Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 3967525, 14 pages
https://doi.org/10.1155/2018/3967525

Research Article

A Bimodel Algorithm with Data-Divider to Predict Stock Index

Zhaoyue Wang, Jinsong Hu, and Yongjie Wu

School of Computer Science & Engineering, South China University of Technology, Guangzhou, Guangdong 510006, China

Correspondence should be addressed to Jinsong Hu; cshjs@scut.edu.cn

Received 12 August 2017; Accepted 30 January 2018; Published 5 March 2018

Academic Editor: Daniela Boso

There is not yet reliable software for stock prediction, because most experts of this area have been trying to predict an exact stock index. Considering that the fluctuation of a stock index usually is no more than 1% in a day, the error between the forecasted and the actual values should be no more than 0.5%. It is too difficult to realize. However, forecasting whether a stock index will rise or fall does not need to be so exact a numerical value. A few scholars noted the fact, but their systems do not yet work very well because different periods of a stock have different inherent laws. So, we should not depend on a single model or a set of parameters to solve the problem. In this paper, we developed a data-divider to divide a set of historical stock data into two parts according to rising period and falling period, training, respectively, two neural networks optimized by a GA. Above all, the data-divider enables us to avoid the most difficult problem, the effect of unexpected news, which could hardly be predicted. Experiments show that the accuracy of our method increases 20% compared to those of traditional methods.

1. Introduction

People have been trying to predict stock prices or indexes since a successful prediction means huge income. However, from the point of view of system theory, the formation mechanism of stock price is a nonlinear system with a high degree of complexity [1], so it is difficult to predict. With the development of neural networks, its strong nonlinear fitting ability has shown huge potential in stock prediction. According to predicted objects about stocks, there are two primary methods to construct neural networks models for stock prediction.

Most scholars prefer the first method, predicting an exact stock price or index (exact prediction for short). Qiu et al. tried to predict the return of Nikkei 225 index by combining the performance of the accuracy of prediction and run time and concluded that the hybrid of genetic algorithm (GA) and backpropagation (BP) provides more accurate forecasting of future values than other prediction models [2] do. In J. Wang and J. Wang's paper, they introduced the stochastic time effective neural network with principal component analysis model to forecast the indexes of Shanghai Stock Exchange (SSE), Hong Kong Hang Seng 300 Index (HS300), Standard & Poor's 500 Index (S&P500), and Dow Jones Industrial Average Index (DJIA). Their results show that the proposed neural network model improves the accuracy of forecasting results [3]. In Al-Hnaity and Abbod's paper, they proposed a hybrid ensemble model based on BP neural network and EEMD to predict FTSE100 closing price. The results show that this method can effectively reduce the stock price prediction error [4]. The accuracy of this kind of prediction methods seems very high, but it is still not enough to help stock investors to make a decision since the fluctuation of a stock price or index usually is no more than 1% in a day (that is to say, prediction accuracy should be up to 99.5%).

The others, a few scholars, prefer the rise or fall of a stock index (trend prediction for short) to its exact numerical value. In Wang et al.'s paper, they used an NNK-ELM model which is based on market news and stock prices to forecast Hong Kong stock indexes. The prediction results show that the proposed method is better than the traditional BP algorithm in trend prediction accuracy [5]. Sun and Gao directly used the forecast accuracy of the stock trend as the criterion of their model. They proposed a hybrid BP neural network combining adaptive particle swarm optimization algorithm (HBP-PSO) to predict the stock price of "Zhong Guo Yi Yao" (600056). The results support that the trend prediction accuracy of HBP-PSO is better than that of the simple neural

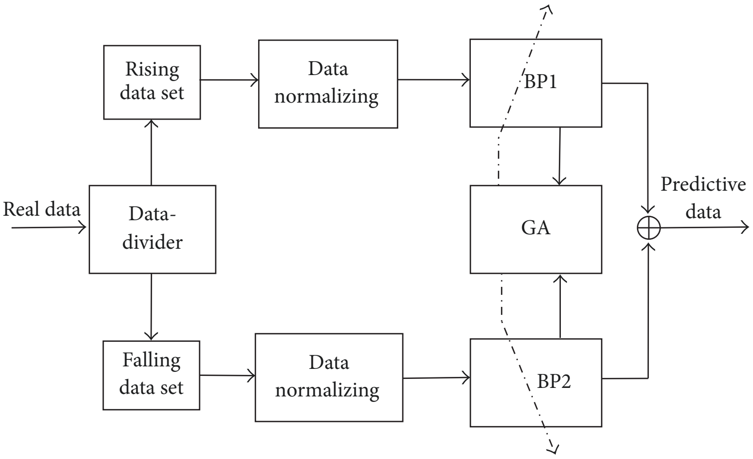

Figure 1: The general framework of BADD. BP1 and BP2 are two BP neural networks. GA is a genetic algorithm.

network [6]. Using this criterion can ensure the prediction accuracy of the up and down signals of stock price to reduce the possibility of trading errors.

