Volatility Modelling and Prediction by Hybrid Support Vector Regression with Chaotic Genetic Algorithms

Phichhang Ou and Hengshan Wang
Business School, University of Shanghai for Science and Technology, China

Abstract: *In this paper, a new econometric model of volatility is proposed using hybrid Support Vector machine for Regression (SVR) combined with Chaotic Genetic Algorithm (CGA) to fit conditional mean and then conditional variance of stock market returns. The CGA, integrated by chaotic optimization algorithm with Genetic Algorithm (GA), is used to overcome premature local optimum in determining three hyperparameters of SVR model. The proposed hybrid SVRCGA model is achieved, which includes the selection of input variables by ARMA approach for fitting both mean and variance functions of returns, and also the searching process of obtaining the optimal SVR hyperparameters based on the CGA while training the SVR. Real data of complex stock markets (NASDAQ) are applied to validate and check the predicting accuracy of the hybrid SVRCGA model. The experimental results showed that the proposed model outperforms the other competing models including SVR with GA, standard SVR, Kernel smoothing and several parametric GARCH type models.*

Keywords: *Chaotic optimization, GA, CGA, SVR, volatility.*

Received November 15, 2011; accepted May 22, 2012; published online April 4, 2013

1. Introduction

Accurate forecasting of volatility is essential for investors and stock market players as the volatility is a key ingredient used for portfolio selection, option pricing, and for calculation of Value-at-Risk in risk management.

Recently, many researchers have used machine learning technique, Neural Network (NN), to improve the prediction of the financial volatility [2, 8, 9]. However, the NN suffers from over-fitting problems and the algorithm may have difficulty in obtaining a stable solution [23]. Support Vector machine for Regression (SVR), developed by Vapnik and coworkers [25], is a novel NN algorithm and it can overcome the overfitting problem more successfully than NN. Another merit of SVR is that the SVR is trained as a convex optimization problem, which can be estimated as a global solution [23]. Hence, the SVR is good at approximating any (nonlinear) functions without prior assumptions on the data property. Also, SVR has been shown to exhibit excellent performance in financial time series prediction such as stock price index [4, 23, 24], and financial volatility [5, 10, 20, 22].

There are three key parameters such as the regularized parameter C, kernel parameter δ (for RBF kernel) and loss function parameter ε, playing a crucial role in the performance of SVR. Unfortunately, it is not known beforehand which (C, δ, ε) are the best for one problem. The common way to choose these parameters is using cross-validation based on grid-search techniques. But these methods are computationally expensive because the model must be evaluated at many points within the grid for each parameter and the number of actual SVR calculations would be further multiplied by the number of cross-validation folds (5 or 10). For large models, this approach may be computationally infeasible.

To avoid this disadvantage, Genetic Algorithm (GA) has been proposed to search for appropriate SVR parameters. GA, a stochastic search based on evolutionary theory, is better than the grid-search for solving parameter optimization problems [10, 12, 19]. Hybrid SVR models trained by GA (or SVRGA) have successfully been applied in system reliability forecasting [19]; in electricity load forecasting [17, 18] and bankruptcy prediction [16]. Furthermore, Gu *et al.* [11] achieved more accurate result by using the SVRGA than by SVR combined with Grey model in the case of Chinese housing price forecasting.

Nevertheless, there are two major drawbacks on the GA. For instance, the algorithm converges slowly and the solution reaches local optimum, which are mainly caused from the population diversity reduction. When the initial population is not well designed, the GA's searches get trapped into local optimum. Therefore, evolutionary versions of the simple GA have been proposed to remove such disadvantages; for instance, the quantum GA, gene express programming and Chaotic Genetic Algorithm (CGA), to name a few. Interested researchers are referred to [13] for the promising performance of the quantum GA over the

simple GA, and [1] for the hybrid combination of ARMA model and Gene Expression Programming.

