



Forecasting stock market indexes using principle component analysis and stochastic time effective neural networks



Jie Wang*, Jun Wang*

Institute of Financial Mathematics and Financial Engineering, School of Science, Beijing Jiaotong University, Beijing 100044, PR China

ARTICLE INFO

Article history:

Received 25 July 2014

Received in revised form

18 October 2014

Accepted 30 December 2014

Communicated by P. Zhang

Available online 9 January 2015

Keywords:

Forecast

Stochastic time effective neural network

Principal component analysis

Financial time series model

ABSTRACT

Financial market dynamics forecasting has long been a focus of economic research. A stochastic time effective function neural network (STNN) with principal component analysis (PCA) developed for financial time series prediction is presented in the present work. In the training modeling, we first use the approach of PCA to extract the principal components from the input data, then integrate the STNN model to perform the financial price series prediction. By taking the proposed model compared with the traditional backpropagation neural network (BPNN), PCA-BPNN and STNN, the empirical analysis shows that the forecasting results of the proposed neural network display a better performance in financial time series forecasting. Further, the empirical research is performed in testing the predictive effects of SSE, HS300, S&P500 and DJIA in the established model, and the corresponding statistical comparisons of the above market indices are also exhibited.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Financial price time series prediction has recently garnered significant interest among investors and professional analysts. Artificial neural network (ANN) is one of the technologies that have made great progress in the study of stock market dynamics. Neural networks can provide models for a large class of natural and artificial phenomena that are difficult to handle using classical parametric techniques [1–5]. Usually stock prices can be seen as a random time sequence with noise, and a number of analysis methods have utilized artificial neural networks to predict stock price trends [6–11]. Artificial neural networks have good self-learning ability, a strong anti-jamming capability, and have been widely used in the financial fields such as stock prices, profits, exchange rate and risk analysis and prediction [12–14].

To improve predicting precision, various network architectures and learning algorithms have been developed in the literature [15–19]. The backpropagation neural network (BPNN) is a neural network training algorithm for financial forecasting, which has powerful problem-solving ability. Multilayer perceptron (MLP) is one of the most prevalent neural networks, which has the capability of complex mapping between inputs and outputs that makes it possible to approximate nonlinear function [19,20]. In the present work, we apply MLP with backpropagation algorithm and stochastic time strength function to develop a stock price volatility forecasting model. In the

real financial markets, the investing environments as well as the fluctuation behaviors of the markets are not invariant. If all the data are used to train the network equivalently, the expressing ability of network is likely to be curtailed for policymakers concerning about the current regular system. For example, see the Chinese stock markets in 2007, the data in the training set should be time-variant, reflecting the different behavior patterns of the markets at different times. If only the recent data are selected, a lot of useful information (which the early data hold) will be lost. In this research, a stochastic time effective neural network (STNN) and the corresponding learning algorithm are presented. For this improved network model, each of historical data is given a weight depending on the time at which it occurs. The degree of impact of historical data on the market is expressed by a stochastic process [17–19], where a drift function and the Brownian motion are introduced in the time strength function in order to make the model have the effect of random movement while maintaining the original trend.

This paper presents an improved method which integrates the principal component analysis (PCA) into a stochastic time strength neural network (STNN) for forecasting financial time series, called PCA-STNN model. The approach of PCA-STNN is to extract the principal components (PCs) from the input data according to the PCA method, and use PCs as the input of STNN model which can eliminate redundancies of original information and remove the correlation between the inputs [21–28]. In order to display that the PCA-STNN model outperforms the PCA-BPNN model, the BPNN model and the STNN model in forecasting the fluctuations of stock markets, we compare the forecasting performance of the above four forecasting models by selecting the data of the global stock indices,

* Corresponding author.

E-mail address: wangjiebjtu@yeah.net (J. Wang).

including Shanghai Stock Exchange (SSE) Composite Index, Hong Kong Hang Seng 300 Index (HS300), Dow Jones Industrial Average Index (DJIA), and Standard & Poor's 500 Index (S&P 500).

2. Methodology

2.1. Stochastic time effective neural network (STNN)

Artificial neural network has been extensively used as a method to forecast financial market behaviors. The backpropagation algorithm has emerged as one of the most widely used learning procedures for multilayer networks [27–29]. In Fig. 1, a three-layer multi-input BPNN model is exhibited, the corresponding structure is $m \times n \times 1$, where m is the number of inputs, n is the number of neurons in the hidden layer and one output unit. Let x_{it} ($i = 1, 2, \dots, m$) denote the set of input vector of neurons at time t , and y_{t+1} denote the output of the network at time $t + 1$. Between the inputs and the output, there is a layer of processing units called hidden units. Let z_{jt} ($j = 1, 2, \dots, n$) denote the output of hidden layer neurons at time t , w_{ij} is the weight that connects the node i in the input layer neurons to the node j in the hidden layer, v_j is the weight that connects the node j in the hidden layer neurons to the node in the output layer. Hidden layer stage is as follows: The input of all neurons in the hidden layer is given by

$$\text{net}_{jt} = \sum_{i=1}^n w_{ij}x_{it} - \theta_j, \quad i = 1, 2, \dots, n. \quad (1)$$

The output of hidden neuron is given by

$$z_{jt} = f_H(\text{net}_{jt}) = f_H\left(\sum_{i=1}^n w_{ij}x_{it} - \theta_j\right), \quad i = 1, 2, \dots, n \quad (2)$$

where θ_j is the threshold of neuron in hidden layer. The sigmoid function in hidden layer is selected as the activation function: $f_H(x) = 1/(1 + e^{-x})$. The output of the hidden layer is given as follows:

$$y_{t+1} = f_T\left(\sum_{j=1}^m v_j z_{jt} - \theta_T\right) \quad (3)$$

where θ_j is the threshold of neuron in output layer and $f_T(x)$ is an identity map as the activation function.

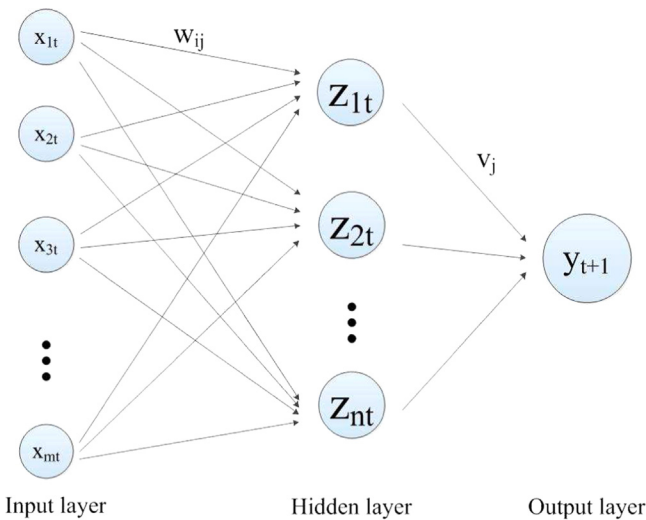


Fig. 1. Three-layer neural network with one-output.

2.2. Predicting algorithm with a stochastic time effective function

The backpropagation algorithm is a supervised learning algorithm which minimizes the global error E by using the gradient descent method. For the STNN model, we assume that the error of the output is given by $\varepsilon_{t_n} = d_{t_n} - y_{t_n}$ and the error of the sample n is defined as

$$E(t_n) = \frac{1}{2} \varphi(t_n)(d_{t_n} - y_{t_n})^2 \quad (4)$$

where t_n is the time of the sample n ($n = 1, \dots, N$), d_{t_n} is the actual value, y_{t_n} is the output at time t_n , and $\varphi(t_n)$ is the stochastic time effective function which endows each historical data with a weight depending on the time at which it occurs. We define $\varphi(t_n)$ as follows:

$$\varphi_{t_n} = \frac{1}{\beta} \exp\left\{\int_{t_0}^{t_n} \mu(t) dt + \int_{t_0}^{t_n} \sigma(t) dB(t)\right\} \quad (5)$$

where $\beta(>0)$ is the time strength coefficient, t_0 is the time of the newest data in the data training set, and t_n is an arbitrary time point in the data training set. $\mu(t)$ is the drift function, $\sigma(t)$ is the volatility function, and $B(t)$ is the standard Brownian motion.

