**IEEE** *Access*

# An Adaptive SVR for High-Frequency Stock Price Forecasting

**YANHUI GUO** [1,2]**, SIMING HAN**[2]**, CHUANHE SHEN**[1]**, YING LI**[1]**,
XIJIE YIN**[1]**, AND YU BAI**[3]**, (Member, IEEE)**
[1]School of Information Technology, Shandong Women's University, Jinan 250300, China
[2]School of Computer Science, Shaanxi Normal University, Xi'an 710062, China
[3]School of Engineering and Computer Science, California State University, Fullerton, CA 90831, USA

Corresponding author: Ying Li (cherry_jn@126.com) and Yu Bai (ybai@fullerton.edu)

**ABSTRACT** In order to mitigate investments, stock price forecasting has attracted more attention in recent years. Aiming at the discreteness, non-normality, high-noise in high-frequency data, a support vector machine regression (SVR) algorithm is introduced in this paper. However, the characteristics in different periods of the same stock, or the same periods of different stocks are significantly different. So, SVR with fixed parameters is difficult to satisfy with the constantly changing data flow. To tackle this problem, an adaptive SVR was proposed for stock data at three different time scales, including daily data, 30-min data, and 5-min data. Experiments show that the improved SVR with dynamic optimization of learning parameters by particle swarm optimization can get a better result than compared methods including SVR and back-propagation neural network.

**INDEX TERMS** Support vector machine, regression, particle swarm optimization, stock price forecasting, high-frequency data.

## I. INTRODUCTION

Stock price forecasting is one of the most challenging tasks encountered by financial organizations and private investors. To effectively mitigate the risk and to gain high investment return, a large number of prediction models are proposed. Then, with the development of information technology, the research on finance changes from macro to micro. Research direction of study is shifted from low-frequency data to high-frequency data. Charles Sutcliffe [1] pointed out that high-frequency data offers many opportunities for more detailed analysis of market activity. Jacquier *et al.* [2] presented that the financial high-frequency data is non-normality and proposed a stochastic volatility model to solve the non-normality. Jobson and Korkie [3] shows that the high-frequency data is highly nonlinear. Subsequently, the paper [4] demonstrates that high variability of intra-minute returns across stocks is shown to have a crude U-shaped pattern.

The above literature shows that the high-frequency data has the characteristics of the discrete, non-linear and non-normal distribution. In response to these characteristics, many scholars to improve the existing financial low-frequency data analysis model to adapt to the new characteristics of high-frequency data. The models of statistical techniques, such as autoregressive integrated moving average [5], are replaced by artificial intelligence (AI) methods, including artificial neural networks (ANNs) [6], Genetic Algorithm [7], and Hidden Markov Model [8]. In general, BP neural network is the most widely used. However, BPNN suffers from the risk of over-fitting, large number of parameters, and difficulty in obtaining a stable solution [9]. To solve the problems described above, intensive research to improving ANNs are presented such as BPNN with genetic algorithm [13], [14] and ANNs with metaheuristics [15]. Recently, support vector machine regression (SVR) has also been more envisioned in nonlinear regression estimation. SVR has a better generalization owing to the structural risk minimization principle. Thus, SVR has been successfully used in many fields of time series prediction such as traffic flow prediction [10], and financial time series

forecasting [11], [12]. Nevertheless, it is difficult to set up the parameters of SVR. A model to fitting parameters fails to meet changing data. Soon afterward, some improved SVR are proposed, such as SVR with fuzzy model [16] and GA-SVR [17]. In this paper, the particle swarm optimization algorithm (PSO) is proposed to optimizing the adaption and parameters for SVR algorithm. Major contributions include:

- We discuss that stock data have different distribution characteristics in different periods or different stocks.
- The parameters of the traditional SVM are fixed in advance, and can not adapt to the changing characteristics of the financial high-frequency data.
- We propose a novel algorithm to cope with changing financial high-frequency data, which applies adaptive mechanism and particle swarm optimization algorithm to adjust parameters dynamically.
- We do a comparative experiment at three different time scales, including traditional SVR and BPNN. Experimental results show that the proposed method is more effective than others.

The rest of the paper is organized as follows. Section II reports on related works about forecasting methods for high-frequency data and time series. Section III describes an analysis of the problem, followed by our experiments in section IV. Section V concludes the paper.

## II. RELATED WORK

This section will present the principle of SVR as described by prior research. The formalization of SVR is described in detail, which is helpful to understand the proposed method in this paper. SVM/R, originally proposed by Cortes and Vapnik [18], uses a linear model to implement nonlinear class boundaries through mapping the input vectors x into the high-dimensional feature space. In the new space, a non-linear model is converted into a linear model. Thus, SVM/R is interested in the solution of the maximum margin hyperplane in the new space. Then, the classification/ regression problem is transformed into a quadratic programming problem, which is easy to solve by an optimization program.