The chaotic genetic algorithm, which was integrated from GA and Chaotic Optimization Algorithm (COA), was introduced by Yuan *et al.* [26] to solve the simple GA's above-stated weakness. The CGA is a powerful tool for solving nonlinear and complex optimization problems. The basic idea of the CGA is to transform the variable of problems from the solution space to chaos' space and then perform search to find out the solution by three characteristics (randomicity, Ergodicity and regularity) of the chaotic variables. Hong *et al.* [12], combined SVR with hybrid CGA for tourism demand forecasting. The result showed that the SVRCGA model obtained by applying the CGA to search for optimal hyperparameters of SVR outperforms other competing models.

Inspired by the superior performance of the SVRCGA, in this work, we propose a new econometric model of volatility using the SVRCGA to fit conditional mean and then approximate conditional variance. The proposed hybrid SVRCGA volatility model is achieved with several steps as shown in the methodology section which include the selection of input variables by ARMA approach for fitting both mean and variance functions of returns, and the searching process of the optimal SVR hyperparameters based on the CGA while training the SVR. By using real data of NASDAQ Composite Index, we test the validity of our proposed model and we also compare this model with the volatility models by SVRGA (SVR with GA), the standard SVR using grid-search, Kernel smoothing model described in [3] and parametric models (GARCH, EGARCH, GJR and FIGARCH).

The experimental results conducted in this study showed that the predicting accuracy of this SVRCGA approach is superior to the other volatility models.

The paper is organized as follow: Section 2 presents a methodology of the hybrid volatility model. In section 3, empirical results based on real data sets are discussed. Section 4 is reserved for conclusion.

2. The Research Method

2.1. Volatility Modelling by Hybrid SVRCGA

Let stock return at time t denotes:

$$y_t = \mu_t + \sigma_t z_t \quad (1)$$

$$\sigma_t^2 = f\,[(y_{t-1} - \mu_{t-1}), ..., (y_{t-q} - \mu_{t-q}), \sigma_{t-1}^2, ..., \sigma_{t-p}^2] \quad (2)$$

$$\mu_t = g(y_{t-1}, ..., y_{t-k}) \quad (3)$$

The new volatility model is obtained by using SVRCGA to approximate these two functions f and g in equations 2 and 3, respectively. The procedure is as follow:

- *Step 1:* We run AR(k) process to y_t to obtain the optimal value of k based on Akaike's Information Criterion (AIC).
- *Step 2:* Then we make regression on y_t against (y_{t-1}, ..., $y_{t-k(optimal)}$) using SVR algorithm, whose hyperparameters are selected by the CGA, as shown in the next section in steps 1-6, to obtain an estimate $\hat{g}_{SVRCGA}$ of g. Then calculate μ_t such that:

$$\mu_t = \hat{g}_{SVRCGA}(y_{t-1}, ..., y_{t-k(optimal)}) \quad (4)$$

- *Step 3:* We run ARMA(p,q) process to s_t^2 (the empirical estimate of σ_t^2) where $s_t^2 = \frac{1}{5}\sum_{k=0}^{4} y_{t-k}^2$ as suggested in [20] to achieve the optimal values of (p, q) using the AIC.
- *Step 4:* We fit again the SVRCGA to s_t^2 with input variables obtained in step 3, say: $[(y_{t-1} - \mu_{t-1}),...,(y_{t-q(optimal)} - \mu_{t-q(optimal)}),\sigma_{t-1}^2,...,\sigma_{t-p(optimal)}^2]$ to achieve an estimate $\hat{f}_{SVRCGA}$ of f. Then calculate σ_t^2 such that:

$$\sigma_t^2 = \hat{f}_{SVRCGA}[(y_{t-1} - \mu_{t-1}), ..., (y_{t-q(optimal)} - \mu_{t-q(optimal)}), \sigma_{t-1}^2, ..., \sigma_{t-p(optimal)}^2] \quad (5)$$

The achieved model is called econometric volatility based on hybrid SVRCGA. In this volatility model, there is a relation between the conditional mean and conditional variance. Hence it refers to the idea of Merton [15] stated that risk and return should be related.

