

Complex System Analysis on Voter Stochastic System and Jump Time Effective Neural Network of Stock Market

Jun Wang, Huopo Pan

*School of Science, Beijing Jiaotong University
Beijing 100044, P.R. CHINA*

Yiduan Wang

*School of Economics and Management, Beijing Jiaotong University
Beijing 100044, P.R. CHINA*

Hongli Niu

*School of Science, Beijing Jiaotong University
Beijing 100044, P.R. CHINA
E-mail: niuhonglibjtu@yeah.net*

Received 8 January 2014

Accepted 25 May 2015

Abstract

The finite-range voter system, one of stochastic particle systems, is applied to model a financial price process for further description and investigation of fluctuations of Shanghai Composite Index. For different parameter values of the intensity λ and the range R , we investigate the statistical behaviors of the simulation data for this financial model. Then we develop the random jump time effective neural network model to forecast the fluctuations of Shanghai Composite Index. Moreover, we compare the two models by analyzing the returns and the absolute returns of Shanghai Composite Index, the simulation data and the predictive data through Detrended Fluctuation Analysis and classic Rescaled Range Analysis.

Keywords: Prediction; financial time series; voter interacting system; random jump time effective neural network; detrended fluctuation analysis; rescaled range analysis

1. Introduction

As the stock markets are becoming deregulated worldwide, the modelling of the dynamics of the forwards prices is becoming a key problem in the risk management, physical assets valuation, and derivatives pricing, and it is also important to understand the statistical properties of fluctuations of stock prices in globalized securities markets. By applying the theory of stochastic systems, some research work has been done to investigate the statistical behaviors of fluctuations for stock markets, and the corresponding valuation and hedging of contingent claims for the price process

models are also studied¹⁻⁷. In these financial models, the main assumption is that the stock price fluctuation is influenced by the information in a stock market, and the investors decide their investment opinions by other investors' attitudes, so the investors' investment attitudes to the stock market lead to the stock price fluctuation. In the present paper, we apply a stochastic interacting particle system---the finite-range voter system⁸⁻¹³ to study the fluctuation behaviors of the return processes. Since interacting particle systems consist of a large number of interacting units, we think that economic systems such as financial markets are similar to interacting particle systems in that they are comprised of

a large number of interacting "agents". Through the computer simulation¹⁴⁻¹⁷, we study the statistical behaviors of returns for this financial model.

Artificial neural networks (ANN) is one of technologies that have made great progress in studying stock markets¹⁸⁻²⁵. ANN has good self-learning ability, strong anti-jamming capability, has been widely used in the financial fields such as stock prices, profits, exchange rate, risk analysis and prediction. In a financial market, although the historical data has a great influence on the investors' positions, we think that the impacts of different historical data on the stock price are not the same. In the present paper, we suppose that the degree of impact of the data depends on its occurrence date (or time), we give the data a high level effect when it is very near to the current state. Furthermore, we also introduce the Brownian motion and Poisson jump in the financial model²⁶⁻²⁷, in order to make the model have the effect of random movement and random jump while maintaining the original trend. In the real stock market, drastic fluctuations (or jumps) in financial assets play a crucial role in volatility forecasting. And jumps have positive and significant impact on future volatilities. In this work, the artificial neural network model based on random jump time effective function is applied to forecast Shanghai Composite Index (SHCI).

It is widely acknowledged that financial market is a complex dynamic system, which are characterized by various interesting statistical fluctuation properties. The simulation and forecasting of the financial stock indices have received considerable attentions from the financial investors. The main purpose in this paper is to perform the complex statistical analysis of the simulated and forecasted data of real market index, with the hoping to find the effectiveness of the proposed simulating and forecasting models in comparison with the actual data whether from a macro point of view or from a micro point of view. China has two stock markets, Shanghai Stock Exchange and Shenzhen Stock Exchange, the most important indices which are issued by these two stock exchanges are Shanghai Composite Index (SHCI) and Shenzhen Compositional Index (SZCI). In the present paper, the empirical research for the financial market fluctuations is made for Shanghai Composite Index (SHCI). We select the daily closing prices of SHCI from the years 2003 to 2007 for training and those of SHCI from 2008 to 2009 for testing by the random

jump time effective neural network, and we also consider the simulation data of the same time period as SHCI by the financial model which is derived from the finite-range voter interacting system. Therefore, the difficulty in this paper is that we have to properly select and adjust modelling parameters in the financial price simulation process as well as to introduce reasonable and useful random jump time function into the forecasting algorithm of the artificial neural network. Further, in the Section 4, we comparatively analyze the returns and the absolute returns of these two systems through Detrended Fluctuation Analysis (DFA) and classic Rescaled Range (R/S), which are both applied to study the extent of long-range correlations in corresponding financial time series.

2. Experimental Modelling

In this section, we introduce two important modelling processes for stock index. One is the financial stock price model by voter interacting system, according to which we can obtain various simulative stock price time series. Another is the random jump time effective neural network model in which we introduce a random jump time effective function to the BP algorithm. From this model, we can achieve the forecasting of stock index.