Given a data $(x_i, y_i)$, $i = 1, 2, \ldots, n$, $x_i = R^n$, $y_i \in R$, where $x_i$ is the $n$-dimensional input, that is historical stock data in this paper, $y_i$ is the output, that is predicted value, $n$ is the number of training set. The form of SVR can be represented as the following equation:

$$f(x) = w^T \varphi(x) + b \quad \varphi : R^n \to F, \ w \in F \quad (1)$$

where $w$ is an element in the high-dimensional spaces, $\varphi(x)$ is the non-linear mapping function, $b$ is the threshold. Then, the above regression problem is to seek to minimize the risk of $f$, by minimizing the structural risk, as the following.

$$R_{reg} = \frac{1}{2} \| w \|^2 + C \times \frac{1}{n} \sum_{i=1}^{n} |y_i - f(x_i)|_\varepsilon \quad (2)$$

Where $\| w \|^2$ is the description function, $R_{emp}$ is the empirical risk, $C$ is a penalty parameter, $|y_i - f(x_i)|_\varepsilon$ is an insensitive parameter. The parameters of $C$ and $\varepsilon$ affect the precision of SVR significantly. The value of $\varepsilon$ is defined as following.

$$|y_i - f(x_i)|_\varepsilon = \begin{cases} 0, & |y_i - f(x_i)| \leq \varepsilon \\ |y_i - f(x_i)| - \varepsilon, & else \end{cases} \quad (3)$$

By introducing nonnegative slack variable $\xi$ and Lagrange function, using Karush-Kuhn-Tucker (KKT) condition, the function $f(x)$ is described as following:

$$f(x) = \sum_{i=1}^{n} (a_i - a_i^*) k(x_i, x) + b \quad (4)$$

Where $a_i$, $a_i^*$ are Lagrange multiplier, $k(x_i, x)$ is defined as the kernel function. There are some different kernels for generating the inner products to construct machines in the input space. Different kernels will have different nonlinear decision surfaces to obtain different results. We should choose the kernel according to the feature of inputs. Generally, common example of the kernel function is Gaussian radial basis function (RBF). We also choose RBF as our kernel in this paper, as following:

$$f(x) = \sum_{i=1}^{n} (a_i - a_i^*) \exp(- \| x_i - x \|^2 / 2\sigma^2) + b \quad (5)$$

Where $x_i$ is the element of training set, $x$ is the element of test set, $\sigma$ is a parameter of RBF.

However, the parameters of SVM are determined in advance. Usually, cross validation is used to determine parameter values. And the SVM with fixed parameters does not apply to constantly changing financial high frequency data.

## III. THE PROPOSED ADAPTIVE SVR

In this section, we first introduce the overall architecture of the proposed adaptive SVR in this paper. Then, details and knowledge of the proposed method are described.

### A. OVERALL ARCHITECTURE

As time goes on, the price of stock will change a lot. So, SVR based on fixed parameters is difficult to adapt the changing. Online learning will waste a lot of computing resources. Meanwhile, the speed of execution cannot meet the demand. We set up a threshold $\theta$, and once the estimation error exceeds the threshold $\theta$, the model will be trained with the latest data. For the sake of improving the parameter optimization, we introduce particle swarm optimization (PSO for short) to SVR. Finally, an adaptive SVR based PSO is proposed in our paper. The process of adaption is shown as Figure 1.

### B. THE ADAPTIVE SVR
#### 1) THE OPTIMIZATION ALGORITHM USED IN SVR
Particle swarm optimization, originally proposed by Eberhart and Kennedy [19], is a kind of evolutionary computation, which is the behavior of birds' predation. In this paper, we use PSO for short. The basic idea of PSO is to find the optimal solution through cooperating and sharing information among individuals. It is simple and easy to be
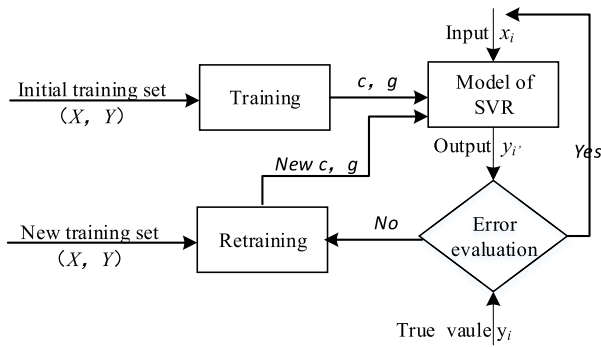
**FIGURE 1.** Process of adaption in SVR.

implemented and does not have many parameters. At present, PSO has been widely used in optimization, neural network training, fuzzy system control and so on.

In PSO, each potential solution is called a particle, the position of the i-th particle in the n-dimensional space, the flight velocity, and the optimal value are expressed as follows:

$$x_i = (x_{i,1}, x_{i,2}, \cdots, x_{i,n}) \in R^n \qquad (6)$$

$$v_i = (v_{i,1}, v_{i,2}, \cdots, v_{i,n}) \in R^n \qquad (7)$$

$$p_i = (p_{i,1}, p_{i,2}, \cdots, p_{i,n}) \in R^n \qquad (8)$$

After obtaining the optimal value of the individual and the global optimal solution, the particles update their speed, position according to the following formula:

$$v_i(t+1) = wv_i(t) + c_1r_1(p_i - x_i(t)) + c_2r_2(p_g - x_i(t)) \qquad (9)$$

$$x_i(t+1) = x_i + v_i(t+1) \qquad (10)$$

Where $i = 1, 2, \cdots, N$, $N$ is the number of training set, and $r_1, r_2$ are rand number ranged by [0, 1]. $c_1, c_2$ are acceleration factors, which range are non-negative numbers. $w$ is a factor of inertia weight.

