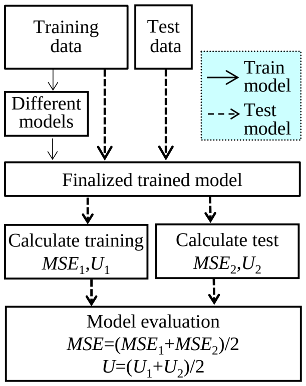

Fig. 6. The process of model evaluation.

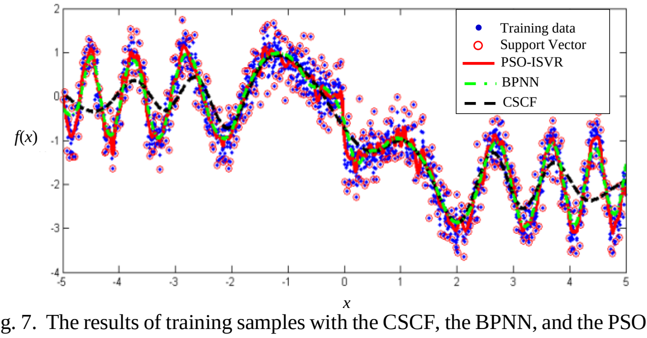

Fig. 7. The results of training samples with the CSCF, the BPNN, and the PSO-ISVR.

index of the data to be predicted. Generally, the model predictive ability is inversely proportional to *MSE* and *U*, especially when *U* tends to 1, the predictive power of model is the worst. The values of the *MSE* and *U* obtained for the training and testing dataset are reported in TABLE V. It can be seen that the values of the *MSE* and *U* provided by the PSO-ISVM are smaller than the ones obtained for the CSCF and the BPNN. In other words, the forecasting accuracy produced by the PSO-ISVM excels the ones produced by the CSCF and the BPNN.

In addition, we observe that the forecasting accuracy of the PSO-ISVR and the BPNN does not depend on the start and the end intervals, while the CSCF did. Fig. 7 shows the results of comparative analysis of the CSCF, the BPNN, and the PSO-ISVR for the training samples. The PSO-ISVR always captures the change point and exhibits better fitting than the CSCF and the BPNN, especially, when the volatility of the endpoints is high. In fact, this phenomenon also is presented in the testing data.

B. *Prediction of Stock Volume Weighted Average Price*

In this part, the importance of the parameters optimization and the kernel function selection in PSO-ISVR are illustrated by an application of VWAP prediction in a big stock data. Its performance is verified by comparing with the SVR and the PSO-SVR models respectively. In an electronic trading system, the VWAP is defined as the ratio of traded value to total traded volume in a certain time horizon (such as one day or an hour). It is a measure of an average price in a stock trading. The VWAP often becomes an optimal benchmark and an implementation strategy. An order with enough large volume is decomposed into smaller suborders, then trade the small suborders throughout a specified period gradually by sequence so that their trading cost is less than or equal to the VWAP value in the corresponding time period [32]. By doing so, it can reduce the market impact, increase the profitability of investors' transactions, and making the selling or the buying in large amounts of shares more secret.

A reliable historical data and a second-level market real-time data determine the success gain or its lack in the VWAP at the end of the process. If stock market participants want to trade at a price as close as possible to the VWAP benchmark [33], they need a model to predict the VWAP by using the real-time existing data on current stock market. Here, we use a stock VWAP in five minutes to predict its new coming five minutes volume of the VWAP in the stock market.

