Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 436795, 11 pages
http://dx.doi.org/10.1155/2013/436795

Research Article

Volatility Degree Forecasting of Stock Market by Stochastic Time Strength Neural Network

Haiyan Mo and Jun Wang

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

Correspondence should be addressed to Jun Wang; wangjun@center.njtu.edu.cn

Received 18 April 2013; Revised 2 September 2013; Accepted 2 September 2013

Academic Editor: Wei-Chiang Hong

In view of the applications of artificial neural networks in economic and financial forecasting, a stochastic time strength function is introduced in the backpropagation neural network model to predict the fluctuations of stock price changes. In this model, stochastic time strength function gives a weight for each historical datum and makes the model have the effect of random movement, and then we investigate and forecast the behavior of volatility degrees of returns for the Chinese stock market indexes and some global market indexes. The empirical research is performed in testing the prediction effect of SSE, SZSE, HSI, DJIA, IXIC, and S&P 500 with different selected volatility degrees in the established model.

1. Introduction

Financial analysis of fluctuations of financial market is an active topic in economic research, with applications to the prediction of interest rates, foreign currency risk, stock market volatility, and so forth [1–8]. With the progress of globalized security markets, forecasting stock market volatility has become a significant financial subject, which attracts much increasing attention [9–11]. A key element of financial planning and financial forecasting is the ability to construct models showing the interrelatedness of financial data. Models showing correlation or causation between variables can be employed to improve financial decision making. The popular model of financial time series analysis is the artificial neural network (ANN), which is composed of artificial neurons or nodes. ANN is nonlinear statistical data modeling or decision making method; it can be used to model complex relationships between inputs and outputs or to find patterns in data. Generally, stock price changes can be regarded as a random time sequence with noise and artificial neural network and as a nonparametric method with large-scale processing elements operating in parallel that depend on their own intrinsic link data and can forecast future behaviors by learning the pattern of market variables without any strict theoretical assumption. ANN possesses data-driven, self-learning, and self-adaptive abilities and has strong antijamming capabilities. ANN has been widely used in the financial fields such as prediction of stock or option price, exchange rate, and risk analysis [12–19].

In the real markets, the noise of financial time series is usually caused by large volatilities, and it is hard to reflect the market variables directly into the model without any assumptions. Therefore, making accurate forecast is a challenging task due to the inherently noisy and nonstationary nature of stock price. To improve predicting precision, various network architectures and learning algorithms have been developed in the literature [20–24]. 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 [25–28]. The present work applies MLP with a backpropagation algorithm and stochastic time strength function to develop a stock price volatility forecasting model; a stochastic time strength neural network (STNN) and the corresponding learning algorithm are presented. The motivation of modeling the STNN is that the data in the data training set should be time variant, reflecting the different behavior patterns of the markets at different times; if all the data are used to train the network equivalently, the network system may not be consistent with the evolvement of the stock markets. It is difficult to use the historical data of the past to reflect the current stock markets; however, if only the recent data are

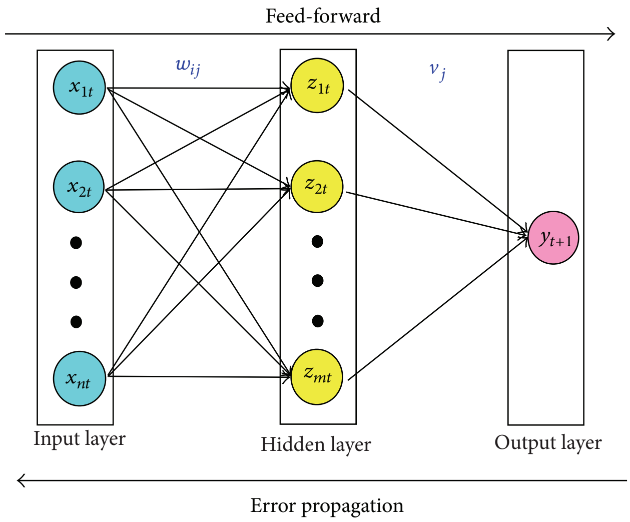


Figure 1: A three-layer multi-input neural network with one output.

selected, a lot of useful information, which the early data hold, will be lost. From this consideration, the historical data are given weights depending on their time, and the Brownian motion is introduced in the time strength function in order to make the model have the effect of random movement while maintaining the original trend. In the present paper, we perform the empirical research focusing on the volatility degree's behaviors of stock returns, and a threshold value $q(\geq 0)$ is introduced to correspond to the volatility degree. For different threshold values, the volatility degree forecasting is made by STNN model. The empirical data are selected from the global stock indexes, including Shanghai Stock Exchange (SSE) Composite Index, Shenzhen Stock Exchange (SZSE) Component Index, Hong Kong Hang Seng Index (HSI), Dow Jones Industrial Average Index (DJIA), Nasdaq Composite Index (IXIC), Standard & Poor's 500 Index (S&P 500), and Japan Nikkei 225 Index (N225).

2. Methodology for STNN Model

The structure of ANN plays an important role in its performance. In Figure 1, we introduce a three-layer multi-input neural network, in which a stochastic time strength function is applied to minimize the error between the network's prediction and the actual target. The architecture consists of a hidden layer of neurons with nonlinear activation functions and an output layer of neurons with linear activation functions. x_{it} $(i = 1, \dots, n)$ represents the input variable at time t; z_{jt} $(j = 1, \dots, m)$ represents the output of hidden layer neurons at time t; and y_{t+1} represents the output of the network at time $t+1$. 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 calculated by the following equation:

$$\mathrm{net}_{jt} = \sum_{i=1}^{n} w_{ij} x_{it} - \theta_j, \quad i = 1, \dots, n. \tag{1}$$

The output of hidden neuron is given by

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

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

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

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

The backpropagation algorithm has emerged as one of the most widely used learning procedures for multilayer networks [7, 29, 30]. That is a supervised learning algorithm which minimizes the global error E_k 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

$$E\left(t_n\right) = \frac{1}{2}\phi\left(t_n\right)\left(d_{t_n} - y_{t_n}\right)^2, \tag{4}$$

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 $\phi(t_n)$ is the stochastic time strength function which endows each historical data with a weight depending on the time at which it occurs. We define $\phi(t_n)$ as follows