### 2) THE ADAPTIVE SVR
In our proposed method, the PSO is used for optimizing the parameters ($C$ and g) of SVR. So, there are two dimensions in the above $x_i$, $C$ and g, respectively. $v_i$ denotes $i$-th particle's speed. And, the data are divided into training set and validation set. We choose the best parameter by optimizing the prediction accuracy of the validation set. The cost function is shown below.

$$f(C, g) = minarg \sum_m^M \sum_n^N Loss(y_i - y_i'(c_m, g_n)) \qquad (11)$$

Where, $y_i$ denotes the true value of $i$-th sample, and the $y_i'(c_m, g_n)$ means a function of $C$ and g. When the prediction model gets a model we need, we terminate the optimization process.

## IV. PROBLEM ANALYSIS
Obviously, our problem is to improve the accuracy of financial high-frequency data prediction. In this section, we mainly focus on how to use SVR for forecasting. In order to accurately forecast stock price with SVR, we studied the impacts

of penalty coefficient $C$ and distribution parameter on SVR, which are two important factors for the accuracy of prediction. We randomly selected five stocks on the Shanghai Stock Exchange as evidence. The results show that the effect of $C$ and on the prediction results is different in different periods or different stocks.

### A. THE EFFECT OF C IN DIFFERENT STOCKS OR DIFFERENT PERIODS
We randomly selected five stocks on the Shanghai Stock Exchange. In the analysis, we employ RMSE, MAPE, MAD to evaluate $C$ of SVR. Meanwhile, without loss of generality, we take three-time scales into account, including daily data, 30-minute data, and 5-minute data. We observed the accuracy of the prediction when the $C$ values ranged from $10^{-5}$ to $10^4$.

Firstly, we study the relationship between the different stocks and the $C$ value in the same period. The three indicators (RMSE, MAPE, MAD) of the five stocks (SH06, SH16, SH26, SH36, SH56, in which SH06 is SH600006 for short) are shown in figure 2. As we saw it, the optimal solutions of different stocks forecasting correspond to different $C$, although they tend to move up and down together in the same period. The experiments in 30-minute data and 5-minute data also came to the same conclusion. So, we only show the result of the daily data.

Subsequently, we validated our assumption that the same stock has different data distributions at different times and that different $C$ values need to be set to obtain the best prediction. So, we selected three stocks (SH06, SH16, SH26), and chosen three periods (from January to March) of each stock to investigate. The detailed results of RMSE in three stocks are illustrated in table 1. S06-15 means the stock SH06 in the year of 2015. The table 1 reveals that the prediction performance of SVR is better when the $C$ value is between 0 and 100 which is the range of particle swarm optimization mentioned later. It also can be seen that the stock SH06 gained the best result when $C$ was 0 in 2015, $C$ is 10 in 2016, and $C$ is 100 in 2017. The same case happens with the other two stocks. Hence, we can get the conclusion that a stock needs to set up different $C$ values for SVR at the different period to get better performance.

### B. THE EFFECT OF G IN DIFFERENT STOCKS OR DIFFERENT PERIODS
g is also of great importance to affect the ability of forecasting in the SVR. We study the effects of g with the same analysis methods as mentioned above. Firstly, we study the different effects of g on three different time scales at the same period, such as daily data, 30-minute data and 5-minute data. Then, we analyze the change of g in the same stock in different periods. In order to seek out the change regulation of g in SVR, we evaluated the error of the SVR model by g ranged from $10^{-5}$ to $10^4$.

Figure 3. shows the changing of SVR prediction results for daily data by different g. It is clear from the Fig.2 that the prediction results are better when the $C$ value is
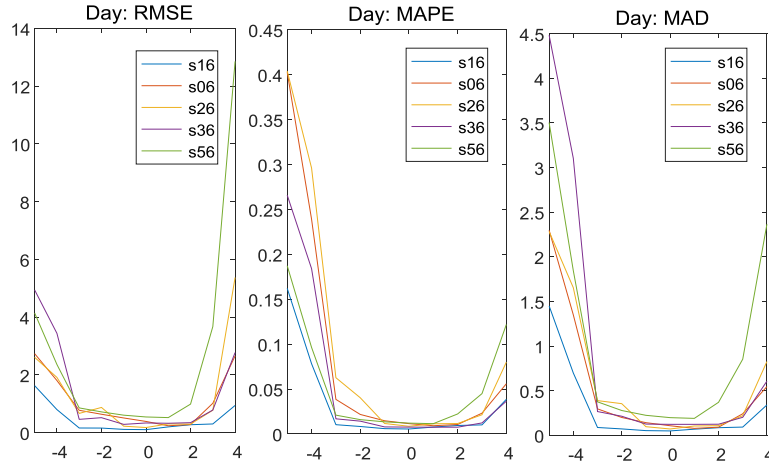
**FIGURE 2.** The effect of C in different stocks.