Assuming that the current trading VWAP is $y(t)$, its prediction volume is $\hat{y}(t)$ in five minutes. The prediction volume is associated with seven factors of the current five minutes trade: opening price (*OPRC*), the highest price (*HPRC*), the lowest price (*LPRC*), closing price (*CPRC*), stock trading volume (*STD*), trading turnover (*TT*), and the current VWAP. Let us arrange these factors in a vector form $y(t)$.We use $y(t)$ to predict $\hat{y}(t)$.

$$y(t)=\{OPRC(t-5),HPRC(t-5),LPRC(t-5),\\ CPRC(t-5),STD(t-5),TT(t-5),VWAP(t-5)\} \quad (28)$$

$$\hat{y}(t)=VWAP(t) \quad (29)$$

(1) *Data sets*

We use an actual stock trading data coming from the Shanghai Stock Exchange, China (*http://english.sse.com.cn/*). The trading data are with 896 trading days, where the first point corresponds to the time moment 9:30 AM and the last to the time 3:00 PM. The time interval between two succeeding time points is five minutes. Therefore, there are 48 points available in a day and this yields 43,008 records during these days. Excluding 63 session records during the periods, we use the remaining 42,945 records to complete analysis and predict the coming VWAP volume in the succeeding five minutes by using the PSO-ISVR method. In practice, these records measure is often from differential metric unit and has the differential expression of variation; the higher values would drive the training process and mask the contribution of lower valued inputs. In such a case, we normalize the data within a specified range (here going from 1 to 2) to reduce the risk of the difference measure. This preprocessing becomes necessary before starting data analysis.

(2) *Prediction results and their analysis*

1) *Selection of parameters and prediction accuracy*

The 42,945 records are separated into 896 subspaces by trading days, each of which comprises 48 records. In each subspace, 24 records are used as a training data to estimate the parameters of each kernel functions by the PSO-ISVR. In the sequel, 24 records are used as the testing set. Finally, we select the kernel and parameters which lead to the lowest value of the fitness function.

Table VI shows the selected parameters obtained for different kernel functions produced by the PSO-ISVR in a subspace. According to the results, a fact has been reflected that the different kernel selection in SVR is crucial, four kernel types have different values of their resulting *MSE*s in their own best parameters in a subspace. The kernel type exhibits a significant impact on the quality of prediction. Here, the RBF kernel function is chosen as the best option for all kernel function types in the subspace due to its lowest *MSE* values.

On the other hand, in order to demonstrate the importance of the parametric optimization, we choose another group parameters of the RBF kernel in the subspace, which slightly deviate from the optimal values of the parameters. As shown in Table VI, the parameters (C=0.12, ε=0.276, δ=0.03) in the SVR differ slightly from the optimal setting (C=0.10,ε=0.2332,δ=0.01). However, the quality of prediction decreases significantly moving up from 0.0209 to 0.3980. Fig.8 shows the difference of the prediction results in the subspace using the SVR and the PSO-ISVR. In Fig.8, the x axis is a day trading records in 5 minutes interval, the y axis is the trade volume of the VWAP. It can be seen that the predicted result of the PSO-ISVR for the coming VWAP in five minutes are close to the original data than that of the SVR, which is essential to real-time stock trading. The PSO-ISVR integrates two processes of the parameters optimization and the kernel functions selection to gain the best fitness and the highest accuracy.

In addition, in order to verify the PSO performance for the parameter selection of the SVR. The searching process for optimum parameters in PSO is studied in detail. Fig.9 illustrates the searching process of the PSO for SVR optimum parameters of the RBF kernel function in a subspace. Its optimal fitness decreases in successive generations of the algorithm, but its average fitness of each generation is fluctuated. The large fluctuation of average fitness from one generation to another generation indicates that the search ability of the overall particles is strong. It is easier for these particles to jump out the local minima or the local maxima and obtain the global optimal parameters. The reliability of searching optimum parameters has been tested in PSO.

In order to illustrate the reliability of the PSO algorithm in the realization of the SVR parametric optimization, the genetic algorithm (GA) [25] is compared with the PSO in SVR (RBF) for a day VWAP prediction. The computational efficiency and the search velocity are the key objectives expressing the efficiency of the optimization. Therefore, attributes such as the fitness *MSE* of the SVR, the generation of the first search optimization parameter become particular interest. A typical GA composes three main operators, selection, crossover, and mutation [25]. The selection probability of 0.9 identifies the chromosomes of the current population. The selected chromosomes mate and generate a new offspring by invoking the crossover operation with probability of 0.8. Offsprings are mutated by the mutation probability of 0.15. The initial parameters of the GA are the same as those used for the PSO: population size is set to 100 and the maximum number of iterations is 1,000. Table VII shows the comparative results of the generation of the first search optimization parameter and the *MSE* in finding the best fitness for a trading day in the SVR model with RBF kernel by using the PSO and the GA respectively. It is clearly shown that the PSO is more accurate than the GA in this optimization problem. In addition, the computational overhead of the PSO is lower than the one encountered in the GA.