Intuitively, the drift function is used to model deterministic trends, the volatility function is often used to model a set of unpredictable events occurring during this motion, and Brownian motion is usually thought as random motion of a particle in liquid (where the future motion of the particle at any given time is not dependent on the past). Brownian motion is a continuous-time stochastic process, and it is the limit of or continuous version of random walks. Since Brownian motion's time derivative is everywhere infinite, it is an idealized approximation to actual random physical processes, which always have a finite time scale. We begin with an explicit definition. A Brownian motion is a real-valued, continuous stochastic process $\{Y(t), t \geq 0\}$ on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, with independent and stationary increments. In detail, (a) continuity: the map $s \mapsto Y(s)$ is continuous \mathbb{P} a.s.; (b) independent increments: if $s \leq t$, $Y_t - Y_s$ is independent of $\mathcal{F} = (Y_u, u \leq s)$; (c) stationary increments: if $s \leq t$, $Y_t - Y_s$ and $Y_{t-s} - Y_0$ have the same probability law. From this definition, if $\{Y(t), t \geq 0\}$ is a Brownian motion, then $Y_t - Y_0$ is a normal random variable with mean rt and variance $\sigma^2 t$, where r and σ are constant real numbers. A Brownian motion is standard (we denote it by $B(t)$) if $B(0) = 0$ \mathbb{P} a.s., $\mathbb{E}[B(t)] = 0$ and $\mathbb{E}[B(t)]^2 = t$. In the above random data-time effective function, the impact of the historical data on the stock market is regarded as a time variable function, the efficiency of the historical data depends on its time. Then the corresponding global error of all the data at each network repeated training set in the output layer is defined as

$$E = \frac{1}{N} \sum_{n=1}^N E(t_n) = \frac{1}{2N} \sum_{n=1}^N \frac{1}{\beta} \exp\left\{\int_{t_0}^{t_n} \mu(t) dt + \int_{t_0}^{t_n} \sigma(t) dB(t)\right\} (d_{t_n} - y_{t_n})^2. \quad (6)$$

The main objective of learning algorithm is to minimize the value of cost function E until it reaches the pre-set minimum value ξ by repeated learning. On each repetition, the output is calculated and the global error E is obtained. The gradient of the cost function is given by $\Delta E = \partial E / \partial W$. For the weight nodes in the input layer, the gradient of the connective weight w_{ij} is given by

$$\Delta w_{ij} = -\eta \frac{\partial E(t_n)}{\partial w_{ij}} = \eta \varepsilon_{t_n} v_j \varphi(t_n) f'_H(\text{net}_{jt_n}) x_{it_n} \quad (7)$$

and for the weight nodes in the hidden layer, the gradient of the connective weight v_j is given by

$$\Delta v_j = -\eta \frac{\partial E(t_n)}{\partial v_j} = \eta \varepsilon_{t_n} \varphi(t_n) f_H(\text{net}_{jt_n}) \quad (8)$$

where η is the learning rate and $f'_H(\text{net}_{jt_n})$ is the derivative of the activation function. So the update rule for the weight w_{ij} and v_j is given by

$$w_{ij}^{k+1} = w_{ij}^k + \Delta w_{ij}^k = w_{ij}^k + \eta \varepsilon_{t_n} v_j \varphi(t_n) f'_H(\text{net}_{jt_n}) x_{it_n} \quad (9)$$

$$v_j^{k+1} = v_j^k + \Delta v_j^k = v_j^k + \eta \varepsilon_{t_n} \varphi(t_n) f_H(\text{net}_{jt_n}). \quad (10)$$

Note that the training aim of the stochastic time effective neural network is to modify the weights so as to minimize the error between the network's prediction and the actual target. When all the training data are the new data (that is $t_0 = t_n$), the stochastic time effective neural network is the general neural network model. In Fig. 2, the training algorithm procedures of the stochastic time effective neural network are displayed, which are as follows: *Step 1*: Perform input data normalization. In STNN model, we choose four kinds of stock prices as the input values in the input layer: daily opening price, daily highest price, daily lowest price, and daily closing price. The output layer is the closing price of the next trading day. Then determine the network structure which is $n \times m \times 1$ three-layer network model, parameters including learning rate η which is between 0 and 1, the maximum training iterations number K , and initial connective weights. *Step 2*: At the beginning of data processing, connective weights v_j and w_{ij} follow the uniform distribution on $(-1, 1)$, and let the neural threshold θ_j and θ_T be 0. *Step 3*: Introduce the stochastic

time effective function $\varphi(t)$ in the error function E . Choose the drift function $\mu(t)$ and the volatility function $\sigma(t)$. Give the transfer function from the input layer to the hidden layer, and the transfer function from the hidden layer to the output layer. *Step 4*: Establish an error acceptable model and set pre-set minimum error ξ . Based on network training objective $E = (1/N) \sum_{n=1}^N E(t_n)$, if the E is below pre-set minimum error, go to *Step 6*, otherwise go to *Step 5*. *Step 5*: Modify the connective weights: calculate the gradient of the connective weights w_{ij} , Δw_{ij}^k . Calculate the gradient of the connective weights v_j , Δv_j^k . Then modify the weights from the layer to the previous layer, w_{ij}^{k+1} or v_j^{k+1} . *Step 6*: Output the predictive value $y_{t+1} = f_T(\sum_{j=1}^m v_j f_H(\sum_{i=1}^n w_{ij} x_{it}))$.

2.3. PCA-STNN forecasting model

PCA is one of the most widely used exploratory multivariate statistical methods which are used to identify latent structures [30]. PCs are linear combinations of original variables in which the weights allocated to the linear combinations of those original variables are termed eigenvectors. PCs are also called factors, latent variables, loading and modes in engineering. The benefit of PCA is to represent complex multidimensional data with fewer PCs without losing much valuable information. One of the objectives of PCA is to discover and reduce dimensionality of the dataset by clustering them [31]. Ouyang [32] used PCA method to evaluate the ambient