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

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 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. A Brownian motion is a real valued, continuous stochastic process $\{X(t), t \geq 0\}$, on a probability space $(\Omega, \mathscr{A}, \mathbb{P})$ with independent and stationary increments, such that $X_t - X_0$ 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 $\mathbb{E}[B(t)]^2 = t$, and the corresponding probability density function is given by $f_t(x) = (1/\sqrt{2\pi t})e^{-x^2/2t}$. The above stochastic time strength function implies that the impact of the historical data on the stock market is a time variable function. Then, the corresponding global error of data training set at kth training iterations is defined as

$$E_k = \frac{1}{N}\sum_{n=1}^{N} E\left(t_n\right). \tag{6}$$

The training objective of STNN is to update the network weights so as to minimize the global error in the data training set. The training algorithm procedure of STNN is described as follows. Step 1 is performing 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. Step 2 is determining 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. At the beginning of data processing, sets w_{ij}^0 and v_j^0 follow the uniform distribution on $(-1, 1)$, and let the neural thresholds θ_j and θ_T be 0. Step 3 is introducing the stochastic time strength function $\phi(t_n)$ in sample n error $E(t_n)$ and the global error E_k, choosing the drift function $\mu(t)$ and the volatility function $\sigma(t_n)$, and determining the activation functions $f_H(x)$ and $f_T(x)$. Step 4 is setting a predefined minimum training threshold ζ. Based on network training objective $E_k = (1/N)\sum_{n=1}^{N} E(t_n)$, if E_k is below the ζ, go to Step 6; otherwise, go to Step 5. Step 5 is updating the STNN connective weights and applying the error to compute the gradient of the weights w_{ij}, Δw_{ij}^k and the gradient of the weights v_j, Δv_j^k. For the weight nodes in the input layer, the gradient of the connective weight w_{ij} is given by

$$\Delta w_{ij} = -\eta \frac{\partial E(t_n)}{\partial w_{ij}} = \eta \varepsilon_{t_n} v_j \phi(t_n) f_H'\left(\mathrm{net}_{jt_n}\right) x_{it_n}, \quad (7)$$

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

$$\Delta v_j = -\eta \frac{\partial E(t_n)}{\partial v_j} = \eta \varepsilon_{t_n} \phi(t_n) f_H\left(\mathrm{net}_{jt_n}\right), \quad (8)$$

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

$$\begin{aligned} w_{ij}^{k+1} &= w_{ij}^k + \Delta w_{ij}^k = w_{ij}^k + \eta \varepsilon_{t_n} v_j \phi(t_n) f_H'\left(\mathrm{net}_{jt_n}\right) x_{it_n}, \\ v_j^{k+1} &= v_j^k + \Delta v_j^k = v_j^k + \eta \varepsilon_{t_n} \phi(t_n) f_H\left(\mathrm{net}_{jt_n}\right). \end{aligned} \quad (9)$$

Step 6: until the global error satisfies the predefined minimum training threshold ζ or training times reach the maximum iterations number, output the predictive value $y_{t+1} = f_T(\sum_{j=1}^{m} v_j f_H(\sum_{i=1}^{n} w_{ij} x_{it}))$.

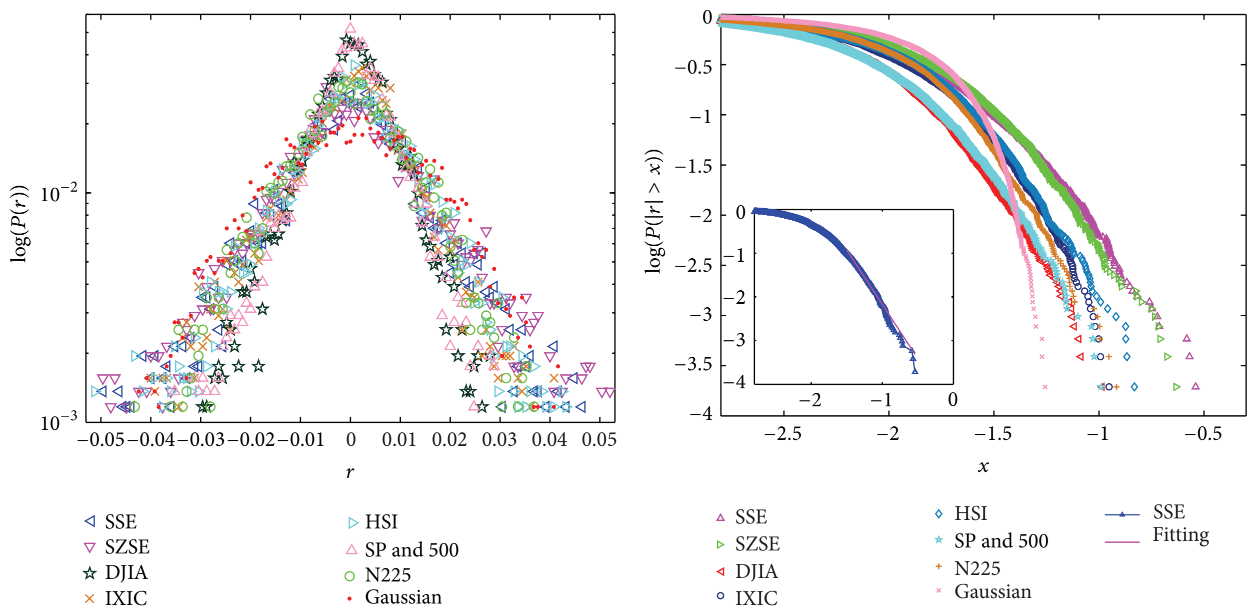
3. Data Selection

3.1. Statistical Behaviors of Price Changes and Volatilities. Study of the behaviors of stock price changes and volatilities has long been a focus in economic research. Recent empirical research shows that returns on financial markets are not Gaussian but exhibit excess kurtosis and fatter tails than the normal distribution, which is usually called the "fat-tail" phenomenon [31–33]. Higher kurtosis, which means more of the variance, is due to infrequent extreme deviations rather than frequent modestly sized deviations, indicating that the distribution is more "peaked." The power-law distribution is an efficient way to study the fat-tail phenomenon. It is a universal property that emerged from complex physics systems and is also observed in economic and financial systems. The empirical evidence has shown that the distribution of logarithmic returns follows the power-law fluctuations $P(|r(t)| > x) \sim x^{-\alpha}$ $(\alpha \approx 3)$ [34–36]; the formula of the stock logarithmic return [37, 38] from $t-1$ to t is as follows:

$$r(t) = \ln \mathcal{S}(t) - \ln \mathcal{S}(t-1), \quad t = 1, 2, \ldots, T, \quad (10)$$

where $\mathcal{S}(t)$ denotes the stock daily closing price at time t. In Figure 2, we investigate the behaviors of probability density distributions and power-law distributions of daily returns for SSE, SZSE, HSI, DJIA, IXIC, S&P 500, and N225 in the 21-year period from April 1991 to February 2012, and the corresponding Gaussian distributions are also presented for comparison. Figure 2(a) exhibits that the peak distributions of returns are obvious and that the fat-tail phenomena are also visible. Figure 2(b) shows that the power-law tails of SSE, SZSE, and HSI decay slower than those of IXIC, N225, S&P 500, and DJIA. Besides, all the indexes' power-law tails decay slower than those of Gaussian distribution. This result illustrates that the fat-tail phenomenon in Chinese financial markets is more obvious than other financial markets of the world. We also study the SSE power-law tail with liner fitting in Figure 2(b) and obtain that $\alpha \approx 2.4862$ (which is smaller than $\alpha \approx 3$); this reveals that Chinese financial markets have more severe volatilities than other stock markets of the world.