TABLE VI
PARAMETERS IN KERNEL FUNCTIONS CHOSEN BY THE PSO-ISVR IN THE VWAP

Methods	Kernel function	C	ε	Kernel parameters		*MSE*
PSO-ISVR	Line	1.4183	0.6856	-	-	1.0656
	Poly	73.1332	0.4399	q =1	d =-8.0429	0.6043
	RBF	**0.1000**	**0.2332**	**δ=0.01**	**-**	**0.0209**
	Sigmoid	8.7189	1	w =0.01	p =-267.938	7.1248
SVR	**RBF**	**0.1200**	**0.2760**	**δ=0.03**	**-**	**0.3980**

TABLE VII
COMPARISON OF THE PSO, THE GA AND THE VARYING PARAMETERS ALGORITHMS FOR THE SVR(RBF) IN A DAY TRADING

Types	Fitness *MSE*(%)		The generation of the first search
	Training	Testing	Training
GA	0.3017	0.4381	17
PSO	0.2880	0.3980	5

TABLE VIII
COMPARISON OF VWAP PREDICTION ANALYSIS ON THE PSO-ISVR AND THE PSO-SVR RESULTS IN A WEEK 5 DAYS

	Samples number	C	ε	Kernel parameters	Fitting	Time(s)	MSE
PSO-ISVR	48	0.1125	0.3545	δ=0.01	0.0000186	0.5390	0.0210
	48	0.1238	0.2612	δ=0.10	0.0000156	0.4924	0.0176
	48	0.1014	0.4100	δ=0.01	0.0000524	0.5538	0.05926
	48	0.1026	0.3360	δ=0.02	5.70×10^{-28}	0.54918	6.41×10^{-25}
	48	0.1026	0.2150	δ=0.01	3.86×10^{-7}	0.5602	0.000437
PSO-ISVR sum	240					**2.6945**	**0.0983**
PSO-SVR	240	61.82	0.6611	40.5587	0.0631	**58.7945**	**6.26735**

Note that the RBF function is selected as the kernel function here.

2) *Performance analysis in the overall space*

We compare the computing speed and prediction accuracy rates of the PSO-ISVR with those of the PSO-SVR with 5 subspaces in 5 days. The PSO-ISVR forms its own best kernel and optimal parameters, respectively in 5 subspaces in the 5 days. The results are reported in Table VIII. It can be seen clearly that the PSO-ISVR has smaller MSE and less time than the PSO-SVR's. The PSO-ISVR using interval division is superior to the PSO-SVM not only in model accuracy (smaller MSE), but also in time consumption for the weekly VWAP prediction. The space division can effectively decompose the complication of the SVR.

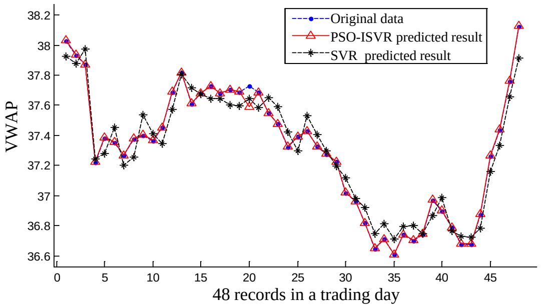

Fig. 8. The comparison of the original data and prediction results in the VWAP.

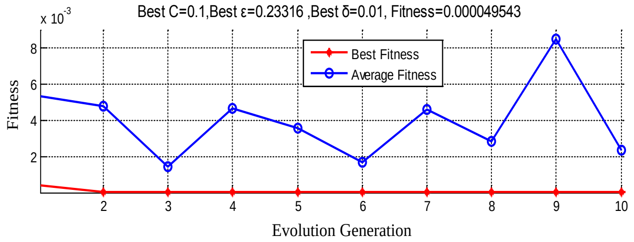

Fig. 9. Fitness function in successive generations.

