

Application of an integrated support vector regression method in prediction of financial returns

Yuchen Fu*

Soochow University /School of Computer Science & Technology, Suzhou, China
Email: yuchenfu@suda.edu.cn

Yuanhu Cheng

Soochow University /School of Computer Science & Technology, Suzhou, China
Email: lidia421cc@163.com

Abstract—Nowadays there are lots of novel forecasting approaches to improve the forecasting accuracy in the financial markets. Support Vector Machine (SVM) as a modern statistical tool has been successfully used to solve nonlinear regression and time series problem. Unlike most conventional neural network models which are based on the empirical risk minimization principle, SVM applies the structural risk minimization principle to minimize an upper bound of the generalization error rather than minimizing the training error. To build an effective SVM model, SVM parameters must be set carefully. This study proposes a novel approach, support vector machine method combined with genetic algorithm (GA) for feature selection and chaotic particle swarm optimization(CPSO) for parameter optimization support vector Regression(SVR),to predict financial returns. The advantage of the GA-CPSO-SVR (Support Vector Regression) is that it can deal with feature selection and SVM parameter optimization simultaneously. A numerical example is employed to compare the performance of the proposed model. Experiment results show that the proposed model outperforms the other approaches in forecasting financial returns.

Index Terms—SVR; GA-CPSO; financial returns; forecasting

I. INTRODUCTION

The ability of accurately predicting financial returns is invaluable for financial institutions. Reliability predictions are used for various purposes, such as reliability assessment, evaluating risks and liabilities, evaluating replacement policies etc. Thus numerous models have been depicted to provide the investors with more precise predictions. Unlike traditional statistical models, SVM is data-driven, non-parametric model and let "the data speak for themselves". Recently, with the introduction of Vapnik's ϵ -insensitive loss function, SVM has been extended to solve nonlinear regression estimation problems and has been used successfully to solve forecasting problem in various fields. At present, SVR has been used in predicted areas widely. It's not easy to judge whether financial returns is purely linear or nonlinear due to the complexity of financial returns. Therefore, many empirical results have indicated that combination of different is able to increase the forecasting accuracy as well as improve the robustness of forecasting models. In this article, the linear and

nonlinear SVM models are combined to capture the data pattern of financial returns.

Several studies have proved that the SVM is capable to correctly predict, but there remains two problems for the SVM, namely feature selection and optimal SVM parameters setting, which influence each other, that means that they should be dealt with at the same time.

Therefore, an integrated Support Vector Regression model is built to overcome the problem. In the new model feature selection and SVM parameter optimization are considered simultaneously.

Genetic algorithm(GA) is used for the feature selection and chaotic particle swarm optimization algorithm(CPSO) is used for parameter optimization. A series of experiments show that the proposed GA-CPSO-SVR possesses better ability of finding the global optimum compared with other methods. More detail of the proposed model is presented in Section II.

II. A INTEGRATED SUPPORT VECTOR REGRESSION (GA-CPSO-SVR)

The Support Vector Machine (SVM) developed at AT&T Bell Laboratories by Vapnik and co-works, as a new machine learning method, has been very successful in pattern classification and regression problems for crisp data^[3].

A. SUPPORT VECTOR REGRESSION(SVR)

The basic concept of the SVR is to map nonlinearly the original data x into a higher dimensional feature space. Hence, suppose we are given a training data set $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \subset N * R$, where N denotes the space of input patterns—for instance, R_k .

In ϵ -SVM Regression, the goal is to find a regression function $f(x)$:that has at most ϵ deviation from the actually obtained targets y_i for all the training data. In other words, we do not care about error as long as they are less than ϵ , but will not accept any deviation larger than ϵ . An ϵ -insensitive loss function

$$L_{\epsilon}(x, y, f(x)) = \begin{cases} |y - f(x)| - \epsilon & \text{if } |y - f(x)| > \epsilon \\ 0 & \text{else} \end{cases} \quad (1)$$

so the error is penalized only if it is outside the ϵ -tube. Fig.1 depicts this situation graphically. To make the SVM regression nonlinear, this could be achieved by

simply mapping the training patterns x_i by a nonlinear transform $\phi: N \rightarrow F$ into some high dimensional feature space F . A simple example of the nonlinear transform ϕ is the polynomial transform function:

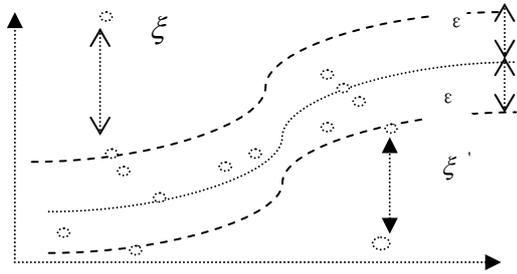


Figure 1. The epsilon insensitive loss setting corresponds for a linear Support Vector regression machine

$$x = (x_1, x_2) \rightarrow \phi(x) = (\sqrt{3}x_1, \sqrt{3}x_2, \sqrt{3}x_1^2, \sqrt{3}x_2^2, \sqrt{6}x_1x_2, \sqrt{3}x_1^2x_2, \sqrt{3}x_1x_2^2, x_1^3, x_2^3) \quad (2)$$