2.1. Modelling a stock price model by voter interacting system

First we give the brief definitions and properties of voter interacting system. Voter interacting system is one of stochastic particle systems, we think of the sites of the d -dimensional integer lattice as being occupied by persons who either in favor of or opposed to some issue. Let η_s^d denote the state at time s with the initial state $\eta_0^d = A$ for the voter system^{9,12,13}. For more generally, we can consider the initial distribution as ν_θ , the product measure with density θ , that is, each site is independently occupied with probability θ and let η_s^θ denote the voter system with initial distribution ν_θ . Let $\eta_s^{\{0\}} = \{0\}$, and $\eta_s^{\{0\}}(u)$ be the state of $u \in \square^d$ at time s with the initial point $\{0\}$. More formally, the stochastic dynamics of the voter system η_s is a Markov process on $\Omega = \{0,1\}^{\square^d}$ whose generator has the form

$$Af(\eta) = \sum_u c(u,\eta)[f(\eta^u) - f(\eta)] \quad (1)$$

where the functions f (on Ω) depend on finitely many coordinates, and $\eta^u(v) = \xi(v)$ if $u \neq v$, $\eta^u(v) = 1 - \eta(v)$

if $u = v$, for $u, v \in \mathbf{Z}^d$. $c(u, \eta)$ is the transition rate function for the process which is given by follows¹²⁻¹³, for any $\eta \in \Omega = \{0, 1\}^{\mathbf{Z}^d}$

$$c(u, \eta) \begin{cases} \lambda \sum_{v: |u-v| \leq R} p(u, v) \eta(v) & \text{if } \eta(u) = 0 \\ \sum_{v: |u-v| \leq R} p(u, v) [1 - \eta(v)] & \text{if } \eta(u) = 1 \end{cases} \quad (2)$$

where R is the range of the model, sites in $\{v: |u-v| \leq R\}$ are the neighbors of u , and λ is the intensity parameter of the model. When $\lambda = 1$, the model is called the voter system, and when $\lambda > 1$, the model is called the biased voter system. In the above definition, let $p(u, v) \geq 0$ for $u, v \in \mathbf{Z}^d$, and $\sum_v p(u, v) = 1$ for all $u \in \square^d$, and we assume that $p(u, v)$ is such that the Markov chain with those transition probabilities is irreducible.

For the biased voter model ($\lambda > 1$), there is a “critical value” for the process, that is, on $\Omega = \{0, 1\}^{\square^d}$ and with the corresponding probability P , the critical value λ_c is defined as

$$\lambda_c = \inf \left\{ \lambda : P \left(\left| \eta_s^{(0)} \right| > 0, \text{ for all } s \geq 0 \right) > 0 \right\} \quad (3)$$

where $\left| \eta_s^{(0)} \right|$ is the cardinality of $\eta_s^{(0)}$. On d -dimensional lattice, if $\lambda < \lambda_c$, the process dies out (becomes vacant) exponentially fast, if $\lambda > \lambda_c$, the process survives with the positive probability.

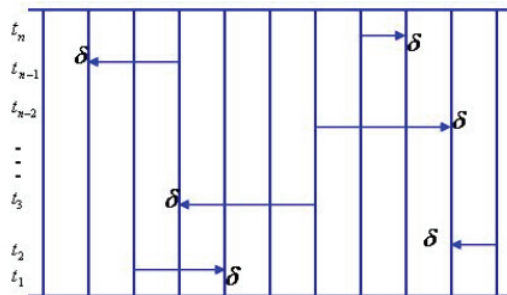


Fig. 1. The chart of finite-range voter system with $d = 1, \lambda = 1$ and $R = 3$.

Now we introduce the graphical representation of the model, since the graphical representation is necessary for our computer simulations. For simplicity, we give the construction of graphical representation for 1-dimensional voter system with $\lambda = 1$ and $R = 3$ for more general cases, see Refs. 9, 12 and 13. Thinking of 1-dimensional integer points as being laid out on a horizontal axis, with the time lines being placed

vertically, above that axis. Define independent Poisson processes with rate for each time lines, at each event time (u, s) , we choose one of its six neighbors with probability $1/6$, and draw an arrow from that neighbor point to (u, s) , and write a δ at (u, s) , see Fig. 1. To construct the process from this “graphical representation”, we imagine fluid entering the bottom at the points in η_0 and flowing up the structure. The δ 's are the dams and the arrows are pipes which allow the fluid to flow in the indicated direction.