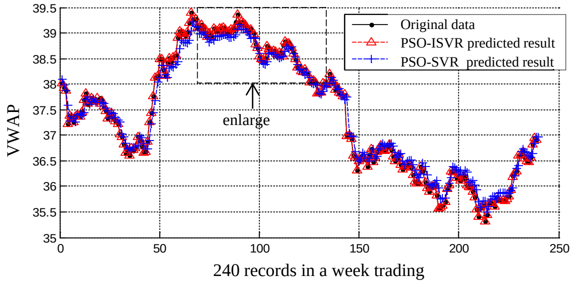

Fig. 10. The weekly VWAP prediction results obtained by the PSO-ISVR and the PSO-SVR.

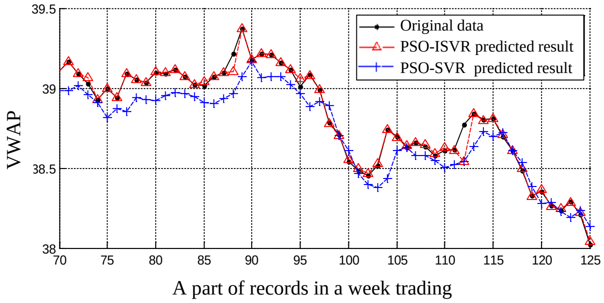

Fig. 11. The result comparison of the PSO-ISVR and the PSO-SVR in a weekly VWAP.

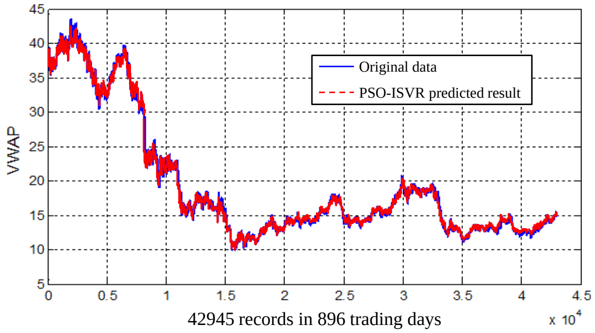

Fig.12. Overall VWAP prediction results by 5 minutes from 2007.07.02 to 2011.03.10.

Fig.10 shows the VWAP–generated prediction results in 5 days. By enlarging a portion of the VWAP prediction results as shown in Fig. 11, there is a clear difference between the results formed by the PSO-ISVR and the PSO-SVR. Noticeable, the PSO-ISVR prediction is around the original point tightly, but the PSO-SVR prediction results exhibit a certain bias.

Fig. 12 shows the prediction results for the entire data obtained with the use of the PSO-ISVR by day separation. The predicted values are close to the experimental data. The computing time is 259.35*s*, while the *MSE* value is 93.682. However, we were not able to run the SVR algorithm for the overall space because of its excessively large space requirements caused by the kernel matrix calculation.

V. Conclusions and Further Studies

In this paper, the novel PSO-ISVR model is proposed, where the feature space is divided into a number of subspaces. An optimal hyper-plane is constructed based on the PSO-SVR in each subspaces and the hyper-planes are then linked to form a global predictor. The main features of the PSO-ISVR model can be listed as follows: (1) An adaptive sliding switch in each subspace is designed, which can choose the optimal kernel function to fit the input data. (2) The PSO algorithm is used to adjust hyper-plane parameters in each subspace to earn the optimal SVR model. (3) The space division can effectively decompose the complication of the SVR, thus reducing the computing overhead and storage requirements, and facilitate further applications. In order to demonstrate its effectiveness, some synthetic data and the exchange index of the VWAP for current stock market are predicted and analyzed. The experimental results show that the PSO-ISVR is more accurate and faster than the PSO-SVR, and the PSO algorithm is reliable in seeking optimum SVR model.