Where the degree of the polynomial transform function is 3, and dimension is input space N and feature space F is 2 and 9, respectively. A best fitting function

$$f(x) = \sum_{i=1}^N w_i \phi_i(x) + b = w^T \phi(x) + b \quad (3)$$

where

$\phi_i(x)$ is the feature of inputs x ; both w_i and b are coefficients which are estimated by minimizing the regularized risk function:

To avoid over-fitting in the very high-dimension feature space, one should add a capacity control term, which in the SVM case results to be $\|w\|^2$. Formally, the SVM regression model can be written as a convex optimization problem by requiring:

$$\epsilon\text{-tube Minimize} : R(w, \zeta, \zeta^*) = \frac{1}{2} \|w\|^2 + C \left(\sum_{i=1}^N (\zeta_i + \zeta_i^*) \right) \quad (4)$$

With the constraints:

$$\begin{aligned} w_i \phi(x_i) + b - d_i &\leq \epsilon + \zeta_i^*, i = 1, 2, \dots, N, \\ -w_i \phi(x_i) - b + d_i &\leq \epsilon + \zeta_i, i = 1, 2, \dots, N, \\ \zeta_i, \zeta_i^* &\geq 0, i = 1, 2, \dots, N \end{aligned}$$

The constant $C > 0$ determines the trade off between the complexity of $f(x)$ and the amount up to which deviations larger than ϵ are tolerated. We can estimate a linear function in the feature space that makes SVM regression attractive. In addition, the regression task was achieved by solving a convex programming with linear constraints that means it has a unique solution. The size of ϵ -insensitive is a pre-defined constant. The modeling performance may seriously affected by the selection of a parameter. The ϵ -insensitive zone represented as a tube shape in the SVM regression, only the training data outside the ϵ -tube will be penalized. In many real-world applications, the effects of the training points are different. We would require that the precise training points must be regressed correctly, and would allow more errors on imprecise training points.

The constrained optimization problem in (4) is solved using the following primal Lagrangian form:

$$Min \quad f = MAPE_{cross \ validation} = \frac{\sum_{i=1}^l |a_i - p_i|}{l} * 100\% \quad (5)$$

Karush-Kuhn-Tucker (KKT) conditions are applied to the regression, and (4) thus yields the dual Lagrangian,

$$J(a_i, a_i^*) = \sum_{i=1}^N d_i (a_i - a_i^*) - \epsilon \sum_{i=1}^N (a_i + a_i^*) - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (a_j - a_j^*) K(x, x_j) \quad (6)$$

s.t.

$$\sum_{i=1}^N (a_i - a_i^*) = 0, \quad 0 \leq a_i, a_i^* \leq C, \quad i = 1, 2, \dots, N.$$

Both α_i and α_i^* are called Lagrangian multipliers that satisfy the equalities $\alpha_i \alpha_i^* = 0$.

Hence an optimal desired weights vector of the regression hyper-plane is represented as

$$w^* = \sum_{i=1}^N (\alpha_i - \alpha_i^*) K(x_i, x) \quad (7)$$

Therefore, the regression function could be

$$f(x, \alpha, \alpha^*) = \sum_{i=1}^N (\alpha_i - \alpha_i^*) K(x_i, x_j) + b \quad (8)$$

Here, $K(x_i, x_j)$ is called the kernel function. The value of the kernel equals the inner product of two vectors x_i and x_j in the feature $\phi(x_i)$ and $\phi(x_j)$; that is $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$. Any function that satisfies Mercer's Condition by Vapnik can be called the Kernel function.

There are several frequently-used kernel functions, all of them have their own advantages and disadvantages.

In this work, the linear kernel function, represented as (9), is employed for linear SVM model.

$$K(x_i, x_j) = x_i^T x_j \quad (9)$$

In general, a mixed kernel function is used for nonlinear SVM model. With the purpose of better learning ability and better generalization ability. In order to here a global kernel and a local kernel are combined to achieve a mixed kernel function. For instance, we usual use addition, multiplication to combine two or more kernels. For instance, there are two kernels k_1 and k_2 that we can possible define a kernel $k = k_1 + k_2$. Given a set of kernel $k = \{k_1, k_2, \dots, K_m\}$, in this article, the linear combination can be represented as:

$$k = \sum_{i=1}^m u_i k_i \quad (10)$$

For example, the Gaussian function and the linear function are mixed with a specified weight m , shown as (11)

$$K(x_i, x_j) = m \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) + (1 - m)x_i^T x_j \quad (11)$$

The selection of the three parameters (σ, ϵ, c) of a SVM model is important to the accuracy of forecasting. Notably, there are only two parameters, ϵ and c , for the linear SVM model. However, structural methods for confirming efficiently the selection of parameters efficiently are lacking. Therefore, CPSO is used in the proposed SVM model to optimize parameter selection.