3.2. Data Selecting and Processing. Financial market volatility is central to the theory and practice of asset allocation, asset pricing, risk management, and stock return volatility forecasting in the literature [11, 39, 40]. In the present paper, we introduce a threshold value $q(\geq 0)$ to correspond to the volatility degrees and regard their corresponding index prices as STNN model's dataset. We use the data samples with a volatility degree range to train the STNN model, and the output will be almost in the same range, providing a new approach to forecast the different volatility degrees. The procedure of STNN model datasets selection is described as follows. (i) Select the stock returns whose absolute returns are greater than a threshold value q; let $\mathcal{R}(q)$ denote this corresponding return set which is given by $\mathcal{R}(q) = \{r(t) : |r(t)| \geq q, t = 1, \ldots, T\}$; see Figure 3. (ii) For a fixed threshold value q, we determine the corresponding stock trading dates in return set $\mathcal{R}(q)$; that is to say, we determine t values that satisfy $|r(t)| \geq q$, $t = 1, \ldots, T$. In order to reflect the volatility of prices, considering that the return series $r(t)$ is decided by the daily closing price values on $(t-1)$th and tth trading days (see the formula of $r(t)$ in Section 3.1), we should select the price values at time $t-1$ as the input variables. The new dates series is also arranged in a chronological order. (iii) Select the daily opening prices, the daily highest prices, the daily lowest prices, and the daily closing prices in those dates that were filtered according to Step 2 as the four input variables.



(a) (b)

Figure 2: Probability density distributions and power-law distributions of returns, respectively, for different indexes and Gaussian distribution.

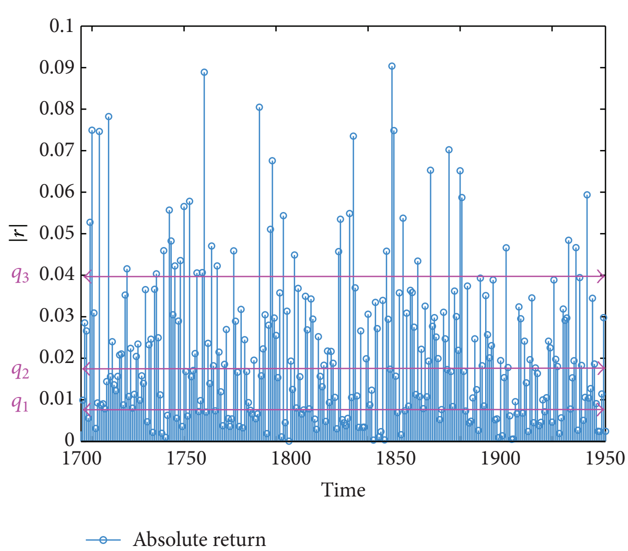


Figure 3: Schematic diagram of volatility degrees of returns for different values of q.

A trading date corresponds to a group of input variables, and the output variable is the daily closing price in the next group of variables. The dataset is divided into two parts, the training set and the testing set. We collect the data in the training set from January 2001 to December 2010 and the data in the testing set from January 2011 to February 2012. In the pretreatment and preprocessing stage, the collected data should be normalized and properly adjusted, in order to reduce the impact of noise in the stock markets; the normalized values of the above-mentioned four kinds of price data can be given as follows:

$$\widehat{\mathcal{S}}(t) = \frac{\mathcal{S}(t) - \min \mathcal{S}(t)}{\max \mathcal{S}(t) - \min \mathcal{S}(t)}. \tag{11}$$

In Table 1, the measures of training and test samples for indexes SSE, SZSE, HSI, DJIA, IXIC and S&P 500 are performed for different values of q. The value 0.0119 is the mean of daily absolute returns of SSE from January 2001 to February 2012 for 2692 trading days, and the mean values of daily absolute returns for SZSE, HSI, DJIA, IXIC, and S&P 500 are 0.0133, 0.0111, 0.0086, 0.0118, and 0.0092, respectively. Since the objective of this work is the volatility degree forecasting, we choose some different volatility degrees for the forecasting analysis. Here, we take the mean of absolute returns of SSE as $q_1 = 0.0119$, and the next five q values (from q_2 to q_6) are all 0.0003 larger than the former. Then, we can compare the errors of the prediction when q values gradually increase. In order to consider the large volatility prediction effect of STNN model, we take $q_7 = 0.0200$ and $q_8 = 0.0250$ into account. For the volatility degrees from q_1 to q_6, here we take values from q_1 to q_6 as examples in Table 1. For other different values of q, the corresponding volatility degree forecasting can be made similarly. Table 1 and Figure 3 show that the numbers of absolute returns that exceed 0.0200 or larger are few. Since q_7 and q_8 belong to large volatilities in the stock market, the events whose absolute returns exceed q_7 or q_8 do not happen frequently. From Table 1, the quantity of data samples gradually reduces with the increase of value q. When q ranges from 0.0119 to 0.0134, the corresponding numbers descend slowly. However, for $q = 0.0200$ and 0.0250,

Table 1: Measures of training and testing samples for indexes with different values of q.

Index	SSE		SZSE		HSI		DJIA		IXIC		S&P 500	
q	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
$q_0 = 0.0000$	2418	274	2418	274	2468	279	2515	286	2513	285	2524	286
$q_1 = 0.0119$	1461	135	1569	161	1356	178	1003	121	1448	162	1074	129
$q_2 = 0.0122$	1432	135	1541	158	1317	177	960	115	1413	158	1036	127
$q_3 = 0.0125$	1388	132	1510	158	1279	172	920	110	1396	155	1009	127
$q_4 = 0.0128$	1372	130	1480	157	1241	168	885	109	1364	151	994	123
$q_5 = 0.0131$	1352	126	1449	155	1203	165	844	109	1329	150	960	121
$q_6 = 0.0134$	1325	122	1419	149	1171	164	819	108	1303	145	938	117
$q_7 = 0.0200$	796	43	926	81	662	70	426	58	786	77	456	63
$q_8 = 0.0250$	520	30	649	46	416	46	254	30	528	45	285	42

the numbers of the data decrease sharply because of large volatilities of returns' relatively rare appearing in the real markets.