There are several important directions worth investigating in the future. First, one can discuss how to construct the kernel functions and analyze their usage in the SVR. Secondly, the interactive relationships among the parameters can be introduced to the model. One might expect that such interaction could result in higher prediction accuracy and reduced computing time.

Appendix

Proof of Theorem 1

Denote loss function $L(y, f(x,a))$ as $Q(z,a)$, the expected risk function can be written as

$$R(\alpha) = \int Q(z,\alpha)dF(z)$$

$$Q(z,a)=\sum_{j=1}^{m} I_{\Omega j}(z)Q_j(z,a)=\sum_{j=1}^{m} I_{\Omega j}(z)L(y_j\,,\,f_j(x,a))$$

Hence, the expected risk function can be described as:

$$R(\alpha)=\sum_{i=1}^{m}\int_{\Omega_j} Q_j(z,\alpha)dF(z)$$

where, $\bigcap_{j=1}^{m}\Omega_j=\phi$, $\bigcup_{j=1}^{m}\Omega_j=1$.

According to the total probability formula, via $P(\Omega)=\sum_{j=1}^{m}p(\Omega_j)=1$, we have

$$R(\alpha)=\sum_{i=1}^{m}P(\Omega_j)\int_{\Omega_j} Q_j(z,\alpha)dF_j(z) \tag{30}$$

where $F_j(z)=\dfrac{F(z)}{P(\Omega_j)}I_{\Omega_j}(z)$ is the joint distribution.

Denote

$$R_j(\alpha)=\int_{\Omega_j} Q_j(z,\alpha)dF_j(z)$$

Hence,

$$R(\alpha)=\sum_{j=1}^{m}P(\Omega_j)R_j(\alpha) \tag{31}$$

This indicates that the overall expected risk is the weighted average of each sub-space expected risk. For the bounded real function set $f(x,\alpha)$, which satisfies the conditions $A\le L(y,\,f(x,\alpha))\le B$, $\alpha\in\Omega_j$, the minimizing experience risk function $R_{j,emp}(\alpha)$ satisfies the following inequality in probability η [1]:

$$R_j(\alpha)\le R_{j,emp}(\alpha)+(B-A)\sqrt{\varepsilon_j(l)} \tag{32}$$

where

$$\varepsilon_j(l)=\frac{h_j(\ln(2l/h_j)+1)-\ln(\eta/4)}{l}$$

$$R_{j,emp}(\alpha)=\frac{1}{l_j}\sum_{i=1}^{l_j}Q_j(z_{ij},\alpha)\,,$$

$0\le\eta\le 1$, h_j is a non-negative integer called the VC dimension in Ω_j, l is the samples number, *j*= 1, 2, ... *m*, *m* is the subspace number.

According to (31) and (32), the following inequality is established in probability $1-2m\eta$:

$$R(a)\le\sum_{j=1}^{m}P(\Omega_j)R_{j,emp}(\alpha)+\sum_{j=1}^{m}P(\Omega_j)(B-A)\sqrt{\varepsilon_j(l)}$$

On the other hand,

$$P(\Omega_j)\le\frac{l_j}{l}+\sqrt{\frac{-\ln\eta}{2l}} \tag{33}$$

Due to $R_{emp}(a)=\sum_{j=1}^{m}\frac{l_j}{l}R_{j,emp}(\alpha)$, according to (33), then

$$\begin{aligned}R(a)\le R_{emp}(\alpha)+\sqrt{\frac{-\ln\eta}{2l}}\sum_{j=1}^{m}R_{j,emp}(\alpha)\\+\sum_{j=1}^{m}\Big(\frac{l_j}{l}+\sqrt{\frac{-\ln\eta}{2l}}\Big)(B-A)\sqrt{\varepsilon_j(l)}\end{aligned} \tag{34}$$

Owing to $R_{emp}(a)=\sum_{j=1}^{m}\frac{l_j}{l}R_{j,emp}(\alpha)$, we get

$$\frac{l_j}{l}R_{j,emp}(\alpha)\le R_{emp}(a),\ \ j=1,2,..,m$$

Then

$$\sum_{j=1}^{m}R_{j,emp}(\alpha)\le\sum_{j=1}^{m}\frac{l}{l_j}R_{emp}(a),\ \ j=1,2,..,m \tag{35}$$