In the following, based on the voter system on d -dimensional integer lattice, we model a return process for a stock market. We assume that market information leads to the fluctuation of a stock price, and there are three kinds of information: buying information, selling information and neutral information. The fluctuation of a stock price relies on the investor’s investment attitudes, which accordingly classify buying stock, selling stock and holding stock. Consider a model of auctions for the same stock defined above, we can derive the stock price process from the auctions. Suppose that each trader can trade the stock several times at each day $t \in \{1, 2, \dots, n\}$, but at most one unit number of the stock at each time. Let l be the time length of trading time in each trading day, we denote the stock price at time s in the t -th trading day by $S_t(s)$, where $s \in [0, l]$. Suppose that this stock consists of $2m_2 + 1$ (m_2 is large enough) investors, who are located in a line $\{-m_2, \dots, -1, 0, 1, \dots, m_2\} \subset \mathbf{Z}$ (s-imilarly for d -dimensional lattice \square^d). At the beginning of trading in each day, suppose that only the investor at the site $\{0\}$ receives some news. We define a random variable $\xi_t(\{0\})$ for this investor, suppose that this investor taking buying position ($\xi_t(\{0\}) = 1$), selling position ($\xi_t(\{0\}) = -1$) or neutral position ($\xi_t(\{0\}) = 0$) with probability p_1, p_{-1} or $1 - (p_1 + p_{-1})$ respectively. Then this investor sends bullish, bearish or neutral signal to his nearest neighbors. According to d -dimensional voter process system, investors can affect each other or the news can be spread, which is assumed as the main factor of price fluctuations for the investors. Moreover, here the investors can change their buying positions or selling positions to neutral positions independently at a constant rate. More specifically: (a) When $\xi_t(\{0\}) = 1$ and if $\eta_s^{(0)}(u) = 1$, we say that the investor at u takes buying position at time s , and this investor recovers to neutral position 0 at a rate equal to the number of the vacant neighbors; if $\eta_s^{(0)}(u) = 0$, we

think the investor at u takes neutral position at time s , and this investor is changed to take buying position by his neighbors at rate $\lambda \sum_{v:|u-v|\leq R} \eta_s^{(0)}(v)$. In this case, the more investors with taking buying positions, the more possible the stock price goes up. (b) When $\xi_t(\{0\}) = -1$ and if $\eta_s^{(0)}(u) = 1$, we say that the investor at u takes selling position at time s , also this investor recovers to neutral position 0 at a rate equal to the number of the vacant neighbors; if $\eta_s^{(0)}(u) = 0$, the investor is changed to take selling position by his neighbors at rate $\lambda \sum_{v:|u-v|\leq R} \eta_s^{(0)}(v)$. In this case, the more investors with taking selling positions, the more possible the stock price goes down. (c) When the initial random variable $\xi_t(\{0\}) = 0$, the process $\eta_s^{(0)}(u)$ is ignored, this means that the investors don't affect the fluctuation of the stock price.

For l large enough, the normalized trend function is given by

$$C_t(s) = \xi_t(\{0\}) \frac{|\eta_s^{(0)}|}{2m_2}, \quad s \in [0, l] \quad (4)$$

where $|\eta_s^{(0)}| = \sum_{u=-m_2}^{m_2} \eta_s^{(0)}(u)$. From the above definitions and Refs. 28-31, we define the stock price at t -th trading day $t (t \in \{1, 2, \dots, n\})$ as

$$S_t(s) = e^{\alpha C_t(s)} S_{t-1}(s), \quad s \in [0, l] \quad (5)$$

where $\alpha > 0$ represents the depth parameter of the market. Then we have

$$S_t(s) = S_0 \exp\left\{\alpha \sum_{k=1}^t C_k(s)\right\}, \quad s \in [0, l] \quad (6)$$

where S_0 is the stock price at time 0.

2.2. Random jump time effective neural network model

First we introduce the three-layer BP neural network model in Fig. 2, for the details see Refs. 17 and 19. For any fixed neuron $n (n = 1, 2, \dots, N)$, the model has the following structure: let $\{x_i(n) : i = 1, 2, \dots, p\}$ denote the set of input of neurons, $\{y_j(n) : j = 1, 2, \dots, m\}$ denote the set of output of hidden layer neurons; V_i is weight that connects the node i in the input layer neurons to the node j in the hidden layer, W_j is weight that connects the node j in the hidden layer neurons to the node k in the output layer; and $\{o_k(n) : k = 1, 2, \dots, q\}$ denote the set of output of neurons. Then the output value for a unit is given by the following function

$$y_j(n) = f\left(\sum_{i=1}^p V_i x_i(n) - \theta_j\right), o_k(n) = f\left(\sum_{j=1}^m W_j y_j(n) - \theta_k\right) \quad (7)$$

where θ_j, θ_k are the neural thresholds, $f(x) = 1/(1 + e^{-x})$ is Sigmoid activation function. Let $T_k(n)$ be the actual value of data sets, then the error of the corresponding neuron k to the output is defined as $\varepsilon_k = T_k - o_k$.

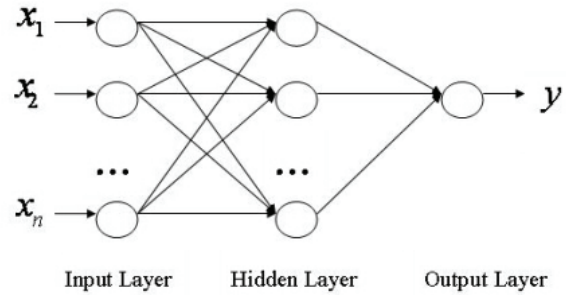


Fig. 2. The plot of three-layer neural network.