2.2. Support Vector Regression

The SVR is nonlinear kernel based approach, formulated as follow. For a given data $D = \{(x_i, z_i)\}_{i=1}^{n}$ where x_i is input vector, z_i is output, and n is total number of data. The SVR approximates a nonlinear function of the form $f(x) = (w.\varphi(x)) + b$ where w is weight, b is a constant, $\varphi(x)$ denotes a mapping function in the feature space and $(w.\varphi(x))$ describes the dot product in the feature space F. The weight vector (w) and constant (b) can be estimated by minimizing the following regularized risk function:

$$Risk = \frac{1}{n} C \sum_{i=1}^{n} L_\varepsilon(f(x_i), z_i) + \frac{1}{2}\|w\|^2 \quad (6)$$

Where $L_\varepsilon(f(x_i), z_i) = \begin{cases} |f(x) - z| - \varepsilon & if\ |f(x) - z| \geq \varepsilon \\ 0 & otherwise \end{cases}$ is ε-insensitive loss function and $\frac{1}{2}\|w\|^2$ is the regularized term which controls the trade-off between the complexity and the approximation accuracy of the regression model to ensure that the model posses an improved generalized performance; C is the regularization constant used to specify the trade-off

between the empirical risk and regularization term. Introducing two slack variables ξ_i and ξ_i^* $i = 1, ..., n$ the following problem is obtained:

$$Minimize : Risk_{reg} = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\left(\xi_i + \xi_i^*\right) \tag{7}$$

$$subject\ to \begin{cases} z_i - (w \cdot \phi(x_i)) - b \le \varepsilon + \xi_i \\ (w \cdot \phi(x_i)) + b - z_i \le \varepsilon + \xi_i^* \\ \xi_i,\ \xi_i^* \ge 0,\ for\ \ i = 1, ..., n \end{cases} \tag{8}$$

By using Lagrangian multipliers and Karush-Kuhn-Tucker conditions, the following dual problem can be formulated as:

Maximize: $La_d(\alpha, \alpha^*) = \sum_{i=1}^{n}(\alpha_i^* - \alpha_i)z_i - \varepsilon\sum_{i=1}^{n}(\alpha_i^* + \alpha_i) - \frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)K(x_i, x_j)$

Subject to $\begin{cases} \sum_{i=1}^{n}(\alpha_i^* - \alpha_i) = 0 \\ 0 \le \alpha_i \le c, i = 1, ..., n \\ 0 \le \alpha_i^* \le c\ \ i = 1, ..., n \end{cases}$

The solution of the above problem can obtained as:

$$f(x, \alpha, \alpha^*) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)K(x, x_i) + b \tag{9}$$

Where $K(x,x_i)$ is called the kernel function defined as $K(x_i, x_j) = (\varphi(x_i) \cdot \varphi(x_j))$; any kernel satisfied Mercer's condition can be used as the SVR kernel. See [6, 25] for detail.

There are several popular SVR kernel functions including Radial Basis Function (RBF), Linear, Polynomial and Sigmoid kernels. Among them, the RBF kernel defined as $K(x_i, x_j) = exp(-\delta\|x_i - x_j\|^2)$ is a reasonable first choice [7]. Therefore, for the whole work, we employ the RBF kernel function to train SVR. Three parameters such as the regularized parameter C, kernel parameter δ and loss function parameter, ε, play an important role in the performance of SVR. Thus proper selection of these parameters is needed to be considered. Next section, we will discuss about CGA used to train the SVR parameters.