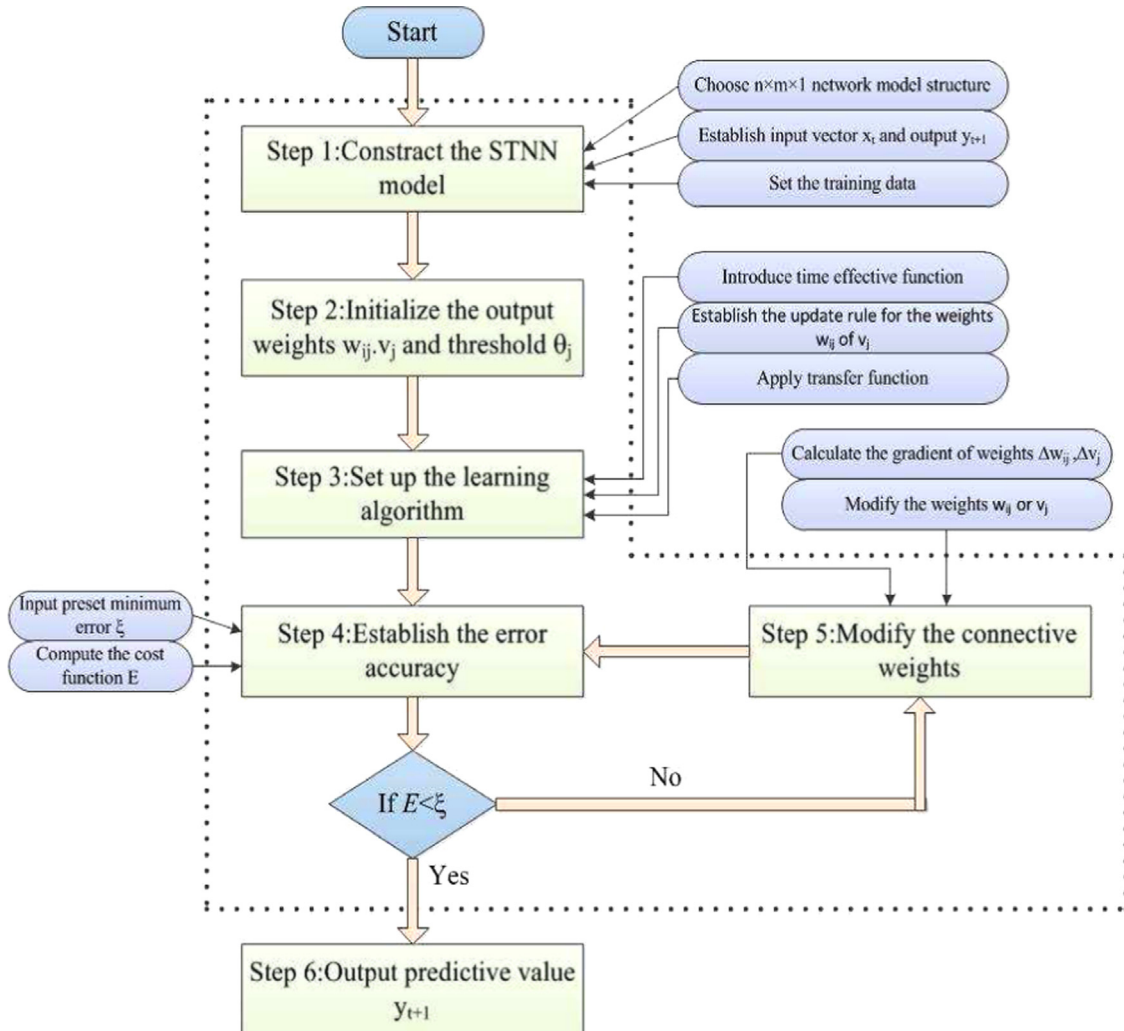


Fig. 2. Training algorithm procedures of stochastic time effective neural network.

water quality monitoring stations located in the main stem of the LSJR. The outcome showed that the number of monitoring stations can be reduced from 22 to 19. Yang et al. [33] built a prediction model for the occurrence of paddy stem borer based on BP neural network, and they applied the PCA approach to create fewer factors to be the input variables for the neural network. Because the essence of PCA is the rotation of space coordinates that does not change the data structure, the obtained PCs are the linear combination of variables, reflect the original information to the greatest degree, and are uncorrelated with each other. The specific steps are as follows: assume the data matrix with m variables, X_1, X_2, \dots, X_m , n times observations:

$$X = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1m} \\ X_{21} & X_{22} & \cdots & X_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{nm} \end{bmatrix}. \quad (11)$$

Firstly, we normalize the original data by using the following method: $Y_{ti} = (X_{ti} - \bar{X}_i) / S_i$, where $\bar{X}_i = (1/n) \sum_{t=1}^n X_{ti}$ and $S_i = \sqrt{(1/(n-1)) \sum_{t=1}^n (X_{ti} - \bar{X}_i)^2}$. For convenience, the normalized Y_{ti} is still denoted as X_{ti} . Let $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m \geq 0$ be the eigenvalues of covariance matrix of normalized data. Also let $\alpha_1, \alpha_2, \dots, \alpha_m$ be the corresponding eigenvector, the i -th principal component is such that $F_i = \alpha_i^T X$, where $i = 1, 2, \dots, m$. Generally, $\lambda_k / \sum_{i=1}^m \lambda_i$ is called the contribution rate of k -th principal component, $\sum_{i=1}^k \lambda_i / \sum_{i=1}^m \lambda_i$ is called the cumulative contribution rate of first k principal components. If the cumulative contribution rate exceeds 85%, the first k principal components contain the most information of m original variables.

The neural network model requires that the input variables should have poor correlation. Because the strong correlation between input variables implies that they carry more repeated information, and it may increase the computational complexity and reduce the prediction accuracy of the model. The concept of the PCA-STNN forecasting model is explained as follows [19,33]. To improve the common PCA-BPNN model, we use PCA method to extract the principal components from the input data of stochastic time effective neural network, then conduct the principal components as the input of the STNN model, that is the method of PCA-STNN model. We introduce the indexes SSE and HS300 to illustrate how to extract the principal components from the input data using the method of PCA. The network inputs include six variables: daily open price, daily closing price, daily highest price, daily lowest price, daily volume and daily turnover (the amount of daily trading money). The network outputs include the closing price of the next trading day. These six input values including daily open price, daily closing price, daily highest price, daily lowest price, daily volume and daily turnover are denoted as x_1, x_2, x_3, x_4, x_5 and x_6 respectively. Tables 1 and 2 respectively exhibit the input datum correlation coefficients of SSE and HS300. From Tables 1 and 2, we can see that the correlation between the six time series clearly, it means that they contain more repeated information.

Table 1
Input value correlation coefficients of SSE.

Input value	x_1	x_2	x_3	x_4	x_5	x_6
x_1	1.000	0.991	0.997	0.997	0.430	0.621
x_2	0.911	1.000	0.997	0.996	0.451	0.636
x_3	0.997	0.997	1.000	0.996	0.452	0.638
x_4	0.997	0.996	0.996	1.000	0.432	0.622
x_5	0.430	0.451	0.452	0.432	1.000	0.963
x_6	0.621	0.636	0.638	0.622	0.963	1.000

Table 2
Input value correlation coefficients of HS300.

Input value	x_1	x_2	x_3	x_4	x_5	x_6
x_1	1.000	0.986	0.995	0.995	0.296	0.443
x_2	0.986	1.000	0.995	0.993	0.334	0.476
x_3	0.995	0.995	1.000	0.994	0.336	0.475
x_4	0.995	0.993	0.994	1.000	0.296	0.444
x_5	0.296	0.334	0.336	0.296	1.000	0.967
x_6	0.443	0.476	0.475	0.444	0.967	1.000

Table 3
The SSE/HS300 PCA results of six time series.

Com.	SSE			HS300		
	Eige.	Con.r	C-con.r	Eige.	Con.r	C-con.r
1	4.807	80.117	80.117	4.469	74.479	74.479
2	1.168	19.464	99.581	1.491	24.851	99.330
3	0.013	0.214	99.794	0.022	0.367	99.697
4	0.008	0.139	99.934	0.013	0.214	99.911
5	0.003	0.057	99.991	0.004	0.075	99.985
6	0.001	0.009	100.000	0.001	0.015	100.000

Table 3 shows the SSE PCA and the HS300 PCA results of six time series, where Com., Con.r, C-con.r and Eige. denote the abbreviations of Component, Contribution rate, Cumulative contribution rate and Eigenvalue respectively. The table indicates that the cumulative contribution rates of the first two PCs exceed 99%, namely the first two PCs contain 99% information of the original data. These two PCs are recorded as F_1 and F_2 , which are conducted as the input data of the PCA-STNN (PCA-BPNN) model instead of the original data.

3. Forecasting and statistical analysis of stock price

3.1. Selecting and preprocessing of the data

To evaluate the performance of the proposed PCA-STNN forecasting model, we select the daily data from Shanghai Stock Exchange Composite Index (SSE), Hong Kong Hang Seng 300 Index (HS300), Dow Jones Industrial Average Index (DJIA), and Standard & Poor's 500 Index (S&P500) to analyze the forecasting models by comparison. The SSE data cover the time period from 21/06/2006 up to 31/08/2012, which accounts to 1513 data points. The HS300 is from 04/01/2005 to 31/08/2012 with 1863 data points. The data of the S&P500 used in this paper is from 04/08/2006 to 31/08/2012 with 1539 data points, while the DJIA is totally 1532 data points from 04/08/2006 to 31/08/2012. Usually, the non-trading time periods are treated as frozen such that we adopt only the time during trading hours. To reduce the impact of noise in the financial market and finally lead to a better prediction, the collected data should be properly adjusted and normalized at the beginning of the modelling. There are different normalization methods that are tested to improve the network training [34,35], which include "the normalized data in the range of $[0, 1]$ " in the following equation, which is also adopted in this work:

$$S(t)' = \frac{S(t) - \min S(t)}{\max S(t) - \min S(t)} \quad (12)$$

where the minimum and maximum values are obtained on the training set during the training process. In order to obtain the true

value after the forecasting, we can revert the output variables as

$$S(t) = S(t)'(\max S(t) - \min S(t)) + \min S(t). \quad (13)$$

Then the data is passed to the network as the nonstationary data.