In stock markets, the empirical research shows that the fluctuations of price changes are believed to follow a Gaussian distribution for long time intervals but to deviate from it for short steps, and the actual price changes usually exhibit excess kurtosis and fatter tails than the normal distribution, which is called the “fat-tail” phenomenon. This phenomenon is usually caused by the drastic fluctuations of stock prices. In view of the above reality status, the error of the output is defined as follows. Let $\varepsilon = \varepsilon_k^2 / 2$, then the error of the sample $n (n = 1, 2, \dots, N)$ is defined as

$$e(n, t) = \frac{1}{2} \phi(t) \sum_{k=1}^q (T_k(n) - o_k(n))^2 \quad (8)$$

where $\phi(t)$ is the random jump time effective function. Now we defined $\phi(t)$ as following

$$\phi(t_1 - t_n) = \frac{1}{\tau} \exp\left\{-\int_{t_n}^{t_1} \mu(t) dt - \int_{t_n}^{t_1} \sigma(t) dB(t)\right\} \prod_{l=1}^{N(t_1-t_n)} e^{J_l} \quad (9)$$

where $\tau (\tau > 0)$ is the time strength coefficient, t_1 is the current time or the time of newest data in data set, t_n is an arbitrary time point in data set. $J_l (l = 1, 2, \dots, N(t))$ are independent and identically distributed jump processes and J_l obey the normal distribution with mean μ_j and variance σ_j . $N(t) (t > 0)$ is a stochastic Poisson process with the intensity γ . $\mu(t)$ is the drift function (or the trend term), and $\sigma(t)$ is the volatility function, $B(t)$ is the standard Brownian motion, see Refs. 29 and 31. The random jump time effective function implies that: (a)

The recent information has a stronger effect on the investors than the old information. In details, the nearer the events happen, the greater the investors and market are affected. (b) The information (or data) at the jump time may have a stronger (or weaker) effect on the investors. Then the total error of all data training set in the set output layer with the random jump time effective function are defined as

$$E = \frac{1}{N} \sum_{n=1}^N E(n, t) = \frac{1}{N} \sum_{n=1}^N \frac{1}{\tau} e^{-\int_{t_n}^t \mu(t) dt - \int_{t_n}^t \sigma(t) dB(t)} \times \prod_{l=1}^{N(t_i-t_n)} e^{J_l} \sum_{k=1}^q \frac{1}{2} (T_k(n) - o_k(n))^2 \quad (10)$$

The training algorithms procedures of the neural network is described as follows:

Step 1: Normalizing the data as follows:

$$S(t)' = [S(t) - \min S(t)] / [\max S(t) - \min S(t)].$$

Step 2: At the beginning of data processing, connective weights V_i and W_j follow the uniform distribution on $(-1, 1)$, and let the neural threshold θ_k, θ_j be 0.

Step 3: Introducing the jump stochastic time effective function $\phi(t)$ in the error function $e(n, t)$. Choosing different volatility parameters. Giving the transfer function from input layer to hidden layer and the transfer function from hidden layer to output layer.

Step 4: Establishing an error acceptable model and setting pre-set minimum error. If output error is below pre-set minimum error, go to Step 6, otherwise go to Step 5.

Step 5: Modifying connective weights: calculate backward for the node in output layer:

$$\delta_o(n) = \frac{1}{\tau} e^{-\int_{t_n}^t \mu(t) dt - \int_{t_n}^t \sigma(t) dB(t)} \prod_{l=1}^{N(t_i-t_n)} e^{J_l} o(n) [o(n) - T(n)] [1 - o(n)]. \quad (11)$$

and calculate δ backward for the node in hidden layer:

$$\delta_h(n) = \frac{1}{\tau} e^{-\int_{t_n}^t \mu(t) dt - \int_{t_n}^t \sigma(t) dB(t)} \prod_{l=1}^{N(t_i-t_n)} e^{J_l} o(n) [1 - o(n)] \sum_{h'} W_j \delta_{h'}(n) \quad (12)$$

where $o(n)$ is the output of the neuron n , $T(n)$ is the actual value of the neuron n in data sets, $o(n)[1 - o(n)]$ is the derivative of the sigmoid activation function and h' is each of the node which connect with the node h and in the next hidden layer after node h . Modifying the weights from the layer to the previous layer:

$$W_j(n+1) = W_j(n) + \eta \delta_o(n) y(n) \text{ or} \quad (13)$$

$$V_j(n+1) = V_j(n) + \eta \delta_k(n) x(n)$$

where η is learning step, which usually takes constant between 0 and 1.

Step 6: Outputting the predictive value.

3. Statistical methodology

3.1. Detrended fluctuation analysis

Detrended fluctuation analysis (DFA) is a scaling analysis method providing the scaling exponent α to represent the correlation properties³². Briefly, for a given stochastic time series $S(i), i=1, 2, \dots, N$ with the sampling period Δt , the DFA method can be implemented as follows:

Step 1: Compute the mean $\bar{S} = 1/N \sum_{i=1}^N S(i)$, and obtain an integrated time series $y(j) = 1/N \sum_{i=1}^j (S(i) - \bar{S})$. Then divide the integrated time series into boxes of equal size, n .