**TABLE 1.** The effect of C in a stock of different periods.

| $C$ | $10^{-5}$ | $10^{-4}$ | $10^{-3}$ | $10^{-2}$ | $10^{-1}$ | $10^{0}$ | $10^{1}$ | $10^{2}$ | $10^{3}$ | $10^{4}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| S06-15 | 38.17759 | 37.29998 | 32.42391 | 8.97841 | 7.908533 | **6.877861** | 6.938308 | 7.052259 | 7.037814 | 10.75368 |
| S06-16 | 332.8834 | 317.1489 | 158.5287 | 37.14391 | 27.50355 | 17.79475 | **16.81279** | 17.13971 | 17.90291 | 48.32511 |
| S06-17 | 247.0708 | 241.9352 | 181.8897 | 18.08049 | 11.29921 | 8.343109 | 8.364469 | **8.042893** | 8.212761 | 23.93901 |
| S16-15 | 467.2349 | 458.7753 | 396.5228 | 34.13317 | 18.06041 | 13.02689 | 14.02715 | **12.28269** | 13.92547 | 47.95066 |
| S16-16 | 468.5434 | 465.9288 | 434.6615 | 141.7109 | 32.81114 | **24.74259** | 30.74406 | 30.40175 | 33.49616 | 202.6654 |
| S16-17 | 207.0085 | 204.7469 | 182.9439 | 34.95354 | 16.19178 | 9.717074 | 9.45994 | **8.495672** | 9.163291 | 20.34024 |
| S26-15 | 38.17759 | 37.29998 | 32.42391 | 8.97841 | 7.908533 | **6.877861** | 6.938308 | 7.052259 | 7.037814 | 10.75368 |
| S26-16 | 332.8834 | 317.1489 | 158.5287 | 37.14391 | 27.50355 | 17.79475 | **16.81279** | 17.13971 | 17.90291 | 48.32511 |
| S26-17 | 247.0708 | 241.9352 | 181.8897 | 18.08049 | 11.29921 | 8.343109 | 8.364469 | **8.042893** | 8.212761 | 23.93901 |

between $10^{-2}$ and $10^{2}$. However, there are different C values for different stocks. Experiments in 30-minute data and 5-minute data also prove that our conjecture is correct. The distribution of stock data is inconsistent at different times.



**FIGURE 3.** The effect of g in different stocks or different periods.

Figure 3. shows the changing of SVR prediction results for daily data by different g. It is clear from the Fig.2 that the prediction results are better when the C value is between $10^{-2}$ and $10^{2}$. However, there are different C values for different stocks. Experiments in 30-minute data and 5-minute data also prove that our conjecture is correct. The distribution of stock data is inconsistent at different times.
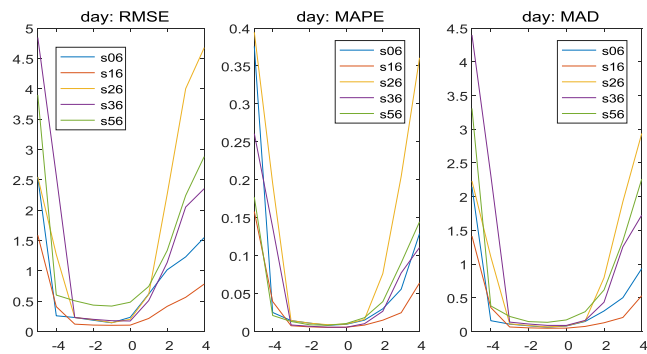
From the above analysis, it is concluded that the impacts of *C* and g on prediction results are different in different stocks or different periods. Therefore, a SVR model of fixed parameters cannot be applicable to different stocks, or different periods of the same stock. To solve this problem, we propose an adaptive SVR based on particle swarm optimization, which can improve the versatility and efficiency of the program. The adaptive SVR applied a feedback mechanism to adjust the parameter of SVR dynamically. When the model error exceeds a threshold $\theta$, the SVR will retrain the model soon.

## V. EXPERIMENTS AND RESULTS
### A. DATASET AND DATA MODELING
#### 1) DATASET
The data set is from Shanghai Stock Exchange, including SH600006, SH600016, SH600026, SH600036, SH600056, which can be download by *http://www.wstock.net*. We choose three different time-scales (or named frequencies) to validate

| stocks | RESE(e-2) | | | MAPE(e-2) | | | MAD(e-2) | | |
|---|---|---|---|---|---|---|---|---|---|
| | **BPNN** | **SVR** | **AD_SVR** | **BPNN** | **SVR** | **AD_SVR** | **BPNN** | **SVR** | **AD_SVR** |
| SH06 | 0.66 | 0.75 | 0.65 | 0.072 | 0.08 | 0.068 | 0.52 | 0.58 | 0.49 |
| SH16 | 0.92 | 1.04 | 0.63 | 0.081 | 0.09 | 0.056 | 0.74 | 0.81 | 0.51 |
| SH26 | 0.77 | 0.79 | 0.70 | 0.084 | 0.089 | 0.077 | 0.61 | 0.64 | 0.56 |
| SH36 | 1.48 | 1.81 | 1.23 | 0.056 | 0.064 | 0.049 | 1.09 | 1.24 | 0.94 |
| SH56 | 4.66 | 6.17 | 2.53 | 0.11 | 0.18 | 0.07 | 2.47 | 3.95 | 1.54 |

our approach namely, the daily data, the high frequency data of 30 minutes, and the high frequency data of 5 minutes. Daily data is from the listing date to March 31, 2017. The 30-minute data is from January 1, 2017 to March 31, 2017. And the 5-minute data ranges from February 1, 2017 to February 28, 2017. By the way, we take eighty percent of the data as the training set, and twenty percent of the data as the test set in our experiment.