B. GENETIC ALGORITHM FOR THE FEATURE SELECTION

Genetic algorithm(GA), proposed by Holland, which is an organized random search technique and which imitates the biological evolution process. The algorithms try to retain genetic information from generation to generation that are based on the principle of the survival of the fittest. GA has been successfully applied to a series of problems such as data mining and optimization. It has also been used for the feature selection in SVM modeling^[4].

As to a specific problem, The GA looks a solution as an individual chromosome. It defines an initial population of these individuals, which represent the solution space of the problem. First a set of chromosomes is randomly chosen from the search space to form the initial population. Second the individuals are selected in a competitive manner, based on their fitness as measured by a specific objective function. The chromosome are binary-encoded, each bit of the chromosome represents a gene.

In our implementation of GA, a binary string with each bit representing a feature('1' means that this gene is kept in the gene subset and '0' represents the gene is not included in the subset)was used to represent a chromosome. The GA is then applied to a population of randomly generated binary strings. The fitness of each string is determined as follows:

$$fitness = W_A * SVM_accuracy + W_F * N_F \quad (12)$$

Where W_A is the SVM classification accuracy weight, N_F is the number of feature selected, W_F is the weight of feature number. The accuracy of 10-fold cross validation was used as *SVM_accuracy*.

W_A and W_F both can be adjusted based on their relative importance.

The Genetic algorithm for the feature selection is presented as follows.

- Step 1: (Initialization): In this study, the initial population was composed of 50 randomly created chromosomes. The population size of 50 was selected as a trade-off between the convergence time and the population diversity.
- Step 2: (Fitness evaluation): Evaluate the fitness of each chromosome. The fitness value for each chromosome was calculated according to

$$Min \quad f = MAPE_{cross \ validation} = \frac{\sum_{i=1}^l |a_i - p_i|}{l} * 100\% \quad (13)$$

Where l is the number of training data samples; a_i is the actual value, and p_i is the predicted value. The solution with a smaller *MAPE_{cross validation}* of the training data set has a smaller fitness value, and thus has a better chance of surviving in the successive generations.

- Step 3: Selection: A standard roulette wheel was employed to select 10 chromosomes from the current population
- Step 4: Crossover. The simulated binary crossover (Deb & Agrawal), 1995; Deb & Kumar, 1995)

was applied to randomly paired chromosomes. The probability of creating new chromosomes in each pair was set to 0.7. The newly created chromosomes constituted a new population.

- Step 5: Mutation: The mutation operation follows the crossover operation and determines whether a chromosome should be mutated in the next generation. This study applied polynomial mutation methods to the proposed model. Each chromosome in the new population was subject to mutation with a probability of 0.08.
- Step 6: Elitist strategy: The fitness value was calculated for the chromosomes of the new population. If the minimum fitness value of the new population is smaller than that of the old population, then the old chromosome can be replaced with the new chromosome of the minimum fitness value.
- Step 7 : Stopping criteria. The process was repeated from 2 to 6 until the number of generations was equal to 60. If the number of generations equals a threshold, then the best chromosomes are presented as a solution; otherwise go back to Step2.

An overview of GA for the feature selection will be given in the next subsequent paragraphs.

The setting probability of parameter is largely effect the converged solution. In this study, the crossover probability is recommended from Holland. The choices of other parameters such as the mutation--probability, population seize are based on numerous experiments as those values provide the smallest *MAPE_{crossvalidation}* on the training data set.

C. CPSO ALGORITHMS FOR SELECTING PARAMETER OF SVR MODEL

In SVR, parameters inappropriately chosen result in over-fitting or under-fitting. Generally, when selecting the parameters, most researchers still follow standard procedure (trial-and-error).First building a few SVR models based on different parameter sets, then testing them on validation set to obtain optimal parameters. However, this procedure is time-consuming and requires some luck.

Particle swarm optimization(PSO), for easy implementation and quick convergence, has gained much attention and wide applications in solving continuous nonlinear optimization problem. Particle swarm optimization system begins with the random initialization of a population and multiple candidate solutions in the problem search space. Each particle looks for the optimal position to land. Finally the best global position can be attained by adjusting the direction of each particle towards its own best location at each generation. Adjust the velocity of each particle to improve the direction of each particle,tracking and memorizing the best position encountered could cumulate each particle's experience. So each particle remembers the best position where it reached in the past, then the PSO system combines local search method with global search methods.