In virtue of (34) and (35), we can get

$$\begin{aligned}R(a)\le R_{emp}(\alpha)\Big(1+\sqrt{\frac{-l\ln\eta}{2}}\sum_{j=1}^{m}\frac{1}{l_j}\Big)\\+\sum_{j=1}^{m}\Big(\frac{l_j}{l}+\sqrt{\frac{-\ln\eta}{2l}}\Big)(B-A)\sqrt{\varepsilon_j(l)}\end{aligned} \tag{36}$$

Appendix B. Proof of Theorem 2

Suppose that $R_j(\alpha_0)$ is the gained minimization expected risk on function set $Q_j(z,\alpha_j)$ and $R_{j,emp}(\alpha_n)$ is the minimization empirical risk on function set $Q_j(z,\alpha_j)$ in Ω_j. According to statistical learning theory [1], the upper bound $\Delta_j(\alpha)=R_j(\alpha_0)-R_{j,emp}(\alpha_n)$ in probability $1-3m\eta$ is

$$\Delta_j(\alpha)=R_j(\alpha_0)-R_{j,emp}(\alpha_n)\le(B-A)\left[\sqrt{\varepsilon_j(l)}+\sqrt{\frac{-\ln\eta}{2l}}\right]$$

where *j* = 1, 2, ... *m*, *m* is the subspace number.

In light of (31) and (33), we obtain

$$\begin{aligned}\Delta(\alpha)=\sum_{j=1}^{m}P(\Omega_j)(R_j(\alpha_0)-R_{j,emp}(\alpha_n))\le\\ \sum_{j=1}^{m}\Big(\frac{l_j}{l}+\sqrt{\frac{-\ln\eta}{2l}}\Big)(B-A)\left[\sqrt{\varepsilon_j(l)}+\sqrt{\frac{-\ln\eta}{2l}}\right]\end{aligned} \tag{37}$$

References

[1] V. N. Vapnik, *The nature of statistical learning theory*: New York: Springer-Verlag, 1995.

[2] S. Kang, and S. Cho, “Approximating support vector machine with artificial neural network for fast prediction,” *Expert Systems with Applications,* vol. 41, no. 10, pp. 4989-4995, Aug. 2014.

[3] J. Shin, H. J. Kim, and Y. Kim, “An adaptive support vector regression for UAV flight control,” *Neural Networks,* vol. 24, no. 1, pp. 109-120, 2011.

[4] X. Li, L. Li, B. Zhang, and Q. Guo, “Hybrid self-adaptive learning based particle swarm optimization and support vector regression model for grade estimation,” *Neurocomputing,* vol. 118, pp. 179-190, Sep. 2013.

[5] C.-F. Huang, “A hybrid stock selection model using genetic algorithms and support vector regression,” *Applied Soft Computing,* vol. 12, no. 2, pp. 807-818, Feb. 2012.

[6] 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.

[7] A. Kazema, E. Sharifia, F. K. Hussainb, M. Saberic, and O. K. Hussaind, “Support vector regression with chaos-based firefly algorithm for stock market price forecasting,” *Applied Soft Computing,* vol. 13, no. 2, pp. 947-958, Feb. 2013.

[8] B. Chen, H. W. Liu, and Z. Bao, “Optimizing the data-dependent kernel under a unified kernel optimization framework,” *Pattern Recognition,* vol. 41, no. 6, pp. 2107-2119, 2008.

[9] F. Bellocchio, S. Ferrari, V. Piuri, and N. A. Borghese, “Hierarchical approach for multiscale support vector regression,” *IEEE Trans. Neural Networks and Learning Systems,* vol. 23, pp. 1448-1460, Sep. 2012.

[10] L. Jian, Z. H. Xia, X. J. Liang, and C. H. Gao, “Design of a multiple kernel learning algorithm for LS-SVM by convex programming,” *Neural Networks,* vol. 24, no. 5, pp. 476-483, 2011.