4. Forecasting Analysis

4.1. Parameters Determining of STNN Model. In the STNN model, the proper number of the hidden layer nodes requires validation techniques to avoid underfitting (too few neurons) and overfitting (too many neurons). According to the procedures of the three-layer network introduced in Section 2, we chose the $4 \times 8 \times 1$ neural network structure in which the number of neural nodes in the input layer is 4, the number of neural nodes in the hidden layer is 8, the number of neural nodes in the output layer is 1, the maximum training iterations number $K = 200$, $\eta = 0.01$, and the predefined minimum training threshold $\zeta = 10^{-5}$. When using STNN model to predict the daily closing price of stock index, we assume $\mu(t)$ (the drift function) and $\sigma(t)$ (the volatility function) are as follows:

$$\mu(t) = \frac{1}{(c-t)^3}, \qquad \sigma(t) = \left[\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2\right]^{1/2}, \tag{12}$$

where c is the parameter which is equal to the number of sample in the datasets, $\beta = 1.25$, and $\overline{x}$ is the mean of the sample data. Then the stochastic time strength function is given by (see Section 2)

$$\phi(t_n) = \frac{1}{1.25}\exp\left\{\int_{t_0}^{t_n}\frac{1}{(c-t)^3}dt + \int_{t_0}^{t_n}\left[\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2\right]^{1/2} dB(t)\right\}. \tag{13}$$

To evaluate the forecasting performance of the STNN model, we use the following error evaluation criteria, absolute error (AE), relative error (RE), and mean absolute error (MAE), root mean-square error (RMSE), mean absolute percentage error (MAPE). These measures are defined as follows:

$$\begin{gathered}\mathrm{AE} = d_t - y_t, \qquad \mathrm{RE} = \frac{d_t - y_t}{d_t}, \\ \mathrm{MAE} = \frac{1}{N}\sum_{t=1}^{N}\left|d_t - y_t\right|, \\ \mathrm{RMSE} = \left[\frac{1}{N}\sum_{t=1}^{N}(d_t - y_t)^2\right]^{1/2}, \\ \mathrm{MAPE} = 100 \times \frac{1}{N}\sum_{t=1}^{N}\left|\frac{d_t - y_t}{d_t}\right|,\end{gathered} \tag{14}$$

where d_t and y_t are the actual and prediction values at time t, respectively, and N is the sample size. Noting that AE, RE, MAE, RMSE, and MAPE are measures of the deviation between prediction and 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 [41].

4.2. Forecasting Results. In Section 3, the method of selecting datasets of STNN model is put forward, and the database is presented. For the different values of q, the training and testing datasets including the quantity and corresponding prices date of sample are various. Next, we predict the fluctuation behaviors of stock prices by the proposed model with different threshold values of q. (1) For $q = 0$, the selected datasets are the original daily price series. Forecasting results of SSE and S&P 500 by STNN are displayed in Figures 4 and 5, respectively.

Figures 4 and 5 exhibit that the predictive values by STNN model and the actual values are close, and the relative errors are almost below 5%. Compared with traditional backpropagation neural network (BPNN), the forecasting results are presented in Table 2, where the MAPE(100) stands for the latest 100 days of MAPE in the testing data. Table 2 shows that the evaluation criteria by STNN model are almost smaller than those by BPNN. Besides, the values of MAPE(100) are smaller than those of MAPE in all stock

Table 2: Comparisons of indexes' prediction by BPNN and STNN for $q = 0$.

Index	SSE		SZSE		HSI	
Errors	BPNN	STNN	BPNN	STNN	BPNN	STNN
MAE	28.3115	28.3190	174.6668	165.4092	317.6058	297.8046
RMSE	35.1332	35.1219	215.7711	204.0298	419.4783	394.2751
MAPE	1.0922	1.0925	1.6163	1.5212	1.5714	1.4660
MAPE (100)	0.9919	0.9922	1.2572	1.2518	0.9475	0.9699
Index	DJIA		IXIC		S&P 500	
Errors	BPNN	STNN	BPNN	STNN	BPNN	STNN
MAE	215.9167	210.2087	52.3185	50.3318	19.2175	19.0905
RMSE	248.7886	242.2453	61.8688	59.9015	22.7849	22.5809
MAPE	1.7796	1.7345	1.9288	1.8581	1.5155	1.5058
MAPE (100)	1.6742	1.6164	1.9579	1.8446	1.4190	1.3931

Table 3: Significant test of errors by STNN and BPNN models for indexes with $q = 0$.

		SSE	SZSE	HSI	DJIA	IXIC	S&P 500
	H	0	1	1	1	1	1
t	t-value	0.2909	−6.7094	−7.5285	−9.4370	−12.2114	−6.6909
	Prob. P	0.7714	$1.1243e-010$	$7.2526e-013$	$1.4218e-018$	$7.5248e-028$	$1.1727e-010$
	H	0	1	1	1	1	1
W	z-value	−0.3793	−6.2424	−7.1109	−8.5639	−11.1696	−6.8902
	Prob. P	0.7044	$4.3084e-010$	$1.1529e-012$	$1.0916e-017$	$5.7430e-029$	$5.5731e-012$

t means t-test and W means Wilcoxon signed rank test.

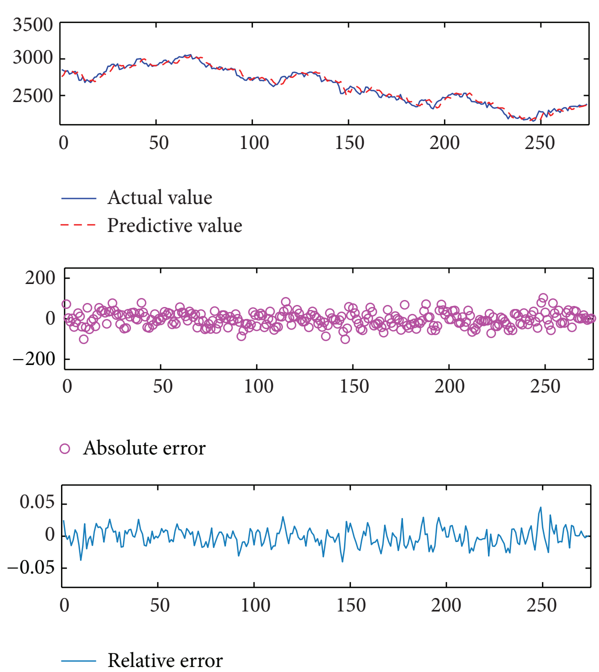


Figure 4: Comparison of forecasting result by STNN model and the daily closing prices of SSE and the corresponding errors' plots.