3.2. Forecasting by PCA-STNN model

In the PCA-STNN model, after analyzing the six original time series by using PCA method (SPSS software is selected to achieve the goal on the step of PCA in this paper), we obtain two PCs (see Section 2.3), the number of neural nodes in the input layer is 2 which corresponds to the two PCs, the number of neural nodes in the hidden layer is 9 [18], the output layer is 1, then the architecture is $2 \times 9 \times 1$. It is note worthy that the data points for these four time series are not the same, the lengths of training data and testing data are also set differently. We choose about 15% of the data as testing set. Then, the training set for the SSE is from 21/06/2006 to 20/10/2011 with totally 1300 data, while that for the HS300 is from 04/01/2005 to 10/03/2011 with data of 1500. The training data for S&P500 is 1300 from 04/08/2006 to 21/09/2011, and for DJIA is 1300 from 04/08/2006 to 30/09/2011. The rest of the data is defined as the testing set. To avoid the impacts of initial parameters on the proposed time effective function, the maximum training iterations number is pre-set $K=200$. After many times experiments, different

datasets have different learning rates η , for SSE $\eta = 0.03$, for HS300, S&P500 and DJIA, $\eta = 0.01, 0.003$ and 0.005 respectively. And the predefined minimum training threshold is $\xi = 10^{-5}$. When using the PCA-STNN model to predict the daily closing price of stock index, we assume that $\mu(t)$ (the drift function) and $\sigma(t)$ (the volatility function) are as follows:

$$\mu(t) = \frac{1}{(c-t)^2}, \quad \sigma(t) = \left[\frac{1}{N-1} \sum_{i=1}^N (x - \bar{x})^2 \right]^{1/2} \quad (14)$$

where c is the parameter which is equal to the number of sample in the datasets, and \bar{x} is the mean of the sample data. Then the corresponding cost function can be written as

$$E = \frac{1}{N} \sum_{n=1}^N E(t_n) \\ = \frac{1}{2N} \sum_{n=1}^N \frac{1}{\beta} \exp \left\{ \int_{t_0}^{t_n} \frac{1}{(c-t)^2} dt + \int_{t_0}^{t_n} \left[\frac{1}{N-1} \sum_{i=1}^N (x - \bar{x})^2 \right]^{1/2} dB(t) \right\} (d_{t_n} - y_{t_n})^2. \quad (15)$$

Fig. 3 shows the predicting results of training and test data for SSE, HS300, S&P500 and DJIA with the PCA-STNN model correspondingly. The curves of the actual data and the predictive data are intuitively very approximating.

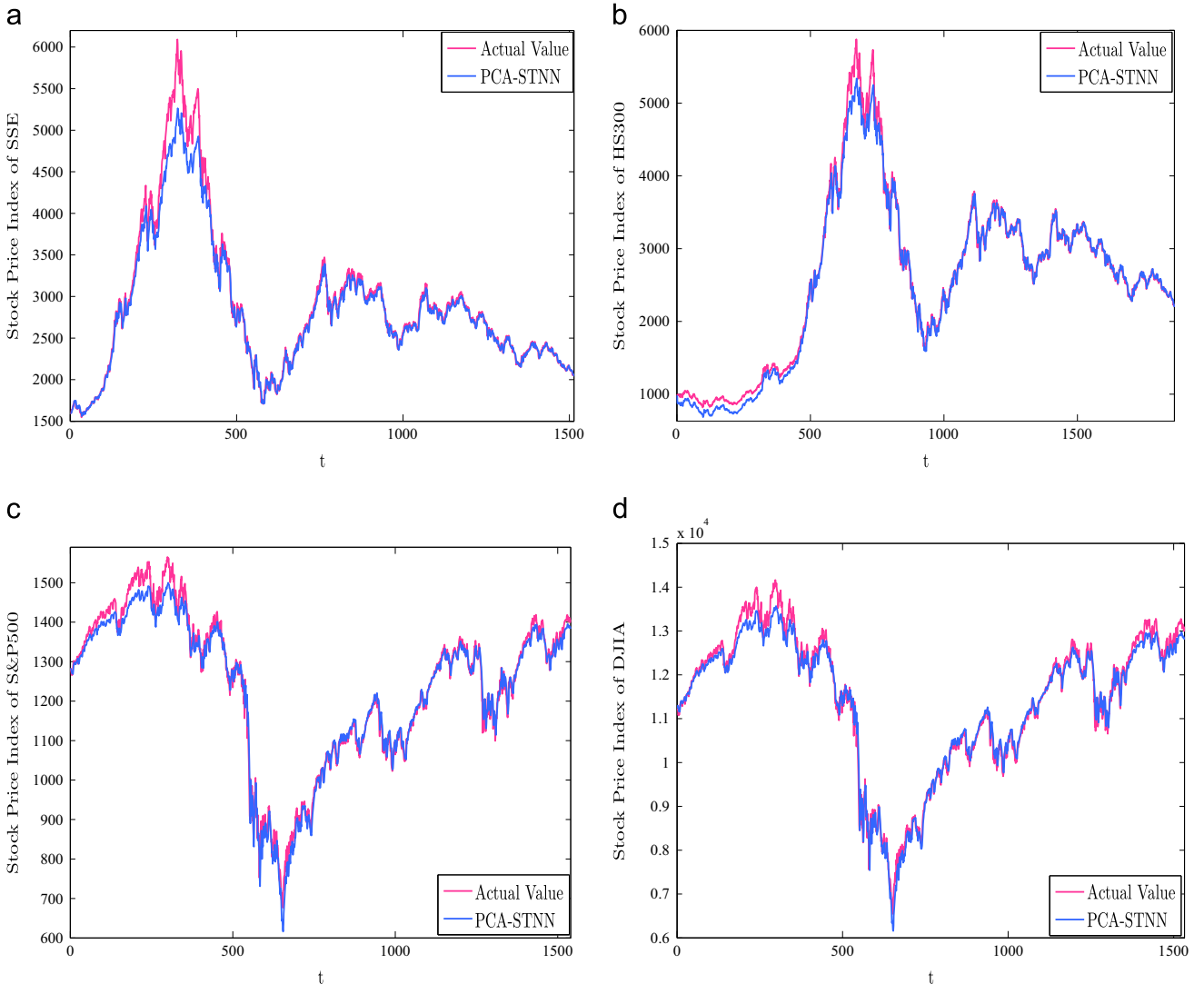


Fig. 3. Comparisons of the predictive data and the actual data for the forecasting models. (a) SSE, (b) HS300, (c) S&P500, and (d) DJIA.