[11] A. Alexandridis, E. Chondrodima, and H. Sarimveis, "Radial Basis Function Network Training Using a Nonsymmetric Partition of the Input Space and Particle Swarm Optimization," *IEEE Trans. Neural Networks and Learning Systems*, vol. 24, pp. 219-230, Feb. 2013.

[12] K. Zhang, and J. T. Kwok, "Clustered Nyström method for large scale manifold learning and dimension reduction," *IEEE Trans. Neural Networks*, vol. 21, pp. 1576-1587, Oct. 2010.

[13] H. L. Shieh, and C. C. Kuo, "A reduced data set method for support vector regression," *Expert Systems with Applications*, vol. 37, no. 12, pp. 7781-7787, 2010.

[14] G. Huang, S. Song, C. Wu, and K. You, "Robust Support Vector Regression for Uncertain Input and Output Data," *IEEE Trans. Neural Networks and Learning Systems*, vol. 23, pp. 1690-1700, Nov. 2012.

[15] O. A. Omitaomu, M. K. Jeong, and A. B. Badiru, "Online support vector regression with varying parameters for time-dependent data," *IEEE Trans. Systems Man and Cybernetics Part A-Systems and Humans*, vol. 41, pp. 191-197, Jan. 2011.

[16] Y. Han, and G. Liu, "Probability-confidence-kernel-based localized multiple kernel learning with lp norm," *IEEE Trans. Systems Man and Cybernetics, Part B: Cybernetics*, vol. 42, pp. 827-837, Jun. 2012.

[17] P. Wittek, and C. L. Tan, "Compactly supported basis functions as support vector kernels for classification," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, pp. 2039-2050, Oct. 2011.

[18] N. M. Mark, and J. H. Shelton, "Using cross validation model selection to determine the shape of nonparametric selectivity curves in fisheries stock assessment models," *Fisheries Research*, vol. 110, no. 2, pp. 283-288, 2011.

[19] B. J. Álvaro, L. L. Jorge, and R. D. José, "Finding optimal model parameters by deterministic and annealed focused grid search," *Neurocomputing*, vol. 72, no. 13-15, pp. 2824-2832, 2009.

[20] J. Huang, Y. C. Bo, and H. Y. Wang, "Electromechanical equipment state forecasting based on genetic algorithm-support vector regression," *Expert Systems with Applications*, vol. 38, no. 7, pp. 8399-8402, 2011.

[21] S. Lin, K. Ying, S. Chen, and Z. Lee, "Particle swarm optimization for parameter determination and feature selection of support vector machines," *Expert Systems with Applications*, vol. 35, no. 4, pp. 1817-1824, 2008.

[22] J. Che, "Support vector regression based on optimal training subset and adaptive particle swarm optimization algorithm," *Applied Soft Computing*, vol. 13, no. 8, pp. 3473-3481, Aug. 2013.

[23] V. Ranaee, A. Ebrahimzadeh, and R. Ghaderi, "Application of the PSO-SVM model for recognition of control chart patterns," *ISA Trans.*, vol. 49, no. 4, pp. 577-586, 2010.

[24] X. Liang, "An effective method of pruning support vector machine classifiers," *IEEE Trans. Neural Networks*, vol. 21, pp. 26-38, 2010.

[25] R. C. Eberhart, Y. H. Shi, and J. Kennedy, "Swarm Intelligence," *The Morgan Kaufmann Series in Evolutionary Computation*, pp. 369-392: USA: Morgan Kaufmann Publishers, 2001.

[26] S. Q. Ren, D. G. Yang, X. Li, and Z. W. Zhuang, "Piecewise support vector machines," *Journal of Computers (in Chinese)*, vol. 2, no. 1, pp. 77-84, 2009.

[27] Z. Lendek, T. M. Guerra, R. Babuška, and B. D. Schutter, "Stability analysis and nonlinear observer design using Takagi-Sugeno fuzzy models " *Studies in Fuzziness and Soft Computing*, pp. 5-24: Springer-Verlag Berlin Heidelberg Press, 2011.