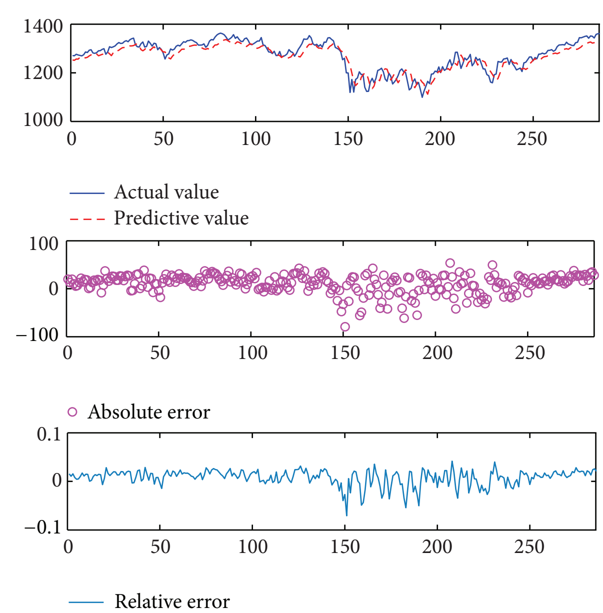


Figure 5: Comparison of forecasting result by STNN model and the daily closing prices of S&P 500 and the corresponding errors' plots.

indexes. Therefore, the short-term prediction outperforms the long-term prediction. In order to comparatively study the effect of fluctuation prediction for the STNN model and the BPNN model, we perform the paired-sample t-test and the nonparametric Wilcoxon signed rank test on two absolute AE value vectors of STNN and BPNN models (see Section 4.1). The corresponding statistical test results of six indexes are presented in Table 3. For indexes SZSE, HSI, DJIA, IXIC, and S&P 500, the values of double-tailed test p in two test

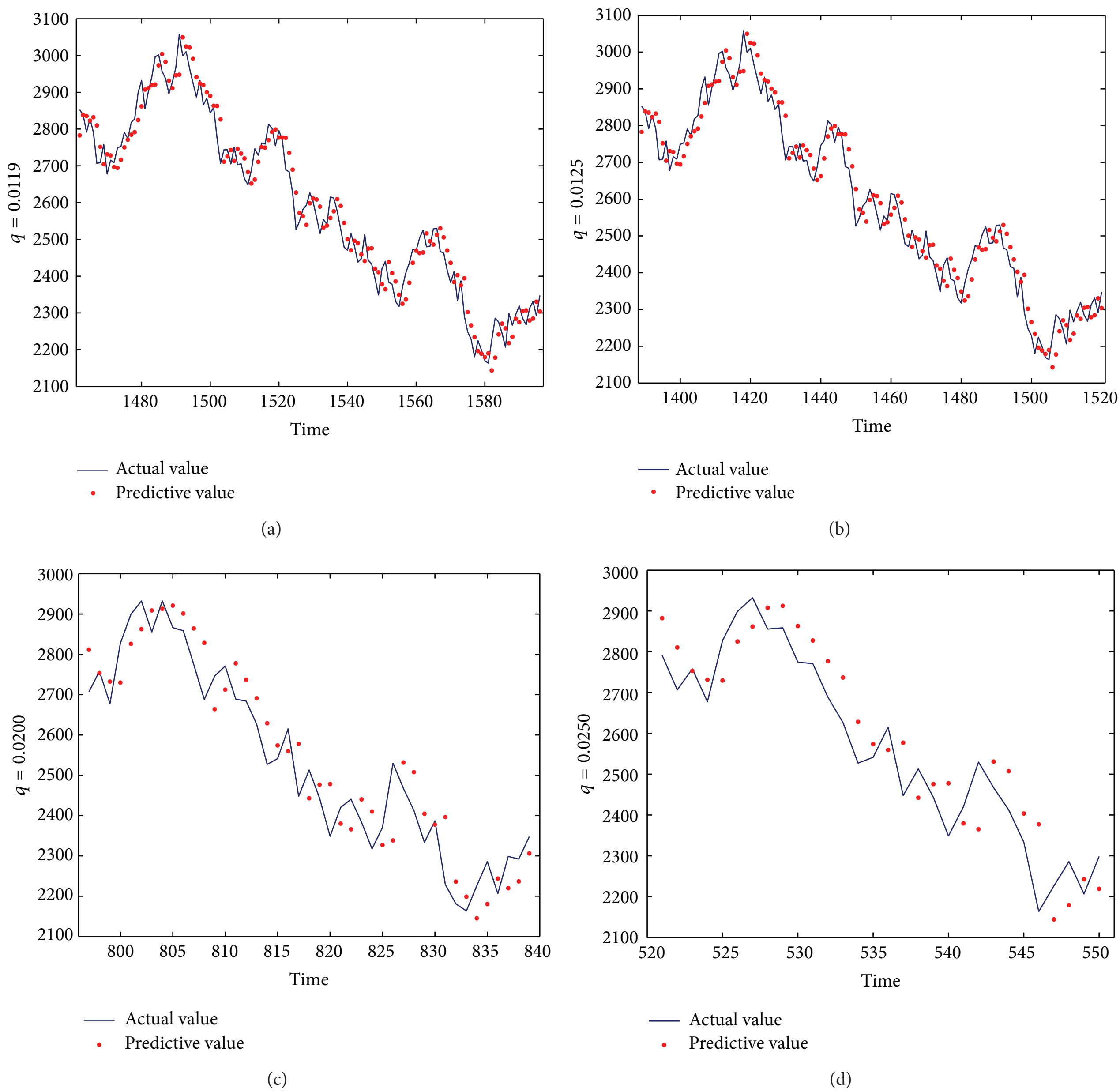


Figure 6: Volatility degree prediction of SSE by STNN with different values of q.

methods approach 0, much smaller than the significance level 0.05, and the values of H are all 1. Thus, the null hypothesis is rejected that the predicted errors of STNN and BPNN models have significant difference. Besides, in Table 2, the error evaluation criteria MAE, RMSE, and MAPE by STNN are all smaller than those by BPNN for the above indexes. This shows that the effect of price fluctuation forecasting of STNN model is superior to that of BPNN model for these five indexes. Only for SSE index, the values of H in two test methods are 0 and the values of double-tailed test p are larger than 0.05. Thus, the null hypothesis is accepted that the predicted errors by STNN and BPNN models have no significant difference for SSE index.