2.3. Training SVR Parameters by Chaotic Genetic Algorithm

Chaotic sequence could often be represented by one-dimension base defined as equation 10:

$$\begin{aligned} t^{(i+1)} &= \tau t^{(i)}(1 - t^{(i)}) \\ t^{(i)} &\in (0,1),\ i = 0, 1, 2, ... \end{aligned} \tag{10}$$

Where $t^{(i)}$ is the value of the chaotic variable t at the i^{th} iteration, τ is the so-called bifurcation parameter of the system, $\tau \in [0,4]$. The following presents the steps:

- *Generating Initial Population by Chaotic Optimization:* For i^{th} iteration, $T_k^{(i)}$, $k = C, \delta, \varepsilon$ represent values of three parameters of SVR. Set $i = 0$:

$$t_k^{(i)} = (T_k^{(i)} - a_k)/(b_k - a_k),\ k = C, \delta, \varepsilon \tag{11}$$

This means that the three parameters are mapped via equation 11 among the intervals (a_k,b_k) into chaotic variable $t_k^{(i)}$ located in the interval (0,1). Where $a_k = Min(k)$ and $b_k = Max(k)$.

Apply equation 10 with $\tau = 4$ to compute $t_k^{(i+1)}$ and then obtain:

$$T_k^{(i+1)} = a_k + t_k^{(i+1)}(b_k - a_k) \tag{12}$$

After this transformation, the three parameters (C, δ, and ε) are encoded into a binary format, and represented by a chromosome that is composed of "genes" of binary numbers. Each chromosome has three genes, which represent three parameters. Each gene has 40bits. For instance, if each gene contains 40bits, a chromosome contains 120bits. More bits in a gene correspond to finer partition of the search space.

- *Evaluating Fitness:* Evaluate the fitness (or forecasting errors) of each chromosome. We use RMSE as the fitness function which defined as equation 13:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(a_i - f_i)^2} \tag{13}$$

Where a_i and f_i represent the actual and forecast values, and n is the number of forecasting periods.

- *Selection:* According to the fitness function, chromosomes with the highest fitness values have chance to produce offspring in the next generation. The roulette wheel selection principle is applied to choose chromosomes for reproduction.
- *Crossover:* In crossovers, chromosomes are paired randomly. The single-point-crossover principle is employed herein. Segments of paired chromosomes between two determined break-points are swapped. Finally, decode the crossover three parameters in a decimal format.
- *Annealing Chaotic Mutation:* For the i^{th} iteration:

$$\hat{t}_k^{(i)} = \frac{(\hat{T}_k^{(i)} - a_k)}{(b_k - a_k)},\quad k = C,\ and\ i = 1, 2, ..., I \tag{14}$$

$$\tilde{t}_k^{(i)} = \hat{t}_k^{(i)} + \vartheta t_k^{(i)} \tag{15}$$

Where $\hat{T}_k^{(i)}$ and $\hat{t}_k^{(i)}$ denote i^{th} iteration crossover population and crossover chaotic variable respectively. $\tilde{t}_k^{(i)}$ is chaotic mutation variable, I is maximum evolutional generation of population and ϑ the annealing operation.

Finally, the chaotic mutation variable obtained in interval [0, 1] is mapped to the solution interval (a_k,b_k) by definite probability of mutation (p_k) and completes a mutative operation:

$$\tilde{T}_k^{(i)} = a_k + \tilde{t}_k^{(i)}(b_k - a_k) \quad (16)$$

- *Stop Criterion:* If the number of generation is equal to a given scale, then the best chromosomes are presented as a solution, otherwise go back to step 2.

3. The Data

We use real data set of NASDAQ Composite Index. The data are taken for the period from January 2001 to December 2010. The continuously compounded daily returns at time t are calculated as $y_t = 100\ (\ln P_t - \ln P_{t-1})$, where P_t is the index price at time t. Before modelling volatility, the whole data are divided into three subsets: training data (2001-2006), validation data (2007-2008) and testing data (2009-2010) sets.

3.1. Setting Parameters in CGA

We set the population sizes ($p_{size} = 200$), the maximum evolution generations of the population ($I = 500$), the probability of crossover ($p_c = 0.5$), the probability of mutation ($p_m = 0.5$) and the annealing operation parameter ($\vartheta = 0.9$).