### 2) DATA MODELING

Let us denote $X = \{x_1, x_2, \cdots, x_n\}$ as the time series and the training set, where $x_i = \left(x_i^1, \cdots, x_i^5\right) \epsilon R^5$ is the i-th data point and n is the total number of training set, $x_i$ is composed of 5 elements, including open, high, low, close data, and trading volumes. $Y = \{x_2^1, x_3^1, \cdots, x_{n+1}^1\}$ is the result according to the $X$. That is to say, we employ the previous time's trading information to forecast the open price for the next time, which is $\left(x_i^1, \cdots, x_i^5\right) \rightarrow x_{i+1}^1$. In order to fit the model of SVR, the data is preprocessed by normalization. Linear transformation for each column is done, by dividing the maximum value, so that the data is mapped to [0, 1].

### B. EXPERIMENT SETUP AND COMPARISON

### 1) EXPERIMENT SETUP

Good parameter settings will lead to the better performance. This article has three model parameters that need to be set, including traditional SVR, BPNN, and adaptive SVR. Then the paper describes the parameter settings in this experiment in order to reproduce the experimental results. In the traditional SVR model, the parameters are $C = 2.2$ and $g = 2.8$ which are the best settings by PSO optimization. And in the BPNN model, the parameters are $epochs = 1000$, $lr = 0.01$ and $goal = 0.001$, where $epochs$ denotes the maximum number of iterations, $lr$ denotes learning rate which is a common parameter in machine learning algorithms, and $goal$ represents minimum error of training target. The weights of BPNN are randomly initialized.

We use the same parameter setting with mentioned traditional SVR, as the initial value of the adaptive SVR. And the threshold of the adaptive SVR is $\theta = 0.05$ for daily data, $\theta = 0.02$ for other data. The value of $C$ is ranged from 1 to 100, and the g is from 0.1 to 10. In accordance with the method described in section 2.2, the parameters utilized in PSO are set as follows: the inertia weight is $w = 0.7$, the acceleration factors are $c1 = 0.01$ and $c2 = 0.01$. Besides, the maximum number of iterations is $iter = 400$, and the initial population number is 40.

### 2) EVALUATION METRICS

To evaluate the proposed method, we apply three widely used quality indexes, i.e., Root Mean Square Error (RMSE), Mean Absolute Percent Error (MAPE), Mean Absolute Deviation (MAD). The formulas are as following:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(observed_i - predicted_i\right)^2} \qquad (12)$$

$$MAPE = \sum_{i=1}^{N} \left| \frac{observed_i - predicted_i}{observed_i} \right| \times \frac{100}{N} \qquad (13)$$

$$MAD = \frac{1}{N} \sum_{i}^{N} \left| \left(observed_i - predicted_i\right) \right| \qquad (14)$$

### 3) PERFORMANCE COMPARISON

In order to verify the effectiveness and robustness of the proposed method, we performed experiments on different stocks and different time scales. The BPNN model adopted by [11] and traditional SVR model are used as comparison methods which are commonly used for prediction. In the following, we discuss the experimental results in details.

The first experiment is performed on the data set of 5 minutes. We compared the performance of BPNN, traditional SVR, and adaptive SVR by the quantitative index of RESE, MAPE, and MAD. The detailed results are shown in Table 2. Obviously, the result of AD-SVR is the best, followed by BPNN, and SVR is the worst. It can be discerned from the table 2 that the forecast results of the five stocks have been greatly improved. The worst one of the five stocks is SH26, which was 11% lower than the traditional SVR, in RESE, MAPE, and MAD. And the SH26 stock is also better than BPNN. However, the best one of the five stocks is SH56, whose error has been reduced by about 60% compared with traditional SVR, and reduced by about 45% compared with BPNN.

To compare the results intuitively, we show one of our experimental results in figure 2. This figure illustrates the prediction results of SVR, BPNN, AD-SVR, and the true values which ranges from 21.66 to 22.47. It can be seen that the result of AD-SVR is closest to the real value than that of SVR and BPNN, especially where the peak appears. As time went on, the prediction of SVR became worse and worse. As far as the section 3 is concerned, the reason is that the change of data distribution feature leads to lower SVR performance of prediction.