Step 2: In each box, fit the integrated time series by using a polynomial function, $y_{fit}(i)$. For order- l DFA, l order polynomial function should be applied to the fitting, and we choose $l=2$ in this paper. Then calculate the detrended fluctuation function as follows:

$$Y(i) = y(i) - y_{fit}(i). \quad (14)$$

Step 3: For a given box size n , calculate the root mean square fluctuation:

$$F(n) = \left(\frac{1}{N} \sum_{i=1}^N [Y(i)]^2 \right)^{\frac{1}{2}}. \quad (15)$$

A power-law relation between $F(n)$ and the box size n indicates the presence of scaling: $F(n) \propto n^\alpha$ (where $F(n)$ is calculated by $\log F(n) = \beta + \alpha \log n$). The parameter α , called the scaling exponent or correlation exponent, represents the correlation properties of the time series: if $\alpha = 0.5$, there is no correlation and the time series is uncorrelated; if $0 < \alpha < 0.5$, it indicates the power-law anti-correlations; if $0.5 < \alpha < 1$, the time series has the persistent long-range power-law correlations; if $1 < \alpha$, it indicates that the correlations exist but not in a power-law form.

3.2. Rescaled Range Analysis

The evaluation of the long memory of one time series can be made through various methodologies. Hurst exponent H is an important measure of the long memory, which can be calculated through the classic Rescaled Range (R/S) Analysis³³⁻³⁵. The research shows

that, when $H = 0.5$, the time series belongs to Brownian motion and the variables are independent and the correlation coefficients are 0; when $0 < H < 0.5$, the variables are negative correlation and are anti-persistence; when $0.5 < H < 1$, that means the time sequence is persistent and has the quantification of long-term memory. In order to calculate the Hurst exponent H , the time series x_t with N numbers is divided by A parts. And each part marks as $x_{a,k}$, $a = 1, 2, \dots, A$. For $t = 1, 2, \dots, n$, let

$$\begin{aligned} \bar{x}_a &= \frac{1}{n} \sum_{k=1}^n x_{a,k}, \quad X_{a,t} = \sum_{k=1}^t (x_{a,k} - \bar{x}_a) \\ R_a &= \max_{1 \leq t \leq n} X_{a,t} - \min_{1 \leq t \leq n} X_{a,t}, \quad S_a = \left[\frac{1}{n} \sum_{k=1}^n (x_{a,k} - \bar{x}_a)^2 \right]^{1/2} \end{aligned} \quad (16)$$

By computing R_a / S_a by each part data, then we have the mean as follows

$$(R/S)_n = \frac{1}{A} \sum_{a=1}^A (R_a / S_a). \quad (17)$$

The Hurst exponent H is obtained by the expression $\ln(R/S)_n = \ln C + H \ln n$, which is given by Mandelbrot & Wallis³⁴, and the values of H can be calculated by the least ordinary squares.

4. Results and discussions

In this section, we study the statistical properties of returns and absolute returns of the simulated data from the financial stock model by voter interacting system, the predictive data from the random jump time effective neural network model and the actual SHCI data

For the simulated prices, in Fig. 3, we do the computer simulation of the financial model for different values of the intensity λ ($\lambda = 5, 10, 15$) and the finite-range R ($R = 1, 2, 3$). We let the initial price S_0 be 5320, the price of the index SHCI on January 2, 2008. For the predictive price, in the random jump time effective neural network model, we input five kinds of stock prices of SHCI: daily open price, daily closed price, daily input layer is 5, the number of neural nodes in the hidden layer is 13 and the number of neural nodes in output layer is 1. In this paper, we take μ_j and σ_j to be the mean and the highest price, daily lowest price and daily trade volume; and one price of stock prices in the output layer: the closed price of the next trade day. The number of neural nodes in variance of reality historical data of SHCI, and let the intensity γ be 1/30. That is to

say, jump will happen 10 times a year in average. Moreover, we suppose that the values of vector $(\mu(t), \sigma(t))$ are (1, 1). We divide data into two sections: we use the data from 2003 to 2007 for training and forecast the SHCI from 2008 to 2009 for each trading day. In the experiment, we perform 15 times of the random jump time effective neural network model and report the averaged predictive data as the final predictive result. The empirical results show that the model exhibits stable training and predicting ability despite the random fluctuations from the improved algorithm. Fig. 4 presents the predictive data of the model and the actual data.

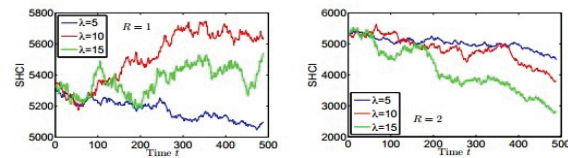


Fig. 3(a)

Fig. 3(b)

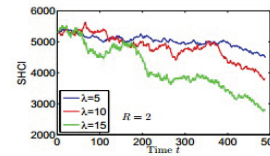


Fig. 3(c)

Fig. 3. Plots of simulation data of the financial model by finite-range voter system.