Training and forecasting approach follows a rolling window scheme to implement and evaluate the model based volatility. Let $T = R + P$, where R denotes the rolling window, P denotes the rolling time and T is the sample observations. The rolling window scheme works in the following manner:

At first, we use observations from 1 to R fed into SVRCGA model, and the structural risk minimization principle is employed to minimize the training error, then obtain one-step ahead forecasting point. In second step, data from 2 to $R + 1$ are used in the model and one-step ahead forecasting point is also obtained, and finally from R to $R + P - 1$. In this way, the rolling window R stays fixed and the forecasts do not overlap. Meanwhile, training error in this training stage is also obtained. While training errors improvement occurs, the three kernel parameters, (C, δ, ε) of SVRCGA model adjusted by CGA are employed to calculate the validation error. Then, the adjusted parameters with minimum validation error are selected as the most appropriate parameters. Finally, the obtained models are applied to forecast the test data. Note that the testing datasets are not used for modelling but for examining the accuracy of the forecasting models. The optimal parameters for SVRCGA and SVRGA models are shown in Tables 1 and 2, respectively. The experimental results of ARMA-GARCH, ARMA-EGARCH, ARMA-GJR and AFIMA-FIGARCH and Smoothing Kernels models are not shown and available upon request. The forecasting results are illustrated in Table 3.

Table 1. Optimal parameters of SVR tuned by CGA.

Optimal Parameters	Mean Equation	Volatility Equation
δ	0.539	3.741
C	5.224×10^3	2.062×10^5
ε	0.38	0.62
Number of Inputs	k=3	p = 2, q = 1
Smallest RMSE	3.783	3.551

Table 2. Optimal parameters of SVR tuned by GA.

Optimal Parameters	Mean Equation	Volatility Equation
δ	1.192	2.044
C	8.052×10^3	4.268×10^5
ε	0.21	0.81
Number of Inputs	k=3	p = 1, q = 2
Smallest RMSE	3.896	3.775

Table 3. Forecasting performance for test data.

	RMSE	NMSE	QLIKE
SVRCGA	3.8076	0.6304	1.2517
SVRGA	3.8461	0.6432	1.2628
SVR	3.8909	0.6583	1.3153
Kernel Smoothing	3.9006	0.6610	1.3255
GARCH	4.2140	0.7722	1.5096
EGARCH	3.9054	0.6632	1.3276
GJR	4.2363	0.7804	1.4851
FIGARCH	4.1544	0.7505	1.3477

3.2. Forecasting Results

We use three different forecasting criteria: RMSE, NMSE and QLIKE to check the predicting accuracy of our proposed model:

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y_t^2 - \hat{\sigma}_t^2)^2}\ ,\ NMSE = \frac{1}{v^2 T}\sum_{t=1}^{T}(y_t^2 - \hat{\sigma}_t^2)^2\ ,\ \text{where}$$

$$v^2 = \frac{1}{T-1}\sum_{t=1}^{T}(y_t^2 - \bar{y}_t^2)^2,\quad QLIKE = \frac{1}{T}\sum_{t=1}^{T}(\ln(\hat{\sigma}_t^2) + y_t^2/\hat{\sigma}_t^2),$$

where squared return y_t^2 is treated as actual volatility and $\hat{\sigma}_t^2$ is volatility forecasted by each model.

The experiment results showed that SVRCGA outperforms the other models for all cases as it can be seen from the Table 3 that the SVRCGA generated smallest values of RMSE, NMSE and QLIKE among all models in this study. Figure 1 shows the plots of volatility forecasts by the competing models against actual volatility. Graphical representation of the forecasts by SVRCGA exhibit more flexible and yield higher prediction performance than the parametric models. Therefore, it is feasible and effective to apply the SVRCGA for volatility prediction.