For most stock investors, forecasting whether a stock price or index will rise or fall (trend prediction) is enough to make a decision, buying or selling. Therefore, the second method is more practical than the first. However, the second way does not yet work very well. One important reason is that different periods of a stock market, for example, rising period and falling period, have different inherent laws and scholars usually use a single neural networks model and the same set of parameters to deal with the different periods [5, 6]. Another reason, the most difficult problem, is the effect of unexpected news, which could hardly be predicted. Recently, some scholars try to deal with the problem through collecting and mining present financial information [7, 8]. However, the effect of the information is difficult to quantize and the relationship between present information and future news is uncertain.

In this paper, we developed a data-divider to divide a set of historical stock data into two parts according to rising period and falling period, training, respectively, two neural networks optimized by a GA. Above all, the data-divider enables us to avoid the most difficult problem, the effect of unexpected news. Experiments show that the accuracy of our method increases 20% than those of traditional methods.

2. The General Frameworks of the Bimodel Algorithm with Data-Divider

The whole framework of the bimodel algorithm with data-divider (BADD) is shown as Figure 1. In Figure 1, arrow lines show directs of data flows. Arrow dotted lines indicate adjusting parameters. In the stage of training neural networks, GA optimizes parameters of BP1 and BP2 separately. After the stage, GA does not work.

In the following paragraphs, we will give details of all components of Figure 1 sequentially.

3. Input and Output

The input of Figure 1, real historical data, include current and previous two day's closing values and volumes and current day's KDJ index, MACD index, and RSI. The output of Figure 1 is predictive data, tomorrow's closing value.

According to Joseph E. Granville's theoretical research on volume-price relationship, volume is the leading indicator of stock price, so this paper uses volume as an important factor of the input. Based on the practical experience of stock exchange, too "old" stock data usually is insignificant for stock forecasting. Therefore, this paper chooses previous two days' closing price and volume as two of input variables.

The Stochastic Oscillator (KDJ), including K value, D value, and J value, is a momentum indicator [9]. When K value, D value, and J value are all above 50, the stock market is in a bullish mood. When they are all below 50, the stock market is in a bearish mood. This paper treats KDJ as an input variable.

The moving average convergence (MACD) is widely used for medium- and long-term stock forecasting. Observing a MACD line, we can easily find out the bullish signal and the bearish signal. Therefore, this paper chooses the MACD value as an input variable.

The Relative Strength Index (RSI) defines a trading rule to measure the speed and change of price movements [9]. Therefore, this paper chooses RSI as part of the input.

As three typical reference indicators, MACD, RSI, and KDJ have become an essential reference standard for many stock traders. So, these three indicators also have psychological significance to a certain extent.

4. Data-Divider

As two different stages, rising period and falling period have different inherent laws. If we use the same neural network with the same parameters to predict stock, these complex data will easily cause the overfitting problems and reduce the neural network prediction accuracy. So, this paper proposes

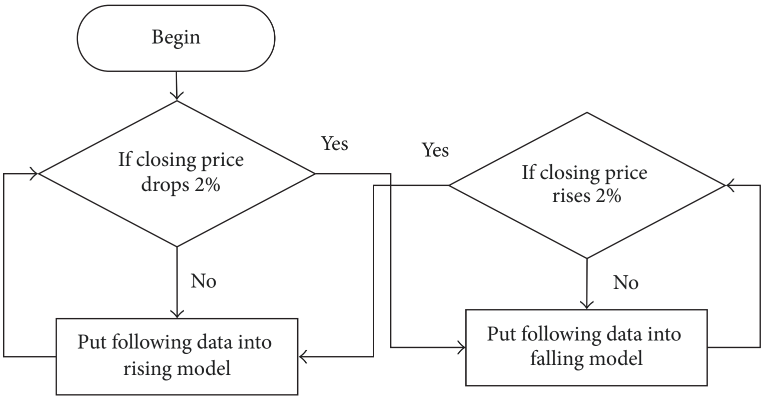
Begin
If closing price drops 2%
Yes
No
Put following data into rising model
Yes
If closing price rises 2%
No
Put following data into falling model

Figure 2: The data-divider.

a data-divider to divide a historical data set into two different training data sets and use two models to train separately.