(2) Let q range from 0.0119 (the mean of absolute returns of SSE from January 2001 to February 2012) to 0.0250. The experiment analysis of the prediction is performed by STNN for indexes SSE, SZSE, HSI, DJIA, IXIC, and S&P 500. Figures 6 and 7 show the volatility degree forecasting of SSE and S&P 500 by STNN model with different values of q, and they illustrate the effectiveness of the corresponding forecasting. When q is small, such as $q = 0.0119$ in Figure 6(a) and $q = 0.0125$ in Figure 6(b), the better performance of volatility prediction is revealed by the empirical results; that is, the predictive values and the actual values in Figures 6(a) and 6(b) are closer than those in Figures 6(c) and 6(d). Similar results are also indicated in Figure 7.

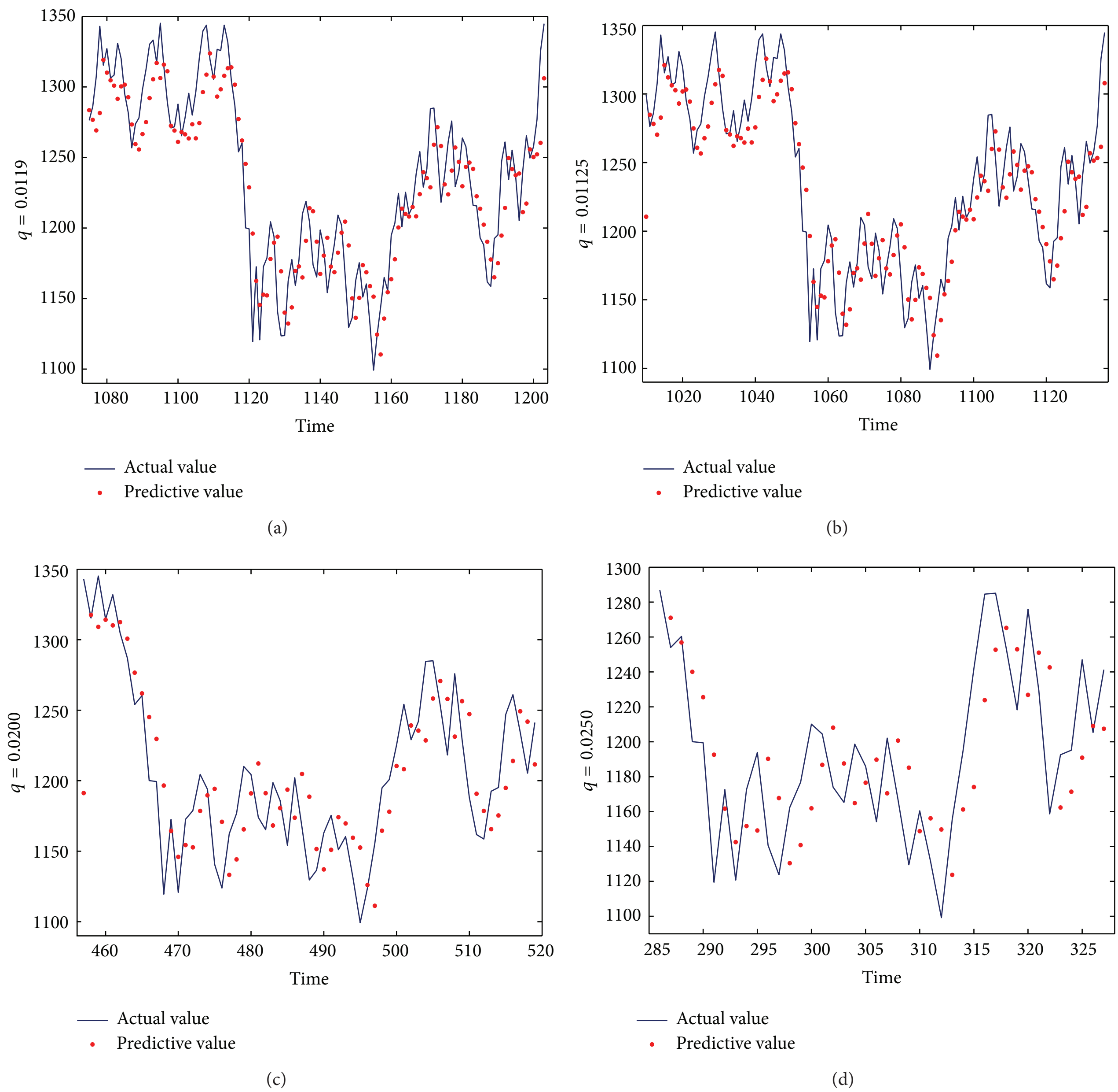


Figure 7: Volatility degree prediction of S&P 500 by STNN with different values of q.

In order to comparatively investigate the volatility degree forecasting results for the other stock indexes more clearly, we present the same forecasting for SZSE, HSI, DJIA, and IXIC. Table 4 shows the prediction performance by STNN model. The MAPE of the prediction becomes gradually larger with the increasing of q, and the experiment results exhibit that the volatility degree forecasting by STNN model is feasible.

4.3. Further Evaluation of Forecasting Performance. To further evaluate the forecasting performance, we employ three statistical analysis methods including directional symmetry (DS), correct up (CP) trend, and correct down (CD) trend [42]. CP and CD provide the correctness of predicted up trend and predicted down trend indexes in terms of percentage, respectively. The forecasting of change direction will be more precise when the values of three statistical metrics become larger. The definitions of the three statistical methods are given as follows:

$$\mathrm{DS} = \frac{100}{N_1}\sum_{t=1}^{N_1} a_t,$$
$$a_t = \begin{cases} 1 & \text{if } (y_t - y_{t-1})(d_t - d_{t-1}) \geq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{15}$$

Table 4: MAPE of indexes' prediction by STNN with different values of q.