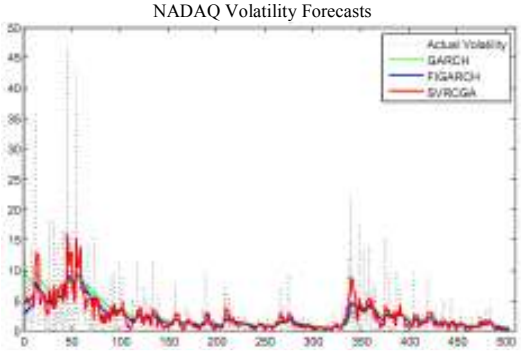

Figure 1. Plot of out-of-sample volatility forecasts.

4. Conclusions

This study proposed a new volatility model using hybrid SVRCGA algorithms to fit mean function and conditional variance of returns. Several conclusions can be made from this work. First, the hybrid volatility models (SVRCGA, SVRGA, and SVR) are valid for forecasting complex stock market volatility and they generate better forecasting results than GARCH, two asymmetric volatility models (EGARCH and GJR) and long memory model, FIGARCH. Second, the Chaotic nonlinear algorithm of CGA is an excellent tool for searching optimal parameters of SVR and it performs better than the standard GA and grid-search technique. Third, as SVR algorithm has a high-powered ability to approximate any (nonlinear) functions without prior assumptions on the underlying data generating process, the combination of SVR and CGA makes the hybrid SVRCGA as the most efficient tool for modelling and predicting the stylized characteristics of complex stock market returns and volatility. Finally, the proposed model is recommended for volatility stock market researchers.

Acknowledgements

We like to thank the reviewers for their helpful comments. We are responsible for all errors. This research is supported by Shanghai Leading Academic Discipline Project, Project Number: S30504.

References

[1] Barbulescu A. and Bautu E., "A Hybrid Approach for Modeling Financial Time Series," *the International Arab Journal of Information Technology*, vol. 9, no. 4, pp. 327-335, 2012.

[2] Bildirici M. and Ersin O., "Improving Forecasts of GARCH Family Models with the Artificial Neural Networks: An Application to the Daily Returns in Istanbul Stock Exchange," *Expert Systems with Applications*, vol. 36, no. 4, pp. 7355-7362, 2009.

[3] Bühlmann P. and McNeil A., "An Algorithm for Nonparametric GARCH Modelling," *Computational Statistics & Data Analysis*, vol. 40, no. 4, pp. 665-683, 2002.

[4] Cao L. and Tay F., "Financial Forecasting Using Support Vector Machines," *Neural Computing & Applications*, vol. 10, no. 2, pp. 184 -192, 2001.

[5] Chen S., Jeong K., and Hardle W., "Forecasting Volatility with Support Vector Machine-Based GARCH Model," *Journal of Forecasting*, vol. 29, no. 4, pp. 406-433, 2010.

[6] Cristianini N. and Shawe-Taylor J., *An Introduction to Support Vector Machines*, Cambridge University Press, Cambridge, 2000.

[7] Dibike Y., Velickov S., Solomatine D., and Abbott M., "Model Induction with Support Vector Machines: Introduction and Application," *Journal of Computing in Civil Engineering*, vol. 15, no. 3, pp. 208-216, 2001.

[8] Donaldson R. and Kamstra M., "An Artificial Neural Network-GARCH Model for International Stock Return Volatility," *Journal of Empirical Finance*, vol. 4, no. 1, pp. 17-46, 1997.

[9] Dunis C. and Huang X., "Forecasting and Trading Currency Volatility: An Application of Recurrent Neural Regression and Model Combination," *Journal of Forecasting*, vol. 21, no. 5, pp. 317-354, 2002.

[10] Gavrishchaka V. and Banerjee S., "Support Vector Machine as an Efficient Framework for Stock Market Volatility Forecasting," *Computational Management Science*, vol. 3, no. 2, pp. 147-160, 2006.