In a SVR model, the velocity, the position, the best position of each particle, due to the three parameters, that can be represented as (14)-(16)

$$x_{k(i)} = [x_{(k)i,1}, x_{(k)i,2}, \dots, x_{(k)i,n}], \quad (14)$$

$$v_{k(i)} = [v_{(k)i,1}, v_{(k)i,2}, \dots, v_{(k)i,n}], \quad (15)$$

$$p_{k(i)} = [p_{(k)i,1}, p_{(k)i,2}, \dots, p_{(k)i,n}], \quad (16)$$

$$k = c, \varepsilon, \sigma, \quad i = 1, 2, \dots, N$$

Thus the global best position among all particles in the swarm $x_k = [x_{(k)1}, x_{(k)2}, \dots, x_{(k)N}]$ is computed by (12)

$$P_{(k)global} = [P_{(k)g,1}, P_{(k)g,2}, \dots, P_{(k)g,d}] \quad (17)$$

The new position of each particle in next generation can be shown as (18)

$$\begin{aligned} V_{(k)i}(t+1) &= lV_{(k)i}(t) + m_1 \text{rand}(1)(P_{(k)i} - x_{(k)i}(t)) \\ &+ m_2 \text{rand}(2)(P_{(k)g} - x_{(k)i}(t)) \end{aligned} \quad (18)$$

$$k = c, \varepsilon, \sigma, \quad i = 1, 2, \dots, N$$

Where l is called the inertia weight that controls the impact of the previous velocity of the particle on its current one, m_1 and m_2 are called acceleration coefficients.

After the velocity has been updated, the new position of the particle for each parameter in the next generation is shown as (19)

$$\begin{aligned} X_{(k)i}(t+1) &= V_{(k)i}(t+1) + X_{(k)i}(t) \\ X &= [X_1, X_2, \dots, X_N] \end{aligned} \quad (19)$$

Where l is called the inertia weight that controls the impact of the previous velocity of the particle on its current one; m_1 and m_2 are two positive constants called acceleration coefficients; $\text{rand}(1)$ and $\text{rand}(2)$ are two independently uniformly distributed random variables with range $[0,1]$, $V_{(k)i} = [V_{(k)i,1}, V_{(k)i,2}, \dots, V_{(k)i,n}]$ represents the distance to be traveled by the particle from its current position; $X = [X_1, X_2, \dots, X_N]$ means the position of particle l , and $p_{best(i)}$ the best previous position of particle l ; $p_{gbest(i)}$ represents the best position among all particles in the population.

The performance depends on its parameters in the procedure of PSO, it often lead to be trapped in local optimum. Thus proper control of l is very important to find the optimum solution accurately.

To overcome the problem, a chaotic PSO (CPSO) method that combined PSO with AIWF (adaptive inertia weight factor) and chaotic local search (CLS) is employed to overcome the shortcoming. CPSO is a two phase iterative strategy based on the proposed PSO with AIWF and CLS, in which AIWF is used to encourage good particles to improve the exploration to refine results by local search, and bad ones to modify searching space with large step. AIWF is determined as (20)

$$l = \begin{cases} l_{\min} + \frac{(l_{\max} - l_{\min})(f_i - f_{\min})}{f_{\text{avg}} - f_{\min}} & \text{if } f_i \leq f_{\text{avg}} \\ l_{\max} & \text{else} \end{cases} \quad (20)$$

Where l_{\max} and l_{\min} are the maximum and minimum of l , respectively, f_i is the current objective value of the particle, f_{avg} and f_{\min} are the average and minimum objective values of all particle, respectively.

CLS is used to perform locally oriented search for the solution f_{global} , which is resulted from PSO. CLS is based on the logistic equation, with sensitive dependence on initial conditions, and is defined by the following equation:

$$cx_{(k)i}^{(\lambda+1)} = 4cx_{(k)i}^{(\lambda)}(1 - cx_{(k)i}^{(\lambda)}) \quad (21)$$

Where $cx_{(k)i}$ is the i_{th} chaotic variable; λ represents the iteration number, $cx_{(k)i}^{(\lambda)}$ is distributed in the range $(0,1)$ and $cx_{(k)i}^{(0)} \in (0,1)$ but $cx_{(k)i}^{(0)} \notin \{0.25, 0.5, 0.75\}$.