MAPE	SSE	SZSE	HSI	DJIA	IXIC	S&P 500
$q_1 = 0.0119$	1.6330	1.9462	1.7611	1.9304	2.3165	1.8847
$q_2 = 0.0122$	1.6312	1.9424	1.7759	1.9365	2.3367	1.9184
$q_3 = 0.0125$	1.6559	1.9504	1.8220	1.9441	2.3533	1.9365
$q_4 = 0.0128$	1.6585	1.9555	1.8483	1.9478	2.3757	1.9812
$q_5 = 0.0131$	1.6897	2.0314	1.8829	1.9486	2.4019	2.0058
$q_6 = 0.0134$	1.7227	2.0025	1.9212	1.9543	2.4074	2.0302
$q_7 = 0.0200$	2.8781	2.6260	2.5884	2.3541	2.9259	2.5139
$q_8 = 0.0250$	3.2507	3.2702	2.9770	3.3216	3.6744	3.2173

Table 5: Statistical metrics for testing data with different values of q.

Index		SSE			SZSE			HSI	
q	DS	CP	CD	DS	CP	CD	DS	CP	CD
$q_1 = 0.0119$	49.25	41.94	55.56	44.38	36.23	50.55	46.33	46.67	45.98
$q_2 = 0.0122$	49.62	41.94	56.34	44.30	34.62	50.67	46.59	47.19	45.98
$q_3 = 0.0125$	50.38	42.62	57.14	44.09	35.82	50.00	47.37	47.67	47.06
$q_4 = 0.0128$	51.16	44.07	57.14	43.95	36.76	50.00	47.31	46.99	47.62
$q_5 = 0.0131$	52.00	44.64	57.97	44.81	36.92	50.56	46.34	45.68	46.99
$q_6 = 0.0134$	51.24	43.40	57.35	46.95	38.10	51.76	46.01	45.68	46.34
$q_7 = 0.0200$	46.51	36.84	54.17	46.25	36.36	53.19	47.82	52.78	42.42
Index		DJIA			IXIC			S&P 500	
q	DS	CP	CD	DS	CP	CD	DS	CP	CD
$q_1 = 0.0119$	48.33	47.76	49.06	52.80	52.87	52.70	45.31	43.66	47.37
$q_2 = 0.0122$	45.61	44.44	47.06	51.59	50.00	53.42	44.44	43.48	45.61
$q_3 = 0.0125$	44.55	42.37	47.06	51.95	51.22	52.78	44.44	43.48	45.61
$q_4 = 0.0128$	44.95	43.10	47.32	52.00	51.85	52.17	46.72	45.45	48.21
$q_5 = 0.0131$	45.04	42.37	48.08	51.01	51.25	50.72	47.50	46.15	49.09
$q_6 = 0.0134$	44.95	42.11	47.06	51.39	51.28	51.52	48.28	47.54	49.09
$q_7 = 0.0200$	45.61	41.38	50.00	42.11	36.11	47.50	45.16	38.71	51.61

where N_1 is the number of training (testing) samples;

$$\mathrm{DP} = \frac{100}{N_2}\sum_{t=1}^{N_2} a_t,$$
$$a_t = \begin{cases} 1 & \text{if } (y_t - y_{t-1}) > 0, \\ & (y_t - y_{t-1})(d_t - d_{t-1}) \geq 0, \\ 0 & \text{otherwise}, \end{cases} \quad (16)$$

where N_2 is the number of training (testing) samples for $(y_t - y_{t-1}) > 0$;

$$\mathrm{CD} = \frac{100}{N_3}\sum_{t=1}^{N_3} a_t,$$
$$a_t = \begin{cases} 1 & \text{if } (y_t - y_{t-1}) < 0, \\ & (y_t - y_{t-1})(d_t - d_{t-1}) \geq 0, \\ 0 & \text{otherwise}, \end{cases} \quad (17)$$

where N_3 is the number of training (testing) samples for $(y_t - y_{t-1}) < 0$.

Table 5 gives the numerical forecasting results for SSE, SZSE, HSI, DJIA, IXIC, and S&P 500. All the stock indexes change a little when q varies from q_1 to q_6. The direction forecasting effect of IXIC index is the best from q_1 to q_6 since the values of DS, CP, and CD all exceed 50. When $q = q_7$, some criteria change sharply; for example, for SSE index, the values of DS and CP decrease from 51.24 and 43.40 (for $q = q_6$) to 46.51 and 36.84 (for $q = q_7$), respectively. The value of CD of S&P 500 increases, but its value of CP reduces when q varies from q_6 to q_7, and HSI index has the adverse result. From Table 5, we can conclude that the direction forecasting by STNN model has little difference when q is small but may cause obvious changes when q is large.

5. Conclusion

In the present paper, we develop a neural network with the stochastic time strength function to forecast the volatility degrees of stock market indexes. The effectiveness of STNN model has been analyzed by performing the numerical experiments on the price data of SSE, SZSE, HSI, DJIA, IXIC, and S&P 500. We also investigate the statistical behaviors of the tail distributions (or volatilities) for SSE, SZSE, and other global indexes by comparison, revealing that the Chinese financial markets have more severe fluctuations than other stock markets of the world. We select different volatility degrees for different threshold values of q, and the corresponding training and testing databases are introduced. The empirical research of this work indicates that some prediction results have been improved by STNN model, and the forecasting effect will decline as q becomes larger. We hope this new approach can make some contributions to financial market volatility forecasting.

Acknowledgment

The authors were supported in part by the National Natural Science Foundation of China Grant no. 71271026 and Grant no. 10971010.

References

[1] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*, Prentice Hall, Englewood Cliffs, NJ, USA, 3rd edition, 1994.

[2] Y. Guo and J. Wang, "Simulation and statistical analysis of market return fluctuation by Zipf method," *Mathematical Problems in Engineering*, vol. 2011, Article ID 253523, 13 pages, 2011.

[3] A. Lendasse, E. D. Bodt, V. Wertz, and M. Verleysen, "Nonlinear financial time series forecasting: application to the Bel 20 stock market index," *European Journal of Economic and Social Systems*, vol. 14, pp. 81–91, 2000.

[4] R. N. Mantegna and H. E. Stanley, *An Introduction to Econophysics*, Cambridge University Press, Cambridge, UK, 1999.

[5] T. C. Mills, *The Econometric Modelling of Financial Time Series*, Cambridge University Press, Cambridge, UK, 2nd edition, 1999.

[6] H. Niu and J. Wang, "Volatility clustering and long memory of financial time series and financial price model," *Digital Signal Processing*, vol. 23, no. 2, pp. 489–498, 2013.

[7] F. Wang and J. Wang, "Statistical analysis and forecasting of return interval for SSE and model by lattice percolation system and neural network," *Computers and Industrial Engineering*, vol. 62, no. 1, pp. 198–205, 2012.