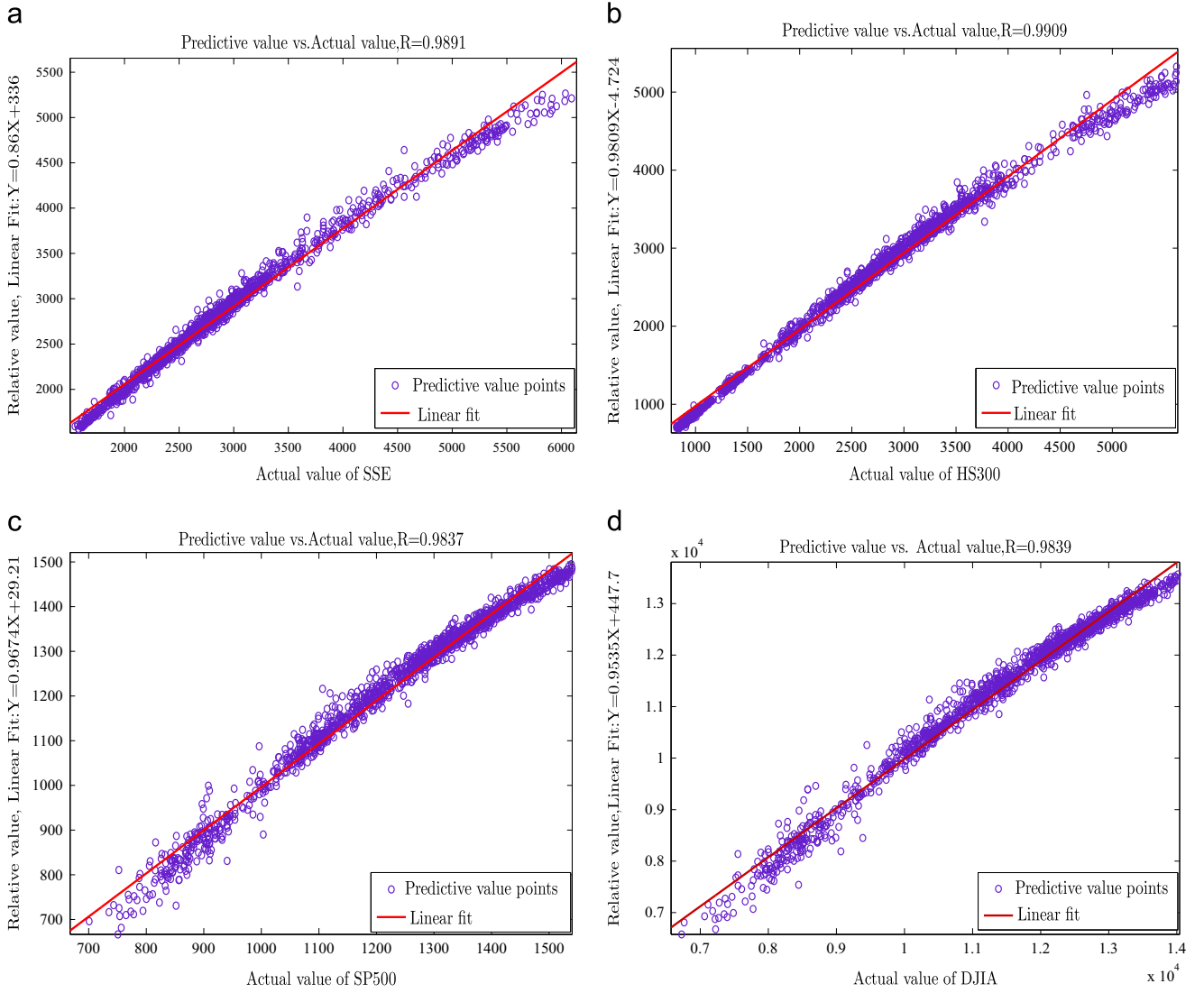


Fig. 4. The comparisons and linear regressions of the actual data and the predictive value for (a) SSE, (b) HS300, (c) S&P500, and (d) DJIA.

The plots of the actual and the predictive data for these four price sequences are shown in Fig. 4. Through the linear regression analysis, we make a comparison of the predictive value of the PCA-STNN model with the actual value. It is known that the linear regression can be used to fit a predictive model to an observed dataset of Y and X . The linear equations of SSE, HS300, S&P500 and DJIA are exhibited respectively in Fig. 4(a), (b), (c) and (d). We can observe that all the slopes of the linear equations for them are drawing near to 1, which implies that the predictive values and the actual values are not deviating too much.

Set the actual data as x -axis, the predictive data as y -axis, and the linear equation is $y = ax + b$. A valuable numerical measure of association between two variables is the correlation coefficient R . Table 4 shows the values of a , b and R for the above indices. R is given as follows:

$$R = \frac{\sum_{i=1}^N (d_t - \bar{d}_t)(y_t - \bar{y}_t)}{\sqrt{\sum_{i=1}^N (d_t - \bar{d}_t)^2 \sum_{i=1}^N (y_t - \bar{y}_t)^2}} \quad (16)$$

where d_t is the actual value, y_t is the predict value, \bar{d}_t is the mean of the actual value, \bar{y}_t is the mean of the predict value and N is the total number of the data.

Table 4

Linear regression parameters of market indices.

Parameter	SSE	SH300	S&P500	DJIA
a	0.86	0.9809	0.9674	0.9535
b	336	-4.724	29.21	447.7
R	0.9891	0.9909	0.9837	0.9839

3.3. Comparisons of forecasting results

In the BPNN and the STNN model, the network inputs include four kinds of data: daily open price, daily closing price, daily highest price, and daily lowest price. The network output is the closing price of the next trading day. The number of neural nodes in the hidden layer is 8. In the stock markets, the practical experience shows us that the above four kinds of data of the last trading day are very important indicators when we predict the closing price of the next trading day at the technical level. The architectures of BPNN model and STNN model both are $4 \times 8 \times 1$. For the common PCA-BPNN model, we choose the same structure as the PCA-STNN model, namely $2 \times 9 \times 1$. Fig. 5(a)–(d) shows the predicting values of the four indexes on the test set. From these plots, the predicting values of

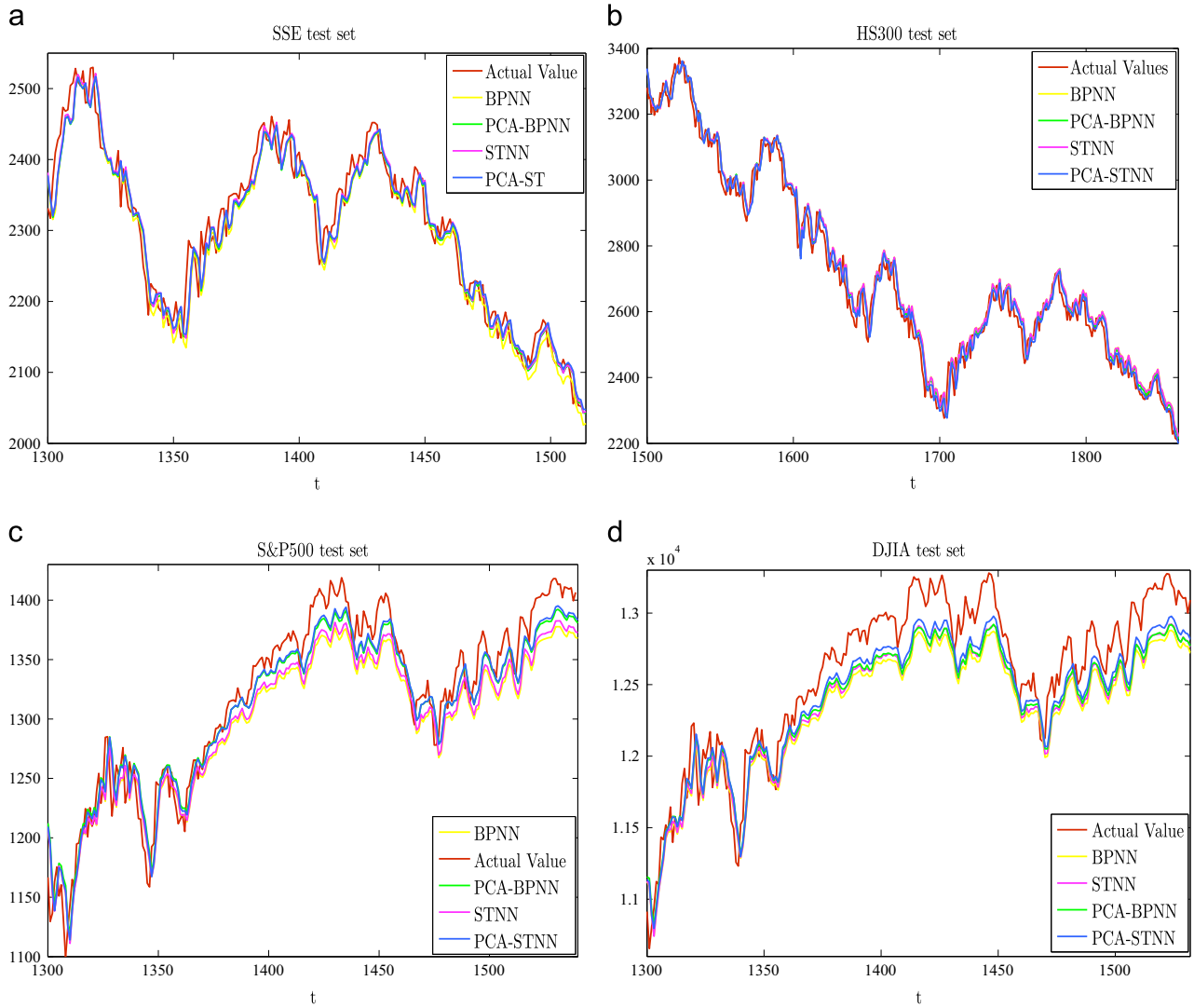


Fig. 5. Predictive values on the test set for (a) SSE, (b) HS300, (c) S&P500, (d) and DJIA.

the PCA-STNN model are more close to the actual values in intuitive sense.