[11] Gu J., Zhu M., and Jiang L., "Housing Price Forecasting Based on Genetic Algorithm and Support Vector Machine," *Expert Systems with Applications*, vol. 38, no. 4, pp. 3383-3386, 2011.

[12] Hong W., Dong Y., Chen L., and Wei S., "SVR with Hybrid Chaotic Genetic Algorithms for Tourism Demand Forecasting," *Applied Soft Computing*, vol. 11, no. 2, pp. 1881-1890, 2011.

[13] Laboudi Z. and Chikhi S., "Comparison of Genetic Algorithm and Quantum Genetic Algorithm," *the International Arab Journal of Information Technology*, vol. 9, no. 3, pp. 243-249, 2012.

[14] Maulik U. and Bandyopadhyay S., "Genetic Algorithm-Based Clustering Technique," *Pattern Recognition*, vol. 33, no. 9, pp. 1455-1465, 2000.

[15] Merton R., "An Intertemporal Capital Asset Pricing Model," *Econometrica*, vol. 41, no. 5, pp. 867-887, 1973.

[16] Min S., Lee J., and Han I., "Hybrid Genetic Algorithms and Support Vector Machines for Bankruptcy Prediction," *Expert Systems with Applications*, vol. 31, no. 3, pp. 652-660, 2006.

[17] Pai P. and Hong W., "Support Vector Machines with Simulated Annealing Algorithms in Electricity Load Forecasting," *Energy Conversion and Management*, vol. 46, no. 17, pp. 2669-2688, 2005.

[18] Pai P. and Hong W., "Forecasting Regional Electric Load Based on Recurrent Support Vector Machines with Genetic Algorithms," *Electric Power Systems Research*, vol. 74, no. 3, pp. 417-25, 2005.

[19] Pai P., "System Reliability Forecasting by Support Vector Machines with Genetic

Algorithms," *Mathematical and Computer Modelling*, vol. 43, no. 3-4, pp. 262-274, 2006.
[20] Perez-Cruz F., Afonso-Rodriguez J., and Giner J., "Estimating GARCH Models Using Support Vector Machines," *Quantitative Finance*, vol. 3, no. 3, pp. 163-172, 2003.
[21] Rubenstein-Montano B., Anandalingam G., and Zandi I., "A Genetic Algorithm Approach to Policy Design for Consequence Minimization," *European Journal of Operational Research*, vol. 124, no. 1, pp. 43-54, 2000.
[22] Tang L., Sheng H., and Tang L., "Forecasting Volatility Based on Wavelet Support Vector Machine," *Expert Systems with Applications*, vol. 36, no. 2, pp. 2901-2909, 2009.
[23] Tay F. and Cao L., "Application of Support Vector Machines in Financial Time Series Forecasting," *Omega*, vol. 29, no. 4, pp. 309-317, 2001.
[24] Tay F. and Cao L., "Modified Support Vector Machines in Financial Time Series Forecasting," *Neurocomputing*, vol. 48, no. 1-4, pp. 847-861, 2002.
[25] Vapnik V., *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, USA, 1995.
[26] Yuan X., Yuan Y., and Zhang Y., "A Hybrid Chaotic Genetic Algorithm for Short-Term Hydro System Scheduling," *Mathematics and Computer in Simulation*, vol. 59, no. 4, pp. 319-327, 2002.

Phichhang Ou is currently working as a lecturer at Pannasastra University of Cambodia and Build Bright University. He got his PhD in management science from the Business School, the University of Shanghai for Science and Technology, Shanghai, China, since May 2012. His main interests of research include: Statistical data mining techniques (neural network, support vector machines, genetic algorithm) and financial time series analysis.

Hengshan Wang is a professor of management information system at business school, the University of Shanghai for Science and Technology (USST). He was a former Dean of Business School, USST and his research interests include: Data mining techniques, management information system, and complex networks analysis.