However, these data should not be roughly divided once and for all since the accidental fluctuation of stock markets complicate stock data. A long stock period include some short rising periods and falling periods, which may be nested within each other. Further, there is always a long-term rising with some small range fluctuations which will not affect the general trend. The falling period is so too.

The presented data-divider, as Figure 2 shows, can reduce the complexity of a data set and improve the prediction accuracy of neural networks. From this figure, we can see that those data in a small range of fluctuation (around 2%) are retained, for the reason that stock with absolutely monotonic trend without any fluctuation does not exist in actual stock market. Additionally, the data-divider “ignores” the data of big unexpected news making violent changes (the index drop or rise over 2%). That is to say, we do not try to predict the effect of big unexpected news but predict the following trend.

5. Data Normalization

Sigmoid function is used as the activation function between the input layer and the hidden layer. The formula of this function is

$$f(x) = \frac{1}{1 + e^{-x}}. \tag{1}$$

Because the range of this function is between 0 and 1 and when the range is $(0, 0.2)$ and $(0.8, 1.0)$, the trend of this function is smooth. However, stock data are very complicated. For example, trading volume could be very large, even reaching the level of 10^9, but the value of MACD is under the level of 10^2. After normalization, these data are unified into the same reference frame, which makes the calculation of next step easier.

We normalized input data to the range of $(0.2, 0.8)$. The normalization function is

$$y_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} * (R_{\max} - R_{\min}) + R_{\min}. \tag{2}$$

y_i is the result; x_i is the value before normalizing; $R_{\max}$ is the max value of the range.

6. The Approach to Combine BPNN and GA

BP neural network (hereinafter called BPNN) can achieve good results in stock forecasting [10–13]. On the one hand, BPNN may fast converge to a local optimal solution, but it can not ensure a globally optimal solution because of the problem of the local optimum. On the other hand, genetic algorithm (hereinafter called GA) can search the whole space, but its convergence rate is slow. GA-BP algorithm combines the two methods to fast obtain the global optimal solution. There are two different approaches to combine BPNN and GA.

First approach uses BPNN to optimize several different initial values and then obtains several local extreme values. GA combines these extreme values to get better values. However, this approach shows serious premature convergence and easily falls into local optimal solution [14].

Second approach uses a GA to obtain some low-precision solutions firstly. Starting from these solutions, BPNN makes local search to obtain some high-precision solutions. From the point of view of neural networks, the search of the GA is to obtain a group of better initial weights of the neural network than random initial weights. The local search by BPNN is to revise this group of initial weights and obtain the global optimal result. As a result, this approach got a better global solution than the first approach. Therefore, in this paper, we used the second approach. Figure 3 is the flow chart of the second approach.

6.1. Momentum Factor.

6.1. Momentum Factor. Using momentum factor can not only minimize the error of the network, which easily falls into local minimum, but also speed up the convergence process of the algorithm [15]. After using momentum factor, the change of weight is

$$\begin{aligned}\Delta\omega_{ij}(n+1) = \alpha * \Delta\omega_{ij}(n) + \mu * \delta_i(n+1) \\ * o_j(n+1).\end{aligned} \tag{3}$$

In this formula, α is the momentum factor, with a domain of $[0,1]$. If $\Delta\omega_{ij}(n)$ and $\Delta\omega_{ij}(n+1)$ are both positive or negative, this formula will accelerate convergence; if it is not the case, the speed of convergence will be reduced and greater stability will be achieved without falling into a local minimum. A dynamic method is adopted in choosing an appropriate momentum factor:

$$\alpha = a + \frac{b}{n}, \tag{4}$$

where n is the number of iterations. A large momentum factor can speed up convergence in the BP algorithm. As the number of iterations increases, the momentum factor decreases. Therefore, the volatility of results is minimized.

6.2. L2 Regularization. The L2 Regularization can reduce weights, reducing the complexity of the network. This method can avoid overfitting problems caused by high network complexity and increases the accuracy of prediction. The formula of L2 Regularization is

$$C = C_0 + \frac{\lambda}{2n}\sum \omega^2, \tag{5}$$

where C_0 is the cost function and n is the number of input vectors.