To compare the forecasting performance of four considered forecasting models, we use the following error evaluation criteria: the mean absolute error (MAE), the root mean square error (RMSE) and the mean absolute percentage error (MAPE); the corresponding definitions are given as follows:

$$MAE = \frac{1}{N} \sum_{t=1}^N |d_t - y_t|, \quad RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N (d_t - y_t)^2} \quad (17)$$

$$MAPE = 100 \times \frac{1}{N} \sum_{t=1}^N \left| \frac{d_t - y_t}{d_t} \right| \quad (18)$$

where d_t and y_t are the actual value and the predict value at time t respectively. N is the total number of the data. Noting that MAE, RMSE, and MAPE are measures of the deviation between the prediction values and the actual values, the prediction performance is better when the values of these evaluation criteria are smaller. However, if the results are not consistent among these criteria, we choose the MAPE as the benchmark since MAPE is relatively more stable than other criteria [36].

Fig. 6(a), (b), (c) and (d) respectively shows the forecasting results of SSE, HS300, S&P500 and DJIA for four forecasting models.

The empirical research shows that the proposed PCA-STNN model has the best performance, the PCA-BPNN and the STNN both outperform the common BPNN model. In years 2007 and 2008, China's stock markets appeared with large fluctuations which are reflected in Fig. 6. At the same time, we can see that the large fluctuation period forecasting is relatively not accurate from these four models. When the stock market is relatively stable, the forecasting result is nearer to the actual value. Compared with the BPNN, the STNN and the PCA-BPNN models, the forecasting results are also presented in Table 5, where the MAPE(100) stands for the MAPE of the latest 100 days in the testing data. Table 5 shows that the evaluation criteria by the PCA-STNN model are almost smaller than those by other models. So from Table 5 and Fig. 6, these can conclude that the proposed PCA-STNN model is better than the other three models. In Table 5, the evaluation criteria by the STNN model are almost smaller than those by BPNN for four considered indexes. This shows that the effect of price fluctuation forecasting of STNN model is superior to that of BPNN model for these four indexes. Besides, the values of MAPE(100) are smaller than those of MAPE in all stock indexes. Therefore, the short-term prediction outperforms the long-term prediction.

Furthermore, there are more advanced nonlinear methods that recently have been frequently applied with success. The ability of support vector machine (SVM) to solve nonlinear regression

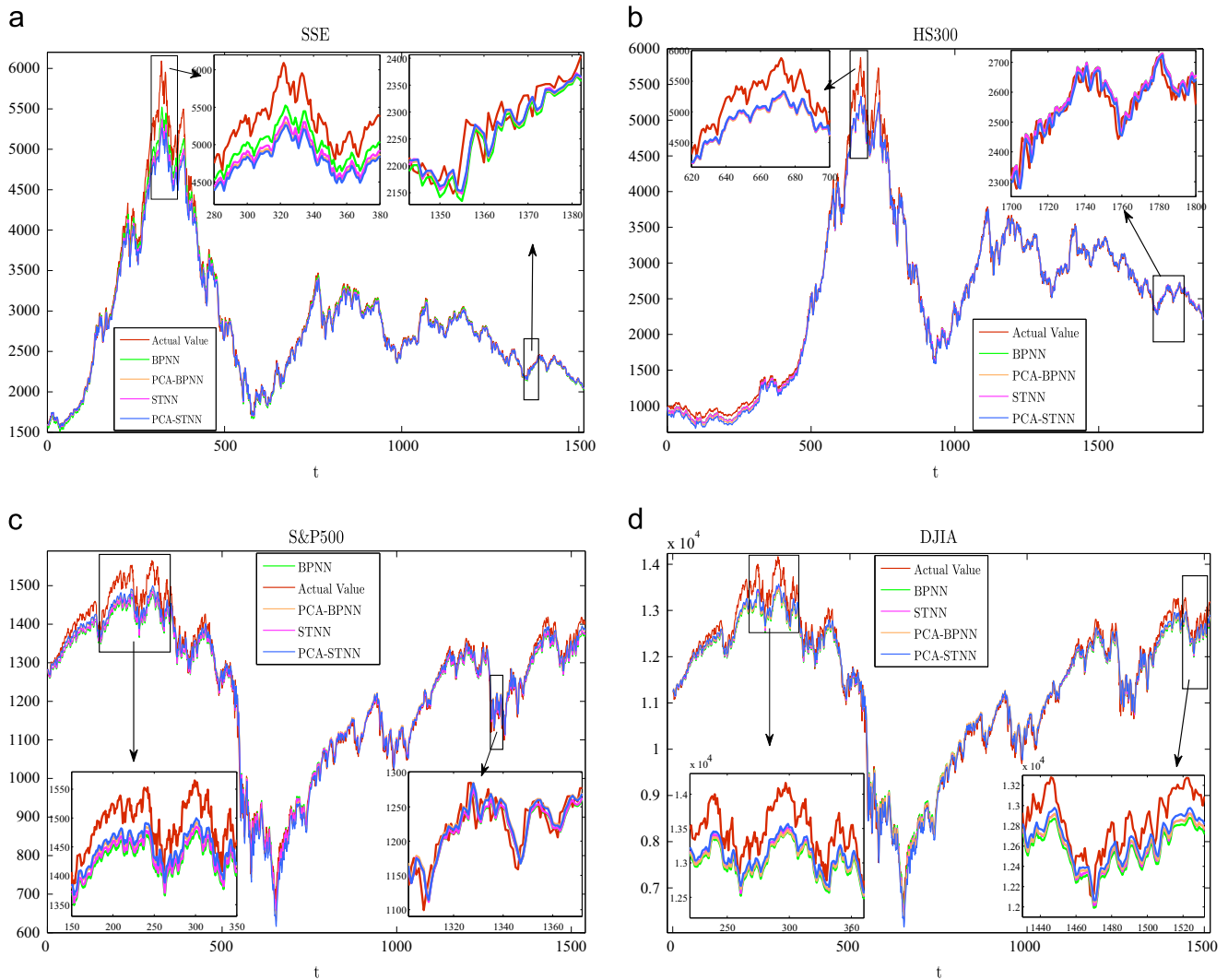


Fig. 6. Comparisons of the actual data and the predictive data for (a) SSE, (b) HS300, (c) S&P500, and (d) DJIA.

Table 5
Comparisons of indexes' predictions for different forecasting models.

Index errors	BPNN	STNN	PCA-BPNN	PCA-STNN	SVM
SSE					
MAE	24.4385	22.8295	22.4485	22.0844	27.8603
RMSE	30.8244	29.0678	28.6826	28.2975	34.5075
MAPE	1.0579	0.9865	0.9691	0.9540	1.2190
MAPE(100)	1.2385	0.9677	1.1703	1.1490	1.3427
HS300					
MAE	34.6557	34.4337	32.7882	31.0770	42.6880
RMSE	42.8477	42.6034	40.7438	39.0088	50.9338
MAPE	1.3015	1.2924	1.2256	1.1557	1.6779
MAPE(100)	0.9839	0.9846	0.9781	0.9742	1.6393
S&P500					
MAE	24.7591	22.1833	16.8138	15.5181	22.9334
RMSE	28.1231	25.5039	20.5378	19.2467	25.9961
MAPE	1.8607	1.6725	1.2820	1.1872	1.7722
MAPE(100)	1.5392	1.4657	1.2814	1.2547	1.6869
DJIA					
MAE	258.4801	230.7871	220.9163	192.1769	278.2667
RMSE	286.6511	258.3063	250.4738	220.4365	302.7930
MAPE	2.0348	1.8193	1.7404	1.5183	2.2677
MAPE(100)	1.7003	1.5713	1.4678	1.3625	2.2707

estimation problems makes SVM successful in time series forecasting [36–38]. It estimates the regression using a set of linear functions that are defined in a high-dimensional feature space and carries out the regression estimation by risk minimization. Here we apply the SVM to forecast the indexes of SSE, HS300, S&P500 and DJIA, where the radial basis function is selected as the kernel function of SVM. In Table 5, the evaluation criteria by SVM are larger than PCA-STNN. The experimental results show that PCA-STNN outperforms the other methods, which further indicates that PCA-STNN could be considered as a promising methodology for financial time-series forecasting.