The second experiment is conducted on the data set of 30 minutes. The errors of SVR, BPNN, and adaptive SVR are shown in Table 3. Compared with traditional SVR,
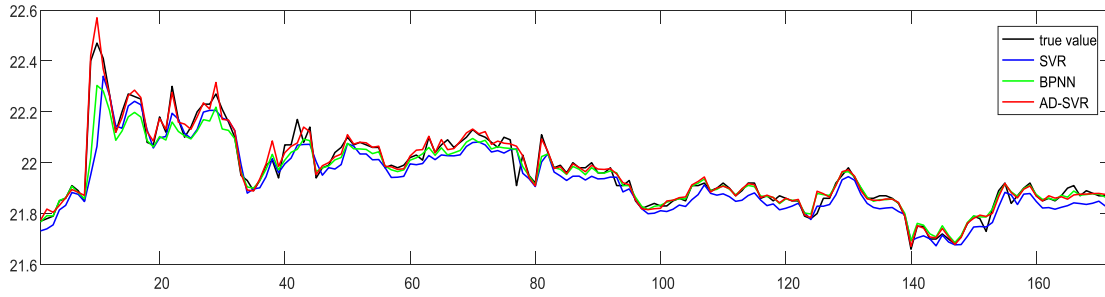
**FIGURE 4.** The result comparison of SH56 stock in 5-min data.

**TABLE 3.** The results for 30 minutes data.

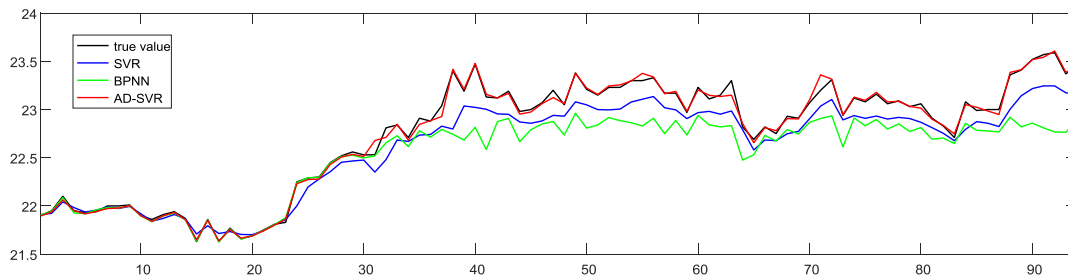| stocks | RESE(e-2) | | | MAPE(e-2) | | | MAD(e-2) | | |
|--------|------|------|--------|------|------|--------|------|------|--------|
| | BPNN | SVR | AD_SVR | BPNN | SVR | AD_SVR | BPNN | SVR | AD_SVR |
| SH06 | 1.47 | 1.47 | 1.25 | 0.16 | 0.15 | 0.12 | 1.08 | 1.05 | 0.83 |
| SH16 | 2.73 | 3.12 | 0.80 | 0.25 | 0.29 | 0.074 | 2.14 | 2.46 | 0.64 |
| SH26 | 2.77 | 5.58 | 2.72 | 0.18 | 0.30 | 0.15 | 1.32 | 2.24 | 1.07 |
| SH36 | 2.99 | 5.32 | 3.46 | 0.07 | 0.13 | 0.09 | 1.29 | 2.55 | 1.71 |
| SH56 | 29.77 | 18.13 | 3.89 | 0.94 | 0.63 | 0.11 | 21.91 | 14.47 | 2.44 |



**FIGURE 5.** The result comparison of SH56 stock in 30-min data.

**TABLE 4.** The results for daily data.

| stocks | RESE(e-2) | | | MAPE(e-2) | | | MAD(e-2) | | |
|--------|------|------|--------|------|------|--------|------|------|--------|
| | BPNN | SVR | AD_SVR | BPNN | SVR | AD_SVR | BPNN | SVR | AD_SVR |
| SH06 | 30.75 | 50.77 | 24.89 | 1.15 | 1.47 | 1.06 | 9.77 | 14.13 | 9.25 |
| SH16 | 11.84 | 11.24 | 6.95 | 0.62 | 0.59 | 0.30 | 5.47 | 5.20 | 3.30 |
| SH26 | 28.32 | 22.34 | 20.52 | 0.92 | 1.16 | 0.88 | 8.13 | 9.60 | 7.23 |
| SH36 | 20.42 | 28.60 | 20.41 | 0.64 | 0.78 | 0.66 | 10.55 | 12.53 | 10.16 |
| SH56 | 55.16 | 60.48 | 35.46 | 1.37 | 1.35 | 0.83 | 22.26 | 22.35 | 14.73 |

it can be seen that improvements are more pronounced than that in the data set of 5 minutes. The worst one of the five stocks is SH06, whose error has been reduced by about 15%. And the stock SH56 is the best one whose error is 79% lower than traditional SVR in RESE, MAPE, MAD. For BPNN, this experiment is difficult to draw conclusions. The results in the SH16, SH26, and SH36 are better than that of traditional SVR. But the result in the SH56 is worse than that of SVR. This could be due to BPNN being affected by initialization parameters. Yet, compared with BPNN, the results of AD_SVR are almost superior to that of BPNN, except for SH36. And AD_SVR is more robust than BPNN due to nonrandom parameters.