6.3. Details of the GA. Since this paper takes into account not only the error of the prediction results but also the trend prediction accuracy, the following fitness function is proposed:

$$F = \frac{1}{\sum_{i=1}^{n}\left(y_i - x_i\right)^2} + \frac{\sum_{i=1}^{n} f\left(y_i, x_i\right)}{n}, \tag{6}$$

where y_i is the ideal output and x_i is the prediction result. $f(y_i, x_i)$ represents the result of trend prediction.

This paper uses the adaptive crossover rate and mutation rate algorithm mentioned in [16]. The formulae of this algorithm are

$$P_c = \begin{cases} \dfrac{P_{c\max} + P_{c\min}}{2} + \dfrac{P_{c\max} - P_{c\min}}{2}\cos\left(\dfrac{f' - f_{\text{avg}}}{f_{\max} - f_{\text{avg}}}\pi\right), & f' \geq f_{\text{avg}} \\ P_{c\max}, & f' < f_{\text{avg}}, \end{cases}$$
$$P_m = \begin{cases} \dfrac{P_{m\max} + P_{m\min}}{2} + \dfrac{P_{m\max} - P_{m\min}}{2}\cos\left(\dfrac{f - f_{\text{avg}}}{f_{\max} - f_{\text{avg}}}\pi\right), & f \geq f_{\text{avg}} \\ P_{m\max}, & f < f_{\text{avg}}. \end{cases} \tag{7}$$

$P_{c\max}$ and $P_{c\min}$ represent the upper and lower limits of the value of crossover rate, respectively; $P_{m\max}$ and $P_{m\min}$ represent the upper and lower limits of the mutation rate value, respectively; $f_{\max}$ is the maximum fitness of the population; f_{avg} is the average fitness; f' is the fitness of the individual currently in crossover process. This algorithm increases the crossover rates and mutation rates of individuals whose fitness values lie between f_{avg} and $(f_{\max} + f_{\text{avg}})/2$ and decreases those of individuals whose fitness values lie between $(f_{\max} + f_{\text{avg}})/2$ and $f_{\max}$. As a result, dominant individuals are retained, and disadvantaged individuals are changed.

The remainder stochastic sampling with replacement (RSSR) selection operator [17] is used and its basic steps are as follows.

Let the population size be N and the sum of all individual fitness be sum. f_i is defined as the fitness of the ith individual, and the survival expectation of the ith individual has the following explicit form:

$$n_i = \left\lfloor \frac{N * f_i}{\text{sum}} \right\rfloor. \tag{8}$$

Let $f_i - (n_i * \text{sum})/N$ be the new fitness of the ith individual. Remaining individuals are generated by ordinary roulette method.

Compared with conventional roulette methods, this selection operator reduces the selection error so that individuals with fitness above average can survive to the next generation and increases population diversity. Therefore, the premature convergence problem of GA can be improved.

An arithmetic crossover operator [18] is used. In the case of longer chromosome, this method will generate better individual and avoid massive destruction of chromosome. Meanwhile, its computing efficiency is high. So, this method fits real-coded chromosomes better. The main operation is

$$\begin{aligned} X' &= c * X + (1 - c) * Y, \\ Y' &= c * Y + (1 - c) * X. \end{aligned} \tag{9}$$

In these two formulae, c is a random parameter and X or Y means a gene (a real number) on the chromosome of the individual to crossover before crossover. X' or Y' means the gene after crossover.

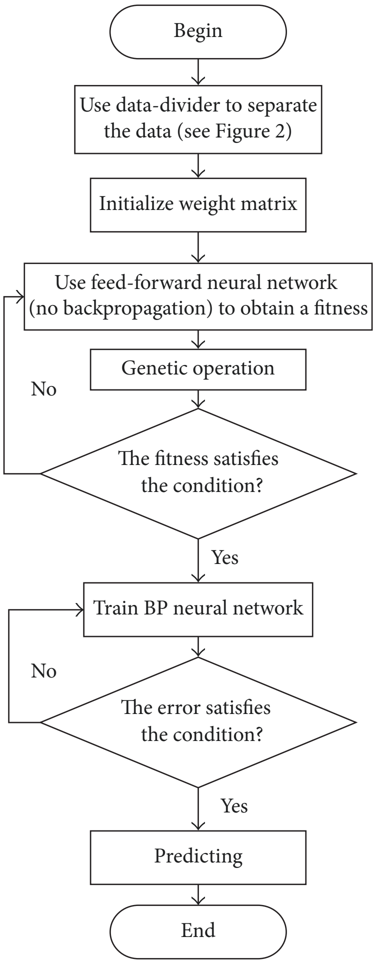
Begin
Use data-divider to separate the data (see Figure 2)
Initialize weight matrix
Use feed-forward neural network (no backpropagation) to obtain a fitness
Genetic operation
No
The fitness satisfies the condition?
Yes
Train BP neural network
No
The error satisfies the condition?
Yes
Predicting
End