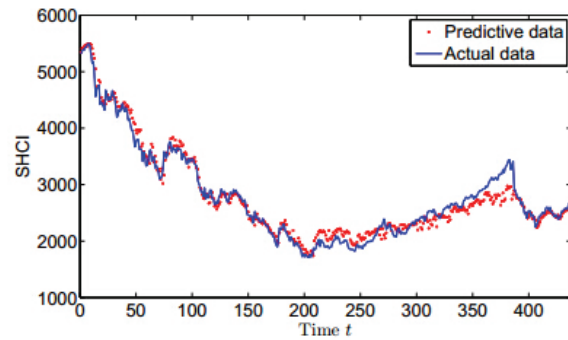


Fig. 4. SHCI forecasted by random jump time effective neural network.

We denote the daily price at time t by $S(t)$ ($t = 1, 2, \dots$), then we have the formula of stock logarithmic return^{29,31}

$$R(t) = \ln S(t+1) - \ln S(t). \quad (18)$$

The corresponding absolute return is defined as

$$R(t) = \ln S(t+1) - \ln S(t). \quad (19)$$

In the following, we begin to investigate and compare the returns and the absolute returns of the simulated data, the predictive data and the actual data by using the detrended fluctuation analysis (DFA) approach and the rescaled range (R/S) statistic. Table 1 and Table 2 show the basic statistical properties of the returns and the absolute returns.

Table 1. Returns statistics of the simulated data, the predictive data and the actual data.

R	λ	Mean	Variance	Skewness	Kurtosis	Minimum	Maximum
1	5	-8.8809e-5	1.9539e-6	1.8258e-2	7.5903e-1	-4.0000e-3	5.0000e-3
1	10	1.1550e-4	5.6737e-6	1.0601e-2	8.8434e-1	-8.2500e-3	8.0000e-3
1	15	8.3162e-4	9.8055e-6	-1.1185e-1	8.0303e-1	-1.0000e-2	9.7500e-2
2	5	-3.4907e-4	1.9365e-5	4.9578e-2	3.7003e-1	-1.2500e-2	1.2750e-2
2	10	-6.8480e-4	6.1716e-5	-1.8834e-1	3.8487e-2	-2.4500e-2	2.200e-2
2	15	-1.3280e-3	1.1827e-4	1.4396e-2	-6.0905e-1	-2.5000e-2	2.5000e-2
3	5	3.7833e-4	7.7932e-5	6.3248e-2	-1.4102e-1	-2.3750e-2	2.3500e-2
3	10	-3.7422e-4	1.9203e-4	-3.9950e-2	-7.5850e-1	-2.5000e-2	2.5000e-2
3	15	-6.1704e-4	7.6169e-4	1.4385e-1	-5.8214e-1	-5.0000e-2	5.0000e-2
predictive data		-1.4110e-3	8.2033e-4	4.5136e-1	2.1941	-8.5685e-1	1.4191e-1
actual data		-1.2906e-3	6.4592e-4	-4.1636e-1	3.1190	-1.0063e-1	1.0063e-1

Table 2. Absolute returns statistics of the simulated data, the predictive data and the actual data.

R	λ	Mean	Variance	Skewness	Kurtosis	Minimum	Maximum
1	5	1.0539e-3	8.4881e-7	1.2842	1.4928	0.0000	5.0000e-3
1	10	1.8085e-3	2.4096e-6	1.3946	2.0432	0.0000	8.2500e-3
1	15	2.3963e-3	4.0584e-6	1.3883	1.9593	0.0000	9.7500e-2
2	5	3.4671e-3	7.4413e-6	1.2401	1.2645	0.0000	1.2750e-2
2	10	6.2402e-3	2.3165e-5	1.0006	6.2121e-1	0.0000	2.4500e-2
2	15	9.0210e-3	3.8492e-5	6.2217e-1	-5.0543e-1	0.0000	2.5000e-2
3	5	7.0200e-3	2.8693e-5	8.9394e-1	2.2112e-1	0.0000	2.3500e-2
3	10	1.1405e-2	6.1832e-5	4.4404e-1	-9.9987e-1	0.0000	2.5000e-2
3	15	2.2210e-2	2.6776e-4	5.4470e-1	-9.9891e-1	0.0000	5.0000e-2
predictive data		2.0687e-2	3.9349e-4	1.7539	4.1275	1.8188e-5	1.4191e-1
actual data		1.8583e-2	3.0155e-4	2.1056	8.2304	3.7712e-5	1.0063e-1

According to the DFA method, we analyze the statistical behaviors of returns of the simulated data, the actual data and the predictive data, see Fig. 5 and Table 3 for empirical results. Specially, the small figures in the upper left and lower right corner of Fig. 5 represent the scaling exponent α value of the actual data and its closest situation. The empirical research shows that the smallest exponent corresponds to the simulated data when $\{R=1, \lambda=5\}$, and the largest one is for the predictive index. In Table 3, the results reveals that five α values are bigger than 0.5, including the actual data, which indicates that these five corresponding time series are long-range auto-correlated according to the concept

of α . And when $\{R=2, \lambda=15\}$, the result of simulation by the finite-range financial model is the closest to the actual situation.