The procedure of CLS is illustrated is presented as follows:

- Step 1: Setting $\lambda = 0$ and map the three parameters into chaotic variable

$$cx_{(k)i}^{\lambda} = \frac{x_{(k)i}^{\lambda} - x_{\min(k)i}}{x_{\max(k)i} - x_{\min(k)i}} \quad (22)$$

- Step 2: Compute the next iteration chaotic variable $cx_{(k)i}^{\lambda(i+1)}$ with (16).
- Step 3: Obtain three parameters for the next iteration by the 17 and compute the new objective value

$$x_{(k)i}^{\lambda(i+1)} = x_{\min(k)i} + cx_{(k)i}^{\lambda(i+1)}(x_{\max(k)i} - x_{\min(k)i}) \quad (23)$$

- Step 4: If the new objective value with smaller forecasting accuracy index value or maximum iteration is reached, then, the new chaotic variable $x_{(k)i}^{\lambda(i+1)}$ and its corresponding objective value is the final solution; otherwise, let $\lambda = \lambda + 1$ and go back to Step 2.

In the investigation, the normalized mean squared error (NMSE), shown as (22), serves as the forecasting accuracy index for identifying suitable parameters, determined in Step 4 of PSO and in Step 5 of CLS, for use in the GA-CPSO-SVR model.

$$NMSE = -\frac{1}{n\delta^2} \sum_{i=1}^n (a_i - f_i)^2 \quad (24)$$

$$\text{Where } \delta^2 = \frac{1}{n} \sum_{i=1}^n (a_i - \bar{a})^2 \text{ and}$$

n is the number of forecasting periods;

a_i is the actual value at period i ;

f_i denotes is the forecasting value at period i .

An overview of CPSO parameter settings will be given in the subsequent paragraphs.

Fig. 2 shows the framework of the proposed GA-CPSO-SVR model.

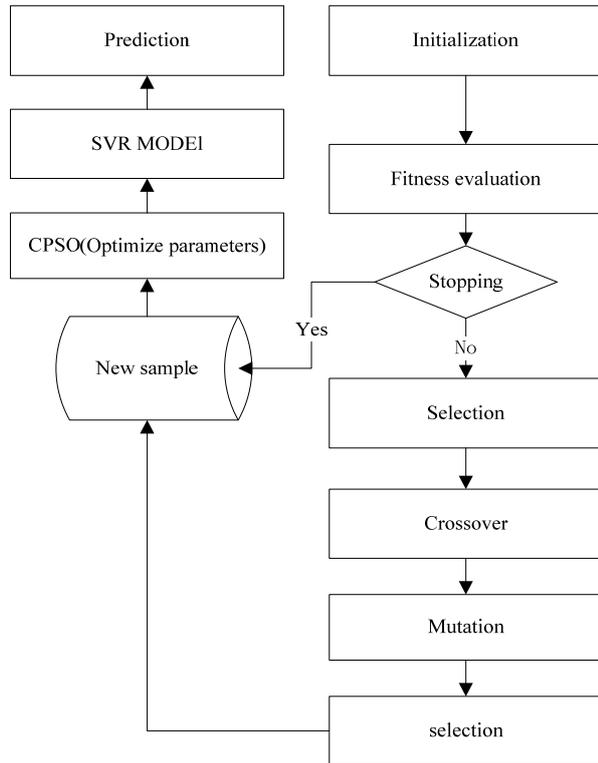


Figure 2. The architecture of a GA-CPSO-SVR model

D THE PROPOSED SVRCPSO MODEL

A strategy which has both linear and nonlinear modeling

abilities is a good alternative for forecasting financial returns for the complex data pattern of financial returns. In this article, the linear GA-CPSO-SVR model is used as a preprocessor to capture the linear data pattern, while the nonlinear GA-CPSO-SVR model is applied right after the linear GA-CPSO-SVR to forecast the nonlinear data pattern of residuals from the linear GA-CPSO-SVR model. The hybrid model (H_i) can be represented as :

$$H_i = L_i + N_i \tag{25}$$

Where L_i is the linear part; and N_i represents the nonlinear part of the hybrid model. Both of them are estimated from the data set, L_i^* is the forecast value of the linear GA-CPSO-SVR model at time i . Let ε_i represent the residual at time i from the linear GA-CPSO-SVR model; then,

$$\varepsilon_i = H_i - L_i^* \tag{26}$$

The residuals are modeled by the nonlinear GA-CPSO-SVR and can be represented as follows:

$$\varepsilon_i = f(\varepsilon_{i-1}, \varepsilon_{i-2}, \dots, \varepsilon_{i-n}) + \Delta_i \tag{27}$$

Where f represents the nonlinear model and Δ_i represents the random error. Thus the combined forecast is

$$L_i^* + N_i^* = H_i^* \tag{28}$$

Notably, N_i^* is the forecast value of (24). Fig. 3 shows the structure of the GA-CPSO-SVR model proposed in this article.