Figure 3: The second approach to combine BPNN and GA.

Mutation operator plays a vital role in maintaining population diversity. A suitable mutation operator can greatly improve premature convergence problem of genetic algorithm. According to Bitwise mutation operator, for every gene on a chromosome, a random value r is calculated, if r is larger than the probability of mutation, another random number is calculated and added to the gene value. A mutation probability that is too high or too low will cause problem like heavy loss of excellent genes or premature convergence. Therefore, to keep a proper mutation probability, this paper chooses adaptive mutation probability.

7. Simulations

In this paper, the data-divider can divide a stock data set into two different data subsets, the rising one and the falling one. So, our system includes two models, rising model and falling model. The falling data set trains falling model and the rising data set trains the rising model. The following experiments will test them, respectively. A conventional method (called single model method in this paper), which uses the same neural network with the same parameter to train historical data, was used for comparison.

This paper focuses on actual stock exchange, which is nothing more than buying or selling. It is an obvious buying signal for the stock traders when the price trend of a stock is rising. And when the price trend of the stock is falling, the stock trader will sell it. In fact, the stock trader is more concerned about the future price trend of a stock, so this paper uses not only average error and maximum error as the criterion of experimental results, but also the accuracy of trend prediction as the criterion.

The formula of the error calculation is

$$g(x_i, y_i) = \frac{|y_i - x_i|}{y_i}, \tag{10}$$

where y_i is the actual closing price and x_i is the prediction result. The following formula was used to judging the trend prediction result

$$f(x_i, y_i) = \begin{cases} \text{true}, & (x_i - y_{i-1}) * (y_i - y_{i-1}) > 0 \\ \text{true}, & |x_i - y_{i-1}| < e, \ y_i = y_{i-1}, \ 0 < e < 10^{-5} \\ \text{false}, & (x_i - y_{i-1}) * (y_i - y_{i-1}) < 0. \end{cases} \tag{11}$$

Table 1: Results of the rising model and the single model.

	Average error	Maximum error	Trend accuracy
Rising model	0.00548318728116	0.0232548639991	0.745454545455
Single model	0.00688573345702	0.031534665601	0.6

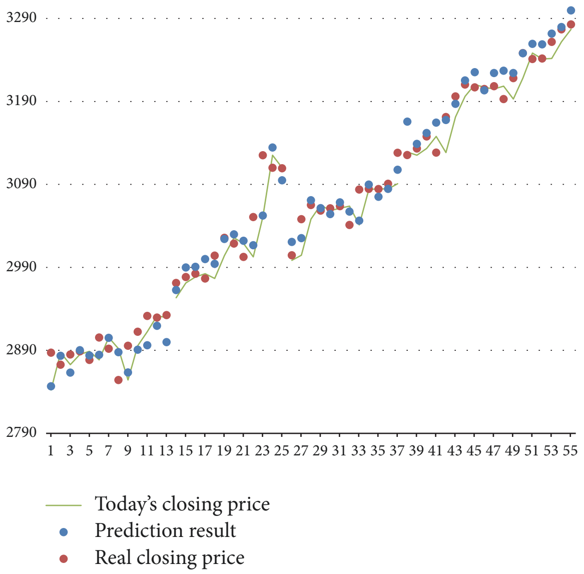

Figure 4: Rising model.