Similarly, we show one of our results by the following figure 5. Compared with 5-minute data, the 30-minute data

fluctuated significantly, ranging from 21.63 to 23.59. As we can see from figure 5, the first 25 predictions of SVR, BPNN, and AD_SVR are almost the same as the true value. As time goes by, the prediction of SVR and BPNN becomes worse and worse. However, the prediction of AD_SVR is always moving up and down around the true value. As shown in figure 5, the red line and the black line are almost coincident.

The third experiment is implemented on the daily data set. The results are indicated in Table 4. The results are similar with the previous experiments. Compared with SVR, the errors of AD_SVR in different stocks have been reduced, ranging from 8% to 41%. It should be noted that the training set of adaptive SVR is much smaller than that of traditional SVR, in order to improve the speed of the program. The results of BPNN is better than that of SVR in SH06, SH26,
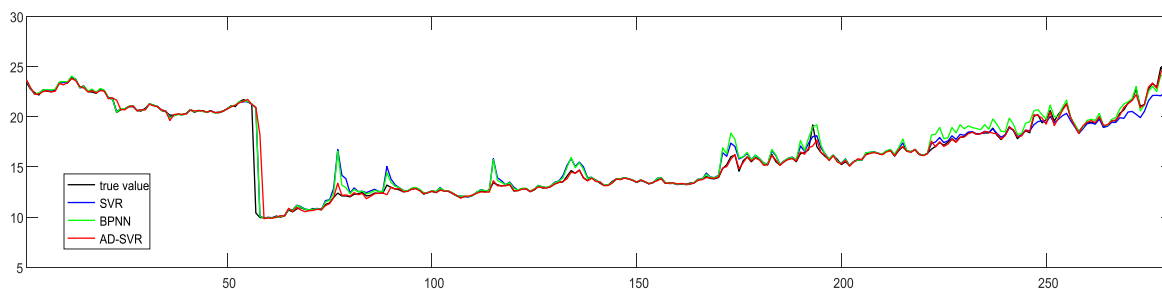
**FIGURE 6.** The result comparison of SH56 stock in daily data.

and worse than that of SVR in SH36 and SH56. However, AD_SVR outperforms BPNN and SVR.

The following figure 6 shows the results of SH56 in daily data, which changes from 9.91 to 25.10. Changes are consistent with those experiments mentioned above. The results of SVR and BPNN are getting worse. We can see that several peaks of the green and blue lines deviated from the true value greatly.

From the experiment we can see that, BPNN is easily affected by random initialization parameters. So, it is difficult to gain a good performance for all stocks. And traditional SVR does not apply to changing stock data. However, AD_SVR has a strong robustness to adapt to changing stocks (different stocks, different periods, different time-scales and so on). The verification experimental results indicate the effectiveness and reliability of AD_SVR model.

## VI. CONCLUSION
In this paper, an adaptive SVR based on PSO is proposed to enhance the versatility of the model and to avoid suffering from adjusting parameters of SVR. We tested our proposed algorithm on three time-scales on the stocks of Shanghai Stock Market. The results showed that the adaptive SVR has better adaptability and better prediction results than the traditional SVR and BPNN. And we don't take the impact of historical models into account in this experiment. So, a weighted adaptive SVR will be introduced in our future work.

## VII. ACKNOWLEDGMENT
Yanhui Guo and Siming Han contributed equally to this work.

## REFERENCES
[1] O. Gwilym and C. Sutcliffe, "Problems encountered when using high frequency financial market data : Suggested solutions," *J. Financial Manage. Anal.*, vol. 25, no. 2, pp. 15–28, Dec. 2012.

[2] E. Jacquier, N. G. Polson, and P. E. Rossi, "Bayesian analysis of stochastic volatility models with fat-tails and correlated errors," *J. Econ.*, vol. 122, no. 1, pp. 185–212, Sep. 2004.

[3] J. D. Jobson and B. Korkie, "Estimation for Markowitz efficient portfolios," *J. Amer. Statist. Assoc.*, vol. 75, no. 371, pp. 544–554, Mar. 1980.

[4] T. H. McInish and R. A. Wood, "A transactions data analysis of the variability of common stock returns during 1980–1984," *J. Banking Finance*, vol. 14, no. 1, pp. 99–112, Mar. 1990.

[5] G. E. P. Box and D. A. Pierce, "Distribution of residual autocorrelations in autoregressive-integrated moving average time series models," *J. Amer. Statist. Assoc.*, vol. 65, no. 332, pp. 1509–1526, Apr. 1970.

[6] G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: The state of the art," *Int. J. Forecasting*, vol. 14, no. 1, pp. 35–62, 1998.

[7] K.-J. Kim and I. Han, "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index," *Expert Syst. Appl.*, vol. 19, no. 2, pp. 125–132, Aug. 2000.

[8] M. R. Hassan and B. Nath, "Stock market forecasting using hidden Markov model: A new approach," in *Proc. ISDA*, Wroclaw, Poland, Sep. 2005, pp. 192–196.

[9] L. Cao and F. E. H. Tay, "Financial forecasting using support vector machines," *Neural Comput. Appl.*, vol. 10, no. 2, pp. 184–192, Mar. 2001.