4. Forecasting and error estimate of stock return

The analysis and the forecast of fluctuations of returns have long been a focus of economic research for a more clear understanding of mechanism and characteristics of financial markets [39–45]. In the following, we consider the statistical behaviors of price returns and relative errors of the PCA-STNN forecasting results. Let $S(t)$ ($t = 1, 2, \dots$) denote the price sequences of SSE, HS300, S&P500 and DJIA at time t , then the formula of stock logarithmic return and

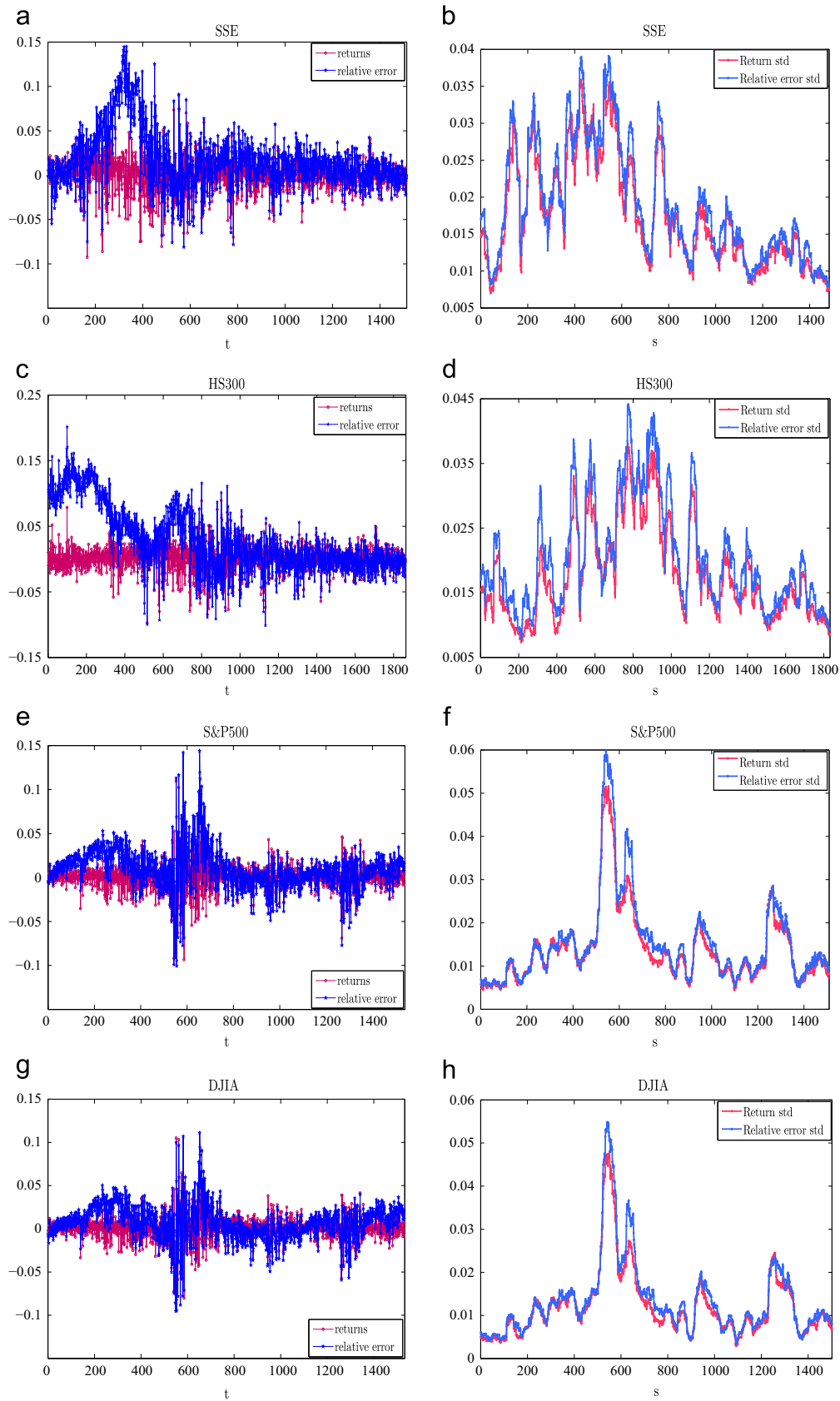


Fig. 7. (a, c, e, g) Daily returns and relative errors of forecasting results from the PCA-STNN model. (b, d, f, h) Corresponding standard deviations of daily returns and relative errors for the PCA-STNN model.

relative error is given as follows:

$$r(t) = \ln S(t+1) - \ln S(t), \quad e(t) = \frac{d_t - y_t}{d_t} \quad (19)$$

where d_t and y_t denote the actual value and the predict value respectively at the time t , $t = 1, 2, \dots$. In Fig. 7(a, c, e, g), we consider the fluctuations of each index daily returns and relative errors of forecasting results from the PCA-STNN model. It can be seen that there are some points with large relative errors of forecasting results in four models. And the empirical results show that these points appear basically in the place where there are large return volatilities. This indicates that the predicting results to the points where the prices fluctuate violently are relatively not satisfactory. Fig. 7(b, d, f, h) better illustrates the relationships between the daily returns of each index and the relative errors of forecasting results from the PCA-STNN model by calculating the standard deviation of the two variables respectively. We adopt the sliding analysis method, specifically, we select every 30 data to calculate the standard deviation, that is $s, s+1, \dots, s+29$ for $s = 1, \dots, N-30$ (N is the sample size). During the large fluctuation periods (or parts), the corresponding standard deviations of daily returns and relative errors are both larger than those of the small fluctuation periods. In a conclusion, the daily returns of each index and the relative errors of forecasting results from the PCA-STNN model have shown the similar fluctuation trend, the small fluctuation leads to the small relative errors, and the large fluctuation leads to the large relative errors.

5. Conclusion

In the present paper, we introduce the stochastic time effective neural network with principal component analysis model to forecast the indexes of SSE, HS300, S&P500, and DJIA. We take the proposed model compared with BPNN, STNN and PCA-BPNN forecasting models. Empirical examinations of predicting precision for the price time series (by the comparisons of predicting measures as MAE, RMSE, MAPE and MAPE(100)) show that the proposed neural network model has the advantage of improving the precision of forecasting, and the forecasting of this proposed model much approaches to the actual financial market movements. By calculating the standard deviations of the daily returns of each index and the relative errors of forecasting results, it can conclude that the daily returns and the relative errors have the similar fluctuation trend patterns. We hope that the proposed model can make some beneficial contributions to ANN research and application in the financial time series forecasting.

Acknowledgments

The authors were supported in part by National Natural Science Foundation of China Grant nos. 71271026 and 10971010.