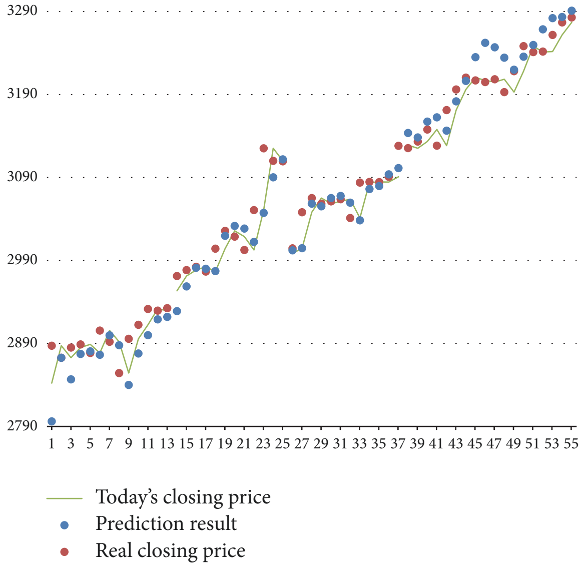

Figure 5: Single model.

7.1. Trend Forecast for Rising Model in a Long Period of Index. The SSE index from 13 October 2009 to 29 January 2016 (1533 groups of samples) was selected as a training set. SSE indexes in four long periods which match the rising model from 15 June 2016 to 29 November 2016 were selected as test sets.

With the same GA-BP algorithm, the results of the rising model (one part of bimodel) and the single model are, respectively, listed in Table 1.

In Figures 4 and 5, green lines represent closing prices, blue points represent prediction results and red points represent real closing prices. For each day, if the red point and the blue point are on the same side, it means that this trend prediction is accurate.

From Table 2, we can easily see that, in four periods, trend prediction accuracy of the rising model method is about 20% higher than that of the single model method. These results

Table 2: Real data and prediction results of the rising model and the single model.

Time	Real closing price	Rising model	Single model
		Prediction result	Prediction result
2016/6/15	2887.21	2846.836	2796.163
2016/6/16	2872.82	2883.434	2872.760
2016/6/17	2885.11	2863.198	2846.830
2016/6/20	2888.81	2890.223	2877.362
2016/6/21	2878.56	2883.827	2880.484
2016/6/22	2905.55	2884.713	2876.291
2016/6/23	2891.96	2905.169	2899.756
2016/6/24	2854.29	2887.931	2887.805
2016/6/27	2895.7	2863.445	2840.101
2016/6/28	2912.56	2890.929	2877.978
2016/6/29	2931.59	2896.139	2900.084
2016/6/30	2929.61	2919.562	2919.075
2016/7/1	2932.48	2900.068	2921.965
2016/8/2	2971.28	2962.792	2928.970
2016/8/3	2978.46	2989.999	2958.781
2016/8/4	2982.43	2990.815	2981.247
2016/8/5	2976.7	3000.188	2980.018
2016/8/8	3004.28	2994.374	2977.334
2016/8/9	3025.68	3024.378	3019.758
2016/8/10	3018.75	3029.842	3031.766
2016/8/11	3002.64	3022.079	3028.418
2016/8/12	3050.67	3016.695	3012.348
2016/8/15	3125.2	3052.524	3047.272
2016/8/16	3110.04	3134.624	3090.307
2016/8/17	3109.55	3095.069	3112.005
2016/9/30	3004.7	3020.829	3002.374
2016/10/10	3048.14	3025.421	3004.802
2016/10/11	3065.25	3070.946	3058.432
2016/10/12	3058.5	3061.450	3055.368
2016/10/13	3061.35	3054.349	3065.218
2016/10/14	3063.81	3068.503	3067.874
2016/10/17	3041.17	3057.303	3059.720
2016/10/18	3083.88	3046.384	3038.485
2016/10/19	3084.72	3089.847	3076.174
2016/10/20	3084.46	3075.266	3079.818
2016/10/21	3090.94	3084.724	3093.903
2016/10/24	3128.25	3107.663	3101.333
2016/11/4	3125.32	3165.698	3143.740
2016/11/7	3133.33	3138.868	3138.245
2016/11/8	3147.89	3152.136	3157.431
2016/11/9	3128.37	3164.585	3162.472
2016/11/10	3171.28	3167.704	3146.561
2016/11/11	3196.04	3187.292	3181.853
2016/11/14	3210.37	3215.444	3206.459
2016/11/15	3206.99	3225.372	3235.153
2016/11/16	3205.06	3203.528	3252.325
2016/11/17	3208.45	3224.321	3246.904
2016/11/18	3192.86	3227.154	3234.399
2016/11/21	3218.15	3224.297	3219.953

Table 2: Continued.