[10] W.-C. Hong, Y. Dong, F. Zheng, and C.-Y. Lai, "Forecasting urban traffic flow by SVR with continuous ACO," *Appl. Math. Model.*, vol. 35, no. 3, pp. 1282–1291, 2011.

[11] X. Liang, H. Zhang, J. Xiao, and Y. Chen, "Improving option price forecasts with neural networks and support vector regressions," *Neurocomputing*, vol. 72, nos. 13–15, pp. 3055–3065, Aug. 2009.

[12] C.-J. Lu, T.-S. Lee, and C.-C. Chiu, "Financial time series forecasting using independent component analysis and support vector regression," *Decision Support Syst.*, vol. 47, no. 2, pp. 115–125, May 2009.

[13] M. T. Motlagh and H. Khaloozadeh, "A new architecture for modeling and prediction of dynamic systems using neural networks: Application in Tehran stock exchange," in *Proc. ICCIA*, Jeju, South Korea, Jun. 2016, pp. 196–201.

[14] S. C. Nayak, B. B. Misra, and H. S. Behera, "An adaptive second order neural network with genetic-algorithm-based training (ASONN-GA) to forecast the closing prices of the stock market," *Int. J. Appl. Metaheuristic Comput.*, vol. 7, no. 2, pp. 39–57, Feb. 2016.

[15] M. Göçken, M. Özçalici, A. Boru, and A. T. Dosdogru, "Integrating metaheuristics and artificial neural networks for improved stock price prediction," *Expert Syst. Appl.*, vol. 44, pp. 320–331, Feb. 2016.

[16] R. J. Kuo and P. S. Li, "Taiwanese export trade forecasting using firefly algorithm based *K*-means algorithm and SVR with wavelet transform," *Comput. Ind. Eng.*, vol. 99, pp. 153–161, Sep. 2016.

[17] Q. Duan, L. Zhang, F. Wei, X. Xiao, and L. Wang, "Forecasting model and validation for aquatic product price based on time series GA-SVR," *Trans. Chin. Soc. Agricult. Eng.*, vol. 33, no. 1, pp. 308–314, Jan. 2017.

[18] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[19] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. MHS*, Nagoya, Japan, Oct. 1995, pp. 39–43.

**YANHUI GUO** received the B.S. degree in information management and information system from the Xi'an University of Finance and Economics, Xi'an, China, in 2006, and the M.S. degree in computer software and theory from Shaanxi Normal University, Xi'an, in 2009, where he is currently pursuing the Ph.D. degree with the School of Computer Science.

Since 2009, he has been with the School of Information Technology, Shandong Women's University, Jinan, China, where he is currently a Senior Lecturer. His research interests include data mining and machine learning.

**SIMING HAN** received the B.S. degree from the National Pilot School of Software, Yunnan University, Kunming, China, in 2014. He is currently pursuing the M.S. degree with the School of Computer Science, Shaanxi Normal University, Xi'an, China.

His research interests include data mining and machine learning.

**CHUANHE SHEN** was born in 1966. He received the Ph.D. degree from the Shandong University of Science and Technology.

He is currently a Professor with Shandong Women's University. He is also the Director of the Institute of Financial Engineering. His research interests are financial engineering, data mining, and risk management. His research works have been published in the *Communications in Statistics*, the *International Journal of Advancements in Computing Technology*, and the *Journal of Applied Sciences*. He has undertaken some research programs from National Natural Science Foundation of China and Humanity and Social Science Research Foundation of Ministry of Education of China.

**YING LI** received the B.S. degree in computer software and theory from Shandong University, China, in 1993, and the M.E. degree in software engineering from the Dalian University of Technology, China, in 2010.

She is currently a Professor of computer software and theory with Shandong Women's University, China. She has authored over 10 peer-reviewed journal and conference papers, and earned two Chinese patents. Her research interests include data mining, intelligent algorithm, and their applications in business, finance, and biology. Her research was supported by the National Science and Technology Support Program of China, the Chinese Department of Technology, and some companies.
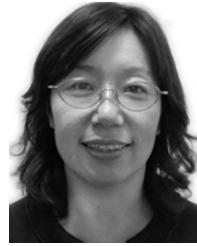
**XIJIE YIN** received the M.S. degree from the Shandong University of Computer Software and Theory, Shandong, China, in 2006.

She is currently a Professor with the School of Computer, Shandong Women's University. She has authored over 10 research papers. Her research interests include data mining and machine learning.

**YU BAI** (M'17) received the B.S. degree in electrical engineering from National Aviation University, Ukraine, in 2008, the M.S. degree in electrical and computer engineering from The University of Texas–Pan American in 2011, and the Ph.D. degree in electrical engineering from the University of Central Florida in 2016. He is currently an Assistant Professor in the Computer Engineering Program with the College of Engineering and Computer Science, California State University, Fullerton. Prior to his academic career, he had been at Siemens Energy Inc. His research interests include stochastic computing, neuromorphic computing, FPGA design, nano-scale computing system with novel silicon and post-silicon devices, and low power digital and mixed-signal CMOS circuit design.

• • •