[28] T. Sauer, and Y. Xu, "On multivariate Lagrange interpolation," *Mathematics of Computation, American Mathematical Society*, vol. 64, pp. 1147-1170, 1995.

[29] J. Platt, *Studies in fast training of support vector machines using sequential minimal optimization*, Cambridge, MA: USA:MIT Press, 1999.

[30] Chang, C. Chung, and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1-27, 2011.

[31] P. J. Green, and B. W. Silverman, *Nonparametric Regression and Generalized Linear Models*, London: Chapman & Hall Press, 1994.

[32] M. L. Humphery-Jenner, "Optimal VWAP trading under noisy conditions," *Journal of Banking & Finance*, vol. 35, no. 9, pp. 2319-2329, 2011.

[33] Y. Xunyu, R. Yan, and H. Li, "Forecasting trading volume in the Chinese stock market based on the dynamic VWAP," *Studies in Nonlinear Dynamics & Econometrics*, vol. 18, no. 2, pp. 125-144, Aug, 2013.

Dr. Yongsheng Ding (M'00-SM'05) is currently a Professor at College of Information Sciences and Technology, Donghua University, Shanghai, China. He obtained the B.S., M.S. and Ph.D. degrees in Electrical Engineering from Donghua University, Shanghai, China in 1989, 1994 and 1998, respectively. From 1996 to 1998, he was a Visiting Scientist at Biomedical Engineering Center, The University of Texas Medical Branch, TX, USA. From February 2005 to April 2005, he was a Visiting Professor at Department of Electrical and Computer Engineering, Wayne State University, MI, USA. From September 2007 to February 2008, he was a Visiting Professor at Harvard Medical School, Harvard University, MA, USA. He serves as Senior Member of Institute of Electrical and Electronics Engineers (IEEE). He has published more than 300 technical papers, and six research monograph/ advanced textbooks. His scientific interests include computational intelligence, network intelligence, nature-inspired technologies, intelligent robots, Internet of things, bio-informatics, and digitized textile technology.

Dr. Lijun Cheng is currently a post-doctoral fellow in Purdue University and Indiana University, Indianapolis, US. She obtained the B.S. degree in Mathematics Application from Xinjiang Normal University in 1998, the M.S. degree in Computer Science from Xinjiang University in 2006, the Ph.D. degree in Control Science and Engineering from Donghua University, Shanghai, China in 2012. She has published more than 10 technical papers. Her current research interests are bio-computing, bio-informatics and machine learning.

Dr. Witold Pedrycz (M'88–SM'90–F'99) is currently Professor and the Canada Research Chair (CRC–Computational Intelligence) with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. He has authored or co-authored numerous papers in journals and conferences, and has authored 14 research monographs on computational intelligence and software engineering. His current research interests include computational intelligence, fuzzy modeling and granular computing, knowledge discovery and data mining, fuzzy control, pattern recognition, knowledge-based neural networks, relational computing, and software engineering. He was a recipient of the Norbert Wiener Award from the IEEE Systems, Man, and Cybernetics Council in 2007, the IEEE Canada Computer Engineering Medal in 2008. He is the Editor-in-Chief of Information Sciences. He is currently an Associate Editor of the IEEE Transactions on Fuzzy Systems and a member of a number of editorial boards of other international journals. In 2012, he was elected as a Fellow of the Royal Society of Canada.

Dr. Kuangrong Hao is currently a Professor at the College of Information Sciences and Technology, Donghua University, Shanghai, China. She obtained her B.S. degree in Mechanical Engineering from Hebei University of Technology, Tianjin, China in 1984, her M.S. degree from Ecole Normale Supérieur de Cachan, Paris, France in 1991, and her Ph.D. degree in Mathematics and Computer Science from Ecole Nationale des Ponts et Chaussées, Paris, France in 1995. She has published more than 100 technical papers, and two research monographs. Her scientific interests include machine vision and image processing, robot control, intelligent control, digitized textile technology.