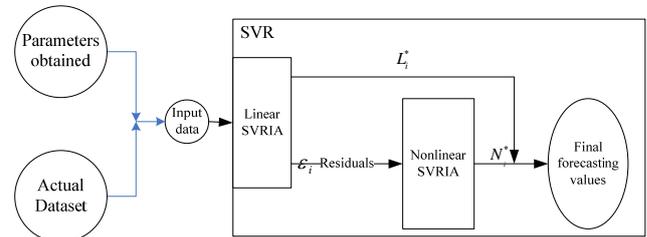


Figure 3. The architecture of proposed GA-SVR-CPSO model for prediction

III A NUMERICAL EXAMPLE

In order to verify the accuracy of the algorithm we proposed in forecasting financial returns. We predict financial returns of the Shanghai Composite Index. The samples of this paper were collected from Shanghai Stock Exchange from February 5, 2006 to January 28, 2007, a total of 358 observations. The samples were ordered from 1 to 358, and we divided them into 3 groups: The training group (from 1 to 150), the validation group (from 151 to 263), the testing group (264-358)

After the CPSO was applied to search for the optimal parameter sets, the optimal parameters in the Linear GA-CPSO-SVR model. where $c=2.639$, $\varepsilon =13.928$, and $\sigma^2 =2.365$, as shown in Fig. 4.

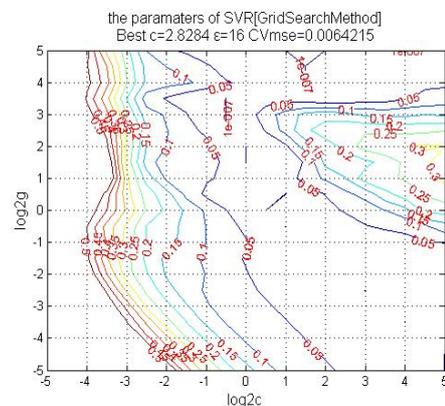


Figure 4. The optimal parameters in the Linear GA-CPSO-SVR model

The parameters of the Nonlinear GA-CPSO-SVR model as Fig. 5 shows

We can see $c=2.8284$, $\varepsilon =16$ in the Nonlinear GA-CPSO-SVR model.

In order to reduce the differences among the different data fractional error. Sample need to be normalized when we have reconstructed the sample. AS the sample which used in this study are all positive. So we shrink them to $[0,1]$ range, data normalization formula is:

$$X_i' = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}} \tag{29}$$

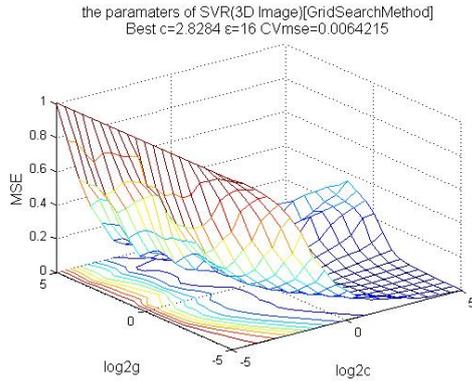


Figure 5. The parameters of the Nonlinear GA-CPSO-SVR model

The results of data after normalization as shown in Fig. 6:

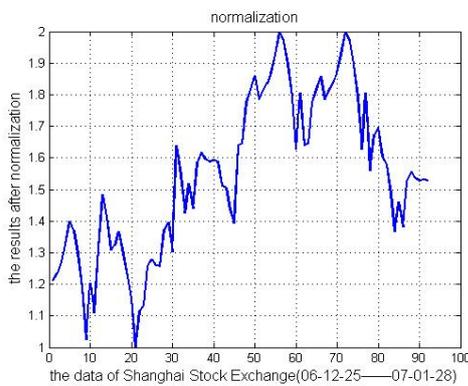


Figure 6. Data of Shanghai Stock Exchange after normalization

Then the final SVR forecasting models were built. The forecasting simulation was performed against the testing data. Fig. 7 makes comparison of actual values and forecasted values by the proposed model.

In the Fig. 8, we can see the fractional error of actual values and forecasted values with the model we built.

Compare the results obtained with the forecasting results from the Back Propagation Neural Network model (BPNN) and the Auto Regressive Integrated Moving Average model (ARIMA) and Support Vector Regression and CPSO-SVR. Here we use the criterion MAPE (the smallest value of MAPE) to illustrate the accuracy index.