By applying the method of R/S test, the exponent H of the returns for the simulated data, the actual data and the predictive data are investigated in Fig. 6 and Table 4. The H of three groups' data are all larger than 0.5, which indicate that these three group financial time series are all long-term memory sequences. In addition,

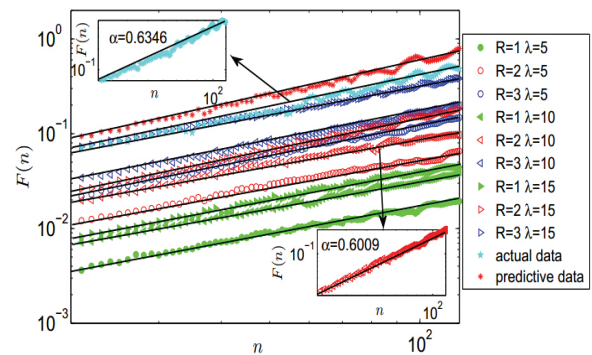


Fig. 5. The plot of DFA test statistics of returns for the simulation data, the actual data and the predictive data

the time series which is closest to the actual situation is the simulated series by finite-range financial model when $R=1$ and $\lambda=10$. Furthermore, Fig. 7 and Table 4 show the exponent H of the absolute returns for the simulated data, the actual data and the predictive data. All the values are bigger than 0.5, which means all the time sequences are persistent and have the quantification of long-term memory.

Table 3. α values of the simulated data, the predictive data and the actual data.

R	λ	α	β
1	5	0.4150	-3.5411
1	10	0.4533	-3.3302
1	15	0.4700	-3.2773
2	5	0.4603	-3.1042
2	10	0.4555	-2.8738
2	15	0.6009	-2.9258
3	5	0.5504	-2.9162
3	10	0.5178	-2.7212
3	15	0.4477	-2.3100
actual	data	0.6346	-2.5634
predictive	data	0.6876	-2.4736

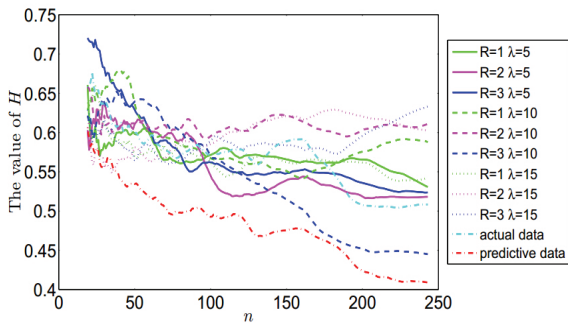


Fig. 6. Fluctuations of Hurst exponents of the returns.

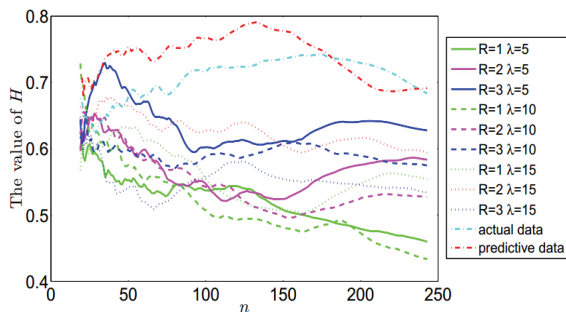


Fig. 7. Fluctuations of Hurst exponents of the absolute returns.

Table 4. Hurst exponents of the simulated data, the predictive data and the actual data.

R	λ	returns	absolute returns
1	5	0.6605	0.6072
1	10	0.6793	0.7286
1	15	0.6604	0.6327
2	5	0.6357	0.6529
2	10	0.6581	0.6551
2	15	0.6293	0.6785
3	5	0.7203	0.7294
3	10	0.6521	0.6468
3	15	0.6331	0.6481
actual	data	0.6063	0.7425
predictive	data	0.6770	0.7909

5. Conclusion

In the present paper, we model a stock price process by applying the finite-range voter system to study the statistical behaviors of fluctuations for the index SHCI. Further we develop the artificial neural network model based on random jump time effective function to forecast the index SHCI. And we compare the statistical properties of the returns and the absolute returns for the simulated data, the actual data and the predictive data by using DFA and classic R/S analysis. We see that scaling exponents of DFA method for these three groups' data are all larger than 0.5, indicating the long-range correlation, and when $\{R=2, \lambda=15\}$, the result of simulated data by the price model is the closest to the actual situation. Similarly, by R/S analysis, when $\{R=1, \lambda=10\}$, the simulated time series is the closest to the actual index. The research results may imply that, from a micro perspective, the ANN model is closer to the actual situation, yet the financial model derived from the voter interacting system may has better performance from the macro point of view.

Acknowledgements

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

References

1. W. Fang and J. Wang, Statistical Properties and Multifractal Behaviors of Market Returns by Ising Dynamic Systems, *Int. J. Mod. Phys. C* **23** (2012) 1250023.
2. F. Wang and J. Wang, Statistical analysis and forecasting of return interval for SSE and model by lattice percolation system and neural network, *Computers Industrial Engineering* **62** (2012) 198-205.