[8] J. Wang, Q. Wang, and J. Shao, "Fluctuations of stock price model by statistical physics systems," *Mathematical and Computer Modelling*, vol. 51, no. 5-6, pp. 431–440, 2010.

[9] T. G. Andersen, T. Bollerslev, and F. X. Diebold, "Roughing it up: including jump components in the measurement, modeling, and forecasting of return volatility," *Review of Economics and Statistics*, vol. 89, no. 4, pp. 701–720, 2007.

[10] A. J. Patton, "Volatility forecast comparison using imperfect volatility proxies," *Journal of Econometrics*, vol. 160, no. 1, pp. 246–256, 2011.

[11] J. Yu, "Forecasting volatility in the New Zealand stock market," *Applied Financial Economics*, vol. 12, no. 3, pp. 193–202, 2002.

[12] E. M. Azoff, *Neural Network Time Series Forecasting of Financial Market*, John Wiley & Sons, New York, NY, USA, 1994.

[13] A.-S. Chen, M. T. Leung, and H. Daouk, "Application of neural networks to an emerging financial market: forecasting and trading the Taiwan Stock Index," *Computers and Operations Research*, vol. 30, no. 6, pp. 901–923, 2003.

[14] S. A. Hamid and Z. Iqbal, "Using neural networks for forecasting volatility of S&P 500 Index futures prices," *Journal of Business Research*, vol. 57, no. 10, pp. 1116–1125, 2004.

[15] I. Kaastra and M. Boyd, "Designing a neural network for forecasting financial and economic time series," *Neurocomputing*, vol. 10, no. 3, pp. 215–236, 1996.

[16] V. Kodogiannis and A. Lolis, "Forecasting financial time series using neural network and fuzzy system-based techniques," *Neural Computing and Applications*, vol. 11, no. 2, pp. 90–102, 2002.

[17] R. R. Trippi and E. Turban, *Neural Networks in Finance and Investing: Using Artificial Intelligence To Improve Real-World Performance*, Probus, Chicago, ILL, USA, 1993.

[18] F. Virili and B. Freisleben, "Neural network model selection for financial time series prediction," *Computational Statistics*, vol. 16, no. 3, pp. 451–463, 2001.

[19] J. Yao, Y. Li, and C. L. Tan, "Option price forecasting using neural networks," *Omega*, vol. 28, no. 4, pp. 455–466, 2000.

[20] G. Armano, M. Marchesi, and A. Murru, "A hybrid genetic-neural architecture for stock indexes forecasting," *Information Sciences*, vol. 170, no. 1, pp. 3–33, 2005.

[21] Z. Liao and J. Wang, "Forecasting model of global stock index by stochastic time effective neural network," *Expert Systems with Applications*, vol. 37, no. 1, pp. 834–841, 2010.

[22] F. Liu and J. Wang, "Fluctuation prediction of stock market index by Legendre neural network with random time strength function," *Neurocomputing*, vol. 83, pp. 12–21, 2012.

[23] H. Liu and J. Wang, "Integrating independent component analysis and principal component analysis with neural network to predict chinese stock market," *Mathematical Problems in Engineering*, vol. 2011, Article ID 382659, 15 pages, 2011.

[24] J. Wang, H. Pan, and F. Liu, "Forecasting crude oil price and stock price by jump stochastic time effective neural network model," *Journal of Applied Mathematics*, vol. 2012, Article ID 646475, 15 pages, 2012.

[25] P. P. Balestrassi, E. Popova, A. P. Paiva, and J. W. Marangon Lima, "Design of experiments on neural network's training for nonlinear time series forecasting," *Neurocomputing*, vol. 72, no. 4–6, pp. 1160–1178, 2009.

[26] S. Haykin, *Neural Networks: A Comprehensive foundation*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1999.

[27] H. White, "Learning in artificial neural networks: a statistical perspective," *Neural Computation*, vol. 1, pp. 425–464, 1989.

[28] G. Zhang, B. Eddy Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: the state of the art," *International Journal of Forecasting*, vol. 14, no. 1, pp. 35–62, 1998.

[29] H. Demuth and M. Beale, *Network Toolbox: For Use with MATLAB*, The Math Works, Natick, Mass, USA, 5th edition, 1998.

[30] D. E. Rumelhart and J. L. McClelland, *Distributed Processing: Explorations in the Micro-Structure of Cognition*, The MIT Press, Cambridge, UK, 1986.

[31] F. Black and M. Scholes, "The pricing of options and corporate liabilities," *Journal of Political Economy*, vol. 81, pp. 637–654, 1973.

[32] R. Gaylord and P. Wellin, *Computer Simulations with Mathematica: Explorations in the Physical, Biological and Social Science*, Springer, New York, NY, USA, 1995.

[33] K. Ilinski, *Physics of Finance: Gauge Modeling in Non-Equilibrium Pricing*, John Wiley & Sons, New York, NY, USA, 2001.

[34] D. Xiao and J. Wang, "Modeling stock price dynamics by continuum percolation system and relevant complex systems analysis," *Physica A*, vol. 391, pp. 4827–4838, 2012.

[35] Y. Yu and J. Wang, "Lattice-oriented percolation system applied to volatility behavior of stock market," *Journal of Applied Statistics*, vol. 39, no. 4, pp. 785–797, 2012.

[36] J. Zhang and J. Wang, "Modeling and simulation of the market fluctuations by the finite range contact systems," *Simulation Modelling Practice and Theory*, vol. 18, no. 6, pp. 910–925, 2010.

[37] D. Lamberton and B. Lapeyre, *Introduction to Stochastic Calculus Applied to Finance*, Chapman & Hall, London, UK, 1996.

[38] S. M. Ross, *An Introduction to Mathematical Finance*, Cambridge University Press, Cambridge, UK, 1999.

[39] R. G. Donaldson and M. Kamstra, "An artificial neural network-GARCH model for international stock return volatility," *Journal of Empirical Finance*, vol. 4, no. 1, pp. 17–46, 1997.

[40] T. Wang, J. Wang, J. Zhang, and W. Fang, "Voter interacting systems applied to Chinese stock markets," *Mathematics and Computers in Simulation*, vol. 81, no. 11, pp. 2492–2506, 2011.

[41] S. Makridakis, "Accuracy measures: theoretical and practical concerns," *International Journal of Forecasting*, vol. 9, no. 4, pp. 527–529, 1993.

[42] L. Cao and F. E. H. Tay, "Financial forecasting using Support Vector Machines," *Neural Computing and Applications*, vol. 10, no. 2, pp. 184–192, 2001.