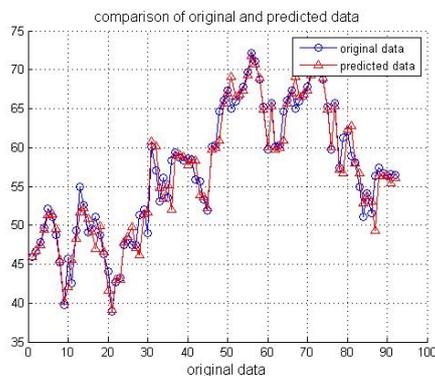


Figure 7. Comparison of actual values and forecasted values by GA-CPSO-SVR model

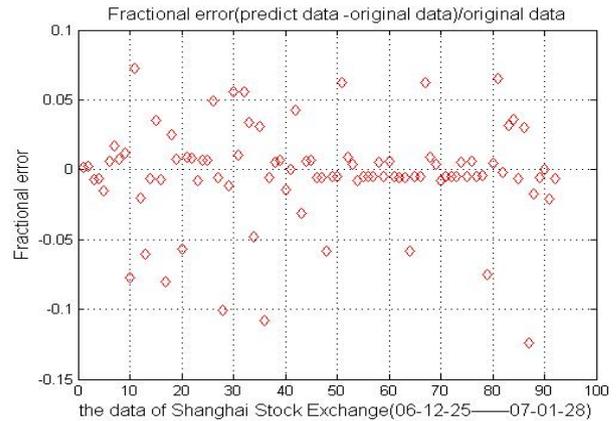


Figure 8. Fractional error of actual values and forecasted values by GA-CPSO-SVR model

TABLE I. PARAMETERS OF HSVRCPSO MODEL

Model	MAPE(%)
GA-CPSO-SVR	3.12
CPSO-SVR	3.69
SVR	4.12
BPNN	4.98
ARIMA	5.64

IV CONCLUSIONS

The motivation of this study is based on the evidence that different forecasting models can complement each other in approximation data sets. This paper applied SVM model composed of linear and nonlinear SVR to the forecasting field of financial returns. Then Genetic algorithm is used for the feature selection in SVR modeling and chaotic particle swarm optimization algorithm is used to search for better combinations three parameters in SVM. Compared with other models, within the forecasting fields of financial returns, the proposed model offers a valid for application in financial market.

However; the algorithm still has to be improved, other advanced searching techniques to determine the suitable parameters should be combined with the SVR model to forecast financial returns. Furthermore the number of input data influences the forecasting performance to a large extent. Thus, to develop a more effective technique available to determine the number of input data will be an important direction for future development.

ACKNOWLEDGMENT

This work was supported in part by a grant from The National Natural Science Foundation of China (61070122, 60873116); Jiangsu Provincial Natural Science Foundation (9KJA520002); Jiangsu Provincial Research Scheme of Natural Science for Higher Education Institutions (09KJA520002); Jiangsu Provincial Key Laboratory for Computer Information Processing Technology (kjs1024); JiangSu Province Support Software Engineering R&D Center for Modern Information Technology Application in Enterprise (SX200902).