3. J. Wang and S. Deng, Fluctuations of interface statistical physics models applied to a stock market model, *Nonlinear Anal.: Real World Appl.* **9** (2008) 718-723.
4. T. S. Wang and J. Wang, Statistical analysis by statistical physics model for the stock markets, *Int. J. Mod. Phys. C* **20** (2009) 1547-1562.
5. J. Wang, Q. Y. Wang and J. G. Shao, Fluctuations of stock price model by statistical physics systems, *Math. Comput. Model.* **51** (2010) 431-440.
6. J. H. Zhang and J. Wang, Modeling and Simulation of the Market Fluctuations by the Finite Range Contact Systems, *Simul. Model. Pract. Theory* **18** (2010) 910-925.
7. J. H. Zhang, J. Wang and J. G. Shao, Finite-range contact process on the market return intervals distributions, *Adv. Complex Syst.* **13** (2010) 643-657.
8. M. F. Chen, *From Markov Chains to Non-Equilibrium Particle Systems* (World Scientific, 1992).
9. R. Durrett, *Lecture Notes on Particle Systems and Percolation* (Wadsworth & Brooks, California, 1988).
10. R. S. Ellis, *Entropy, Large Deviations and Statistical Mechanics* (Springer-Verlag, New York, 1985).
11. G. Grimmett, *Percolation* (Springer-Verlag, Berlin, 1999).
12. T. M. Liggett, *Interacting Particle Systems* (Springer-Verlag, New York, 1985).
13. T. M. Liggett, *Stochastic Interacting Systems: Contact, Voter and Exclusion Processes* (Springer-Verlag, New York, 1999).
14. R. Gaylord and P. Wellin, *Computer simulations with Mathematica: explorations in the physical, biological and social science* (Springer-Verlag, New York, 1995).
15. K. Ilinski, *Physics Of Finance: Gauge Modeling in Non-equilibrium Pricing* (John Wiley, 2001).
16. T. C. Mills, *The Econometric Modeling of Financial Time Series*, 2nd edn. (Cambridge University Press, Cambridge, 1999).
17. Z. Zheng, *Matlab programming and the applications*, (China Railway Publishing House, Beijing, 2003).
18. E. M. Azoff, *Neural network time series forecasting of financial market* (Wiley, New York, 1994).
19. L. E. Calvet and A. J. Fisher, *Multifractal Volatility: Theory, Forecasting, and Pricing*, Academic Press Advanced Finance, Academic Press, 2008).
20. H. Demuth and M. Beale, *Neural Network Toolbox: For Use with MATLAB*, fifth edn. (The Math Works, Inc., Natick, MA, 1998).
21. V. S. Desai and R. Bharati, The efficiency of neural networks in predicting returns on stock a bond indices, *Decis. Sci.* **29** (1998) 405-425.
22. Z. Liao and J. Wang, Forecasting model of global stock index by stochastic time effective neural network, *Expert Syst. Appl.* **37** (2010) 834-841.
23. F. J. Liu and J. Wang, Fluctuation prediction of stock market index by Legendre neural network with random time strength function, *Neurocomputing* **83** (2012) 12-21.
24. H. F. Liu and J. Wang, Integrating Independent Component Analysis and Principal Component Analysis with Neural Network to Predict Chinese Stock Market, *Math. Probl. Eng.* **2011** (2011) 382659.
25. T. H. Roh, Forecasting the volatility of stock price index, *Expert Syst. Appl.* **33** (2007) 916-922.
26. O .B. Nielsen and N. Shephard, Power and bipower variation with stochastic volatility and jumps, *J. Financ. Econ.* **2** (2004) 1-48.
27. D. Pirino, Jump detection and long range dependence, *Phys. A* **388** (2009) 1150-1156.
28. F. Black and M. Scholes, The pricing of options and corporate liabilities, *J. Polit. Econ.* **81** (1973) 637-654.
29. D. Lamberton and B. Lapeyre, *Introduction to Stochastic Calculus Applied to Finance* (Chapman and Hall/CRC, London, 2000).
30. S. R. Pliska, *Introduction to Mathematical Finance*, (Blackwell, USA, 2000).
31. S. M. Ross, *An Introduction to Mathematical Finance*, (Cambridge University Press, Cambridge, 1999).
32. P. Grau-Carles, Long-range power-law correlations in stock returns, *Phys. A* **299** (2001) 521-527.
33. H. E. Hurst, Long Term storage capacity of reservoirs, *Trans. Am. Soc. Civ. Eng.* **116** (1951) 770-799.
34. B. Mandelbrot and J. R. Wallis, Robustness of the rescaled range R/S in the measurement of noncyclic long run statistical dependence, *Water Resour. Res.* **5** (1969) 967-988.
35. V. Teverovsky, M. S. Taquu and W. Willinger, A critical look at Lo's modified R/S statistic, *J. Stat. Plan. Inference* **80** (1999) 211-227.