References

- [1] J.R. Noriega, H. Wang, A direct adaptive neural-network control for unknown-nonlinear systems and its application, *IEEE Trans. Neural Netw.* 9 (1998) 27–34.
- [2] D. Enke, S. Thawornwong, The use of data mining and neural networks for forecasting stockmarket returns, *Expert Syst. Appl.* 29 (2005) 927–940.
- [3] Y.J. Liu, C. Chen, G.X. Wen, S. Tong, Adaptive neural output feedback tracking control for a class of uncertain discrete-time nonlinear systems, *IEEE Trans. Neural Netw.* 22 (2011) 1162–1167.
- [4] F. Wang, J. Wang, Statistical analysis and forecasting of return interval for SSE and model by lattice percolation system and neural network, *Comput. Ind. Eng.* 62 (2012) 198–205.
- [5] M.M. Polycarpou, Stable adaptive neural control scheme for nonlinear systems, *IEEE Trans. Autom. Control* 41 (1996) 447–451.
- [6] T. Chenoweth, Z. Obradovic, A multi-component nonlinear prediction system for the S&P 500 index, *Neurocomputing* 10 (1996) 275–290.
- [7] F. Fernandez-Rodriguez, C. Gonzalez-Martel, S. Sosvilla-Rivebo, On the profitability of technical trading rules based on artificial neural networks: evidence from the Madrid stock market, *Econ. Lett.* 69 (2000) 89–94.
- [8] M. Ghiassi, H. Saidane, D.K. Zimbra, A dynamic artificial neural network model for forecasting time series events, *Int. J. Forecast.* 21 (2005) 341–362.
- [9] H.L. Niu, J. Wang, Volatility clustering and long memory of financial time series and financial price model, *Digit. Signal Process.* 23 (2013) 489–498.
- [10] T.H. Roh, Forecasting the volatility of stock price index, *Expert Syst. Appl.* 33 (2007) 916–922.
- [11] A. Lendasse, E. de Bodt, V. Wertz, M. Verleysen, Non-linear financial time series forecasting application to the Bel 20 stock market index, *Eur. J. Econ. Soc. Syst.* 14 (2000) 81–91.
- [12] R. Pino, J. Parreno, A. Gomez, P. Priore, Forecasting next-day price of electricity in the Spanish energy market using artificial neural networks, *Eng. Appl. Artif. Intell.* 21 (2008) 53–62.
- [13] A. Shtub, P. Versano, Estimating the cost of steel pipe bending, a comparison between neural networks and regression analysis, *J. Prod. Econ.* 62 (1999) 201–207.
- [14] A. Fagner, N. Cristiane, E. Luis, Applying artificial neural networks to prediction of stock price and improvement of the directional prediction index—case study of PETRA, Petrobras, Brazil, *Expert Syst. Appl.* 40 (2013) 7596–7606.
- [15] G. Armano, M. Marchesi, A. Murru, A hybrid geneticneural architecture for stock indexes forecasting, *Inf. Sci.* 170 (2005) 3–33.
- [16] T.S. Wang, J. Wang, J.H. Zhang, W. Fang, Voter interacting systems applied to Chinese stock markets, *Math. Comput. Simul.* 81 (2011) 2492–2506.
- [17] F.J. Liu, J. Wang, Fluctuation prediction of stock market index by Legendre neural network with random time strength function, *Neurocomputing* 83 (2012) 12–21.
- [18] H.F. Liu, J. Wang, Integrating independent component analysis and principal component analysis with neural network to predict Chinese stock market, *Math. Probl. Eng.* (2011) Article ID 382659.
- [19] Z. Liao, J. Wang, Forecasting model of global stock index by stochastic time effective neural network, *Expert Syst. Appl.* 37 (2010) 834–841.
- [20] P.P. Balestrassi, E. Popova, A.P. Paiva, J.W.M. Lima, Design of experiments on neural networks training for nonlinear time series forecasting, *Neurocomputing* 72 (2009) 1160–1178.
- [21] W. Fang, J. Wang, Statistical properties and multifractal behavior of market returns by Ising dynamic systems, *Int. J. Mod. Phys. C* 23 (2012) 1250023.
- [22] B.Y. Lu, Y.L. Chen, Y.Y. Li, The forecast of the pre-processing data with BP neural network principal component analysis, *Sci. Technol. Inf.* 17 (2009) 29–30.
- [23] K.Z. Mohammad, Principal component analysis (PCA) for estimating chlorophyll concentration using forward and generalized regression neural networks, *Appl. Artif. Intell.* 28 (2014) 16–29.
- [24] S. Lee, Comparative analyses of anthropometry associated with overweight and obesity: PCA and ICA approaches, *Theor. Issues Ergon. Sci.* 9 (2008) 441–475.
- [25] D. Shi, F. Tsung, Modelling and diagnosis of feedbackcontrolled processes using dynamic PCA and neural networks, *Int. J. Prod. Res.* 41 (2003) 365–379.
- [26] S. Chitroub, Neural network model for standard PCA and its variants applied to remote sensing, *Int. J. Remote Sens.* 26 (2005) 2197–2218.
- [27] D. Olson, C. Mossman, Neural network forecasts of canadian stock returns using accounting ratios, *Int. J. Forecast.* 19 (2003) 453–465.
- [28] H. Demuth, M. Beale, *Network Toolbox: For Use with MATLAB*, The Math Works, Natick, Mass, USA, 5th edition, 1998.
- [29] D.E. Rumelhart, J.L. McClelland, *Distributed Processing: Explorations in the Micro-Structure of Cognition*, The MIT Press, Cambridge, UK, 1986.
- [30] J.E. Jackson, *A User's Guide to Principal Component Analysis*, John Wiley & Sons, New York, 1991.
- [31] M. Tripathy, Power transformer differential protection using neural network principal component analysis and radial basis function neural network, *Simul. Model. Pract. Theory* 18 (2010) 600–611.
- [32] Y. Ouyang, Evaluation of river water quality monitoring stations by principal component analysis, *Water Res.* 39 (2005) 2621–2635.
- [33] L.N. Yang, L. Peng, L.M. Zhang, L.L. Zhang, S.S. Yang, A prediction model for population occurrence of paddy stem borer (*Scirpophaga incertulas*), based on back propagation artificial neural network and principal component analysis, *Comput. Electron. Agric.* 68 (2009) 200–206.
- [34] D.K. Chaturvedi, P.S. Satsangi, P.K. Kalra, Effect of Different Mappings and Normalization of Neural Network Models, vol. 9, Indian institute of Technology, Kanpur, 1996, pp. 377–386.
- [35] S. Makridakis, Accuracy measures: theoretical and practical concerns, *Int. J. Forecast.* 9 (1993) 527–529.
- [36] S. Lahmiri, A comparison of PNN and SVM for stock market trend prediction using economic and technical information, *Int. J. Comput. Appl.* 29 (2011) 24–30.
- [37] Z.Q. Zhang, Q. Zhao, The application of SVMs method on exchange rates fluctuation, *Discr. Dyn. Nat. Soc.* (2009) Article ID 250206.
- [38] K. Kim, Financial time series forecasting using support vector machines, *Neurocomputing* 55 (2003) 307–319.
- [39] J. Wang, Q.Y. Wang, J.G. Shao, Fluctuations of stock price model by statistical physics systems, *Math. Comput. Model.* 51 (2010) 431–440.
- [40] J.H. Zhang, J. Wang, Modeling and simulation of the market fluctuations by the finite range contact systems, *Simul. Model. Pract. Theory* 18 (2010) 910–925.
- [41] R.N. Mantegna, H.E. Stanley, *An Introduction to Econophysics: Correlations and Complexity in Finance*, Cambridge University Press, Cambridge, 2000.

- [42] D. Xiao, J. Wang, Modeling stock price dynamics by continuum percolation system and a relevant complex systems analysis, *Physica A* 391 (2012) 4827–4838.
- [43] Y. Yu, J. Wang, Lattice oriented percolation system applied to volatility behavior of stock market, *J. Appl. Stat.* 39 (2012) 785–797.
- [44] F. Black, M. Scholes, The pricing of options and corporate liabilities, *J. Polit. Econ.* 81 (1973) 637–654.
- [45] X. Gabaix, P. Gopikrishnan, V. Plerou, H.E. Stanley, A theory of power-law distributions in financial market fluctuations, *Nature* 423 (2003) 267–270.



Jie Wang is a Doctor from Institute of Financial Mathematics and Financial Engineering at Beijing Jiaotong University, China. Her research interests are in Stochastic Systems, Statistical Physics, Stochastic Processes, Artificial Intelligence, Neural Networks, Modeling and Computer Simulation, Probability Theory and Statistics, Financial Mathematics, Financial Engineering, and Financial Statistics.



Jun Wang is a Full Professor in School of Science, and Director of Institute of Financial Mathematics and Financial Engineering at Beijing Jiaotong University, China. He received his B.Sc. degree in Mathematics from Beijing Normal University, and received his Ph.D. degree in Probability and Statistics from Kobe University of Japan. His research interests include Large Scale Interacting Systems, Statistical Physics, Stochastic Systems, Stochastic Processes, Artificial Intelligence, Neural Networks, Modeling and Computer Simulation, Probability Theory and Statistics, Financial Mathematics, Financial Engineering, and Financial Statistics.