REFERENCES

- [1] Granger, C. W. J., Combining forecasts- Twenty years later, *Journal of Forecasting*, Vol.8, pp.167-173, 1989.
- [2] Krogh, A. and Vedelsby, J., Neural network ensembles, cross validation, and active learning, *Advances in Neural Information Processing System*, Vol.7, pp.231-238, 1995.
- [3] Vapnik V N. *The Nature of Statistical Learning Theory* [M]. New York : Springer - Verlag ,1995.
- [4] Chang-Ying Ma, Sheng-yong , Yang,Hui Zhang, Ming-Li Xiang, Qi Huang, Yu-Quan Wei .Predicton models of human plasma protein binding rate and oral bioavailability derived by using GA-CG-SVM method, *Journal of Pharmaceutical and Biomedical Analysis* 47(2008)677-682
- [5] Vojislav, K., *Learning and Soft Computing-Support Vector Machines, Neural Networks and Fuzzy Logic Models*, The MIT Press, Massachusetts, 2001.
- [6] Thissen, U., van Brakel, R., de Weijer, A. P., Melssen, W. J. and Buydens, L. M. C., Using support vector machines for time series prediction, *Chemometrics and intelligent laboratory systems*, Vol.69, pp.35-49, 2003.
- [7] J.Cai,X.Ma,L.Li,H.Peng,Chaotic particle swarm optimization for economic dispatch considering the generator constraints, *Energy Conversion and Management* 48 (2) (2007) 645-653.
- [8] P.F.Pai, C.S. Lin, A hybrid ARIMA and support machines model in stock price forecasting. *OMEGA-International Journal of Management Science*33,(6),(2005),497-505.
- [9] LDavis, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1991.
- [10] Kusum Deep, ManoJ ThakUr. A new crossover operator for real coded genetic Algorithms[J]. *Applied Mathematics and Computation*.2006:16-19
- [11] James E. Pettinger, Richhard M. Everson. Controlling genetic algorithms with reinforcement learning[C] //Proc of the genetic and evolutionary computation conference of contents. 2003, 1-11
- [12] Clerc M. The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization[C]. *Proc of the Congress of Evolutionary Computation*. Washington, 1999:1951-1957.
- [13] Raghuvanshi M.M, Kakde O.G. Genetic Algorithm With Species And Sexual Selection[C] //Proc of the 2006 IEEE Conference on Cybernetics and Intelligent Systems. Bangkok, 2006, 1-8
- [14] Yang Chen, JingLu Hu. Optimizing Reserve Size in Genetic Algorithm with Reserve Selection Using Reinforcement Learning[C] //Proc of the SICE Annual Conference. Takamatsu, 2007, 1341-1347
- [15] Taejin Park, Ri Choe, Kwang Ryel Ryu. Adjusting Population Distance for the Dual-Population Genetic Algorithm[C] //Proc of the 20th Australian joint conference on Artificial Intelligence. Berlin Heidelberg, 2007, 171-180
- [16] Jarno Martikainen, Seppo J. Ovaska. Hierarchical Two-Population Genetic Algorithm[C] //Proc of the 2005 IEEE Mid-Summer Workshop on Soft Computing in Industrial applications. Espoo, Finland, 2005, 91-98
- [17] Shi Y, Eberhart R C. A modified particle swarm optimizer[R]. *IEEE International Conference of Evolutionary Computation*, Anchorage, Alaska, May 1998.
- [18] DW van der Merwe, AP Engelbrecht. Data Clustering using Particle Swarm Optimization[J/OL].
- [19] Clerc M., Kennedy J. The particle swarm-explosion stability, and convergence in a multidimensional complex space[J]
- [20] Eberhart R.C., Shi Y. Comparing between inertia weights and constriction factors in particle swarm optimization [C]//Proceedings of the Congress on Evolutionary Programming ,2000:84-88
- [21] Ven den Bergh F., Engelbrecht A. P. Using Neighbourhoods with the Guaranteed Convergence PSO. *Proceedings of the 2006 IEEE, Swarm Intelligence Symposium*,2003:235-242.
- [22] Angeline P.J. Using Selection to Improve particle swarm optimization[A]. *Proceedings of the 1999 Congress on Evolutionary Computation* [C]. Piscataway, NJ:IEEE Press,1999:84-89
- [23] Brskar S., Suganthan P. N. A Novel Concurrent particle swarm optimization [A]. *Proceedings of the 2004 Congress on Evolutionary Computation* [C]. Piscataway,NJ:IEEE Press,2004:792-796.
- [24] Jain, A., Zongker, D. Feature selection: Evaluation, application, and small sample performance[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997,19:153-158.
- [25] Koller, D.Sahami, M., Toward optimal feature selection[C]. In: *Proceedings of International Conference on Machine Learning*. 1996:284-292.
- [26] Kira, K., Rendell, L.A. The feature selection problem: Traditional methods and a new algorithm[C]. In: *Proceedings of Ninth National Conference on Artificial Intelligence*. 1992:129-134.
- [27] Narendra, P.M., Fukunaga, K.A branch and bound algorithm for feature selection[J]. *IEEE Transactions on Computers*, 1997.26(9):917-922.
- [28] Iqual M. Montes de Oca MA. "An Estimation of Distribution Particle Swarm optimization Algorithm," Springer-verlag Berlin Heidelberg, PP,72-82.2006.
- [29] Shi Y, Eberhart, RC. "A modified particle sarms optimization ," *Evolutionary Programming VII: Proceedings of the Seventh Annual Conference on Evolutionary Programming*, New York, PP.591-600.1998.
- [30] Storn R., Price K., "Differential evolution: A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces," Technical report, Tr-95-012, International Computer Sciences Institute, 1995
- [31] A.E. Eiben,E. Marchiori, V.A. Valko, Evolutionary algorithms with on-the-fly population size adjustment , in: *Proc.8th Conf. on Parallel Problem Solving from Nature*, Birmingham, UK, in: *Lecture Notes in Computer Science*, vol.3242, Springer, Berlin,2004,pp.41-50.
- [32] Osher S, Sethian J.A. Fronts propagating with curvature dependent speed: Algorithms based on the Hamilton-Jacobi formulation [J]. *Journal of Computational Physics*, 1988 , 79 (1): 12-49
- [33] Potter M.A, De Jong K.A. A cooperative coevolutionary approach to function optimization. In: Davidor Y, Schwefel HP, Männer R,eds. *Proc. of the Parallel Problem Solving from Nature—PPSN III, Int'l Conf. on Evolutionary Computation*. LNCS 866, Berlin: Springer-Verlag, 1994, 249–257

Yuchen Fu is an Associate Professor in the School of Computer Science & Technology, Faculty of Machine Learning & Data Mining, Soochow University. He obtained PhD in Computer Application Technology from Wuhan University, China in 2003. His research interests include intelligent computing, machine learning, and data mining.

Yuanhu Cheng, male, born in April 1986 in Tianmen, Hubei Province, major in Management Science and Engineering, the major research interests include electronic commerce, Machine Learning and data mining