Time	Real closing price	Rising model	Single model
		Prediction result	Prediction result
2016/11/22	3248.35	3248.086	3235.721
2016/11/23	3241.14	3259.562	3249.783
2016/11/24	3241.74	3258.817	3268.720
2016/11/25	3261.94	3271.736	3282.042
2016/11/28	3277	3279.733	3283.476
2016/11/29	3282.92	3299.852	3291.134

Table 3: Results of the rising model and the single model.

	Average error	Maximum error	Trend accuracy
Rising model	0.00961777468253	0.0291690512821	0.755102040816
Single model	0.0101503712662	0.0318118420168	0.591836734694

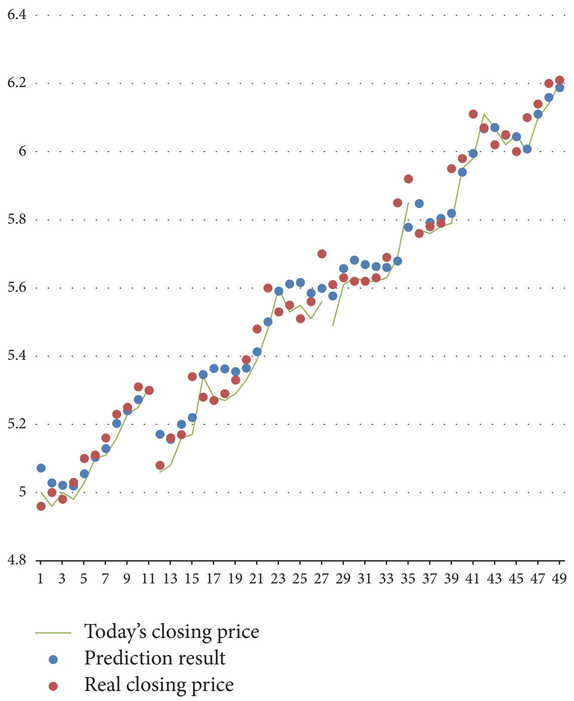

Figure 6: Rising model.

prove that rising model can obtain not only lower prediction error but also higher accuracy of trend prediction. In the figure of single model, for most days, the red point is close to blue point but they are on the different side of the green line, which means prediction failure. So, the single model method may mislead the operation of the stock traders. Oppositely, in the figure of rising model, even if two points are far away, they are still on the same side of the green line, which means prediction success. So rising model, that is, the bimodel method, can greatly reduce misleading signals.

7.2. Trend Forecast for Rising Model in the Long Period of Individual Share. This paper selected the stock of industrial and commercial bank of China (ICBC) from 31 March 2008 to 07 April 2017 (2176 sets of data) as an experimental training set and selected the stock of ICBC from four long periods which match the rising model from 17 May 2017 to 12 October 2017 as test sets. The test sets have 49 groups.

With the same GA-BP algorithm, the result of the rising model method and that of the single model method are shown in Table 3.

In Figures 6 and 7, green lines represent closing prices, blue points represent prediction results, and red points represent real closing prices. For each day, if the red point and the blue point are on the same side, it means that this trend prediction is accurate.

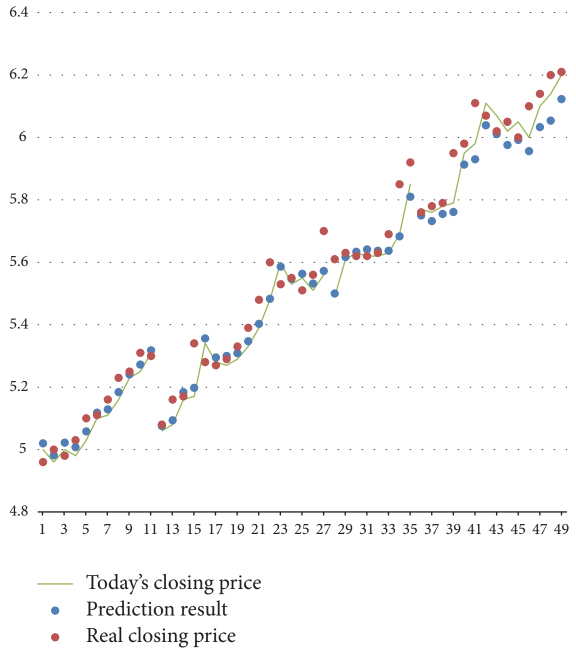

Figure 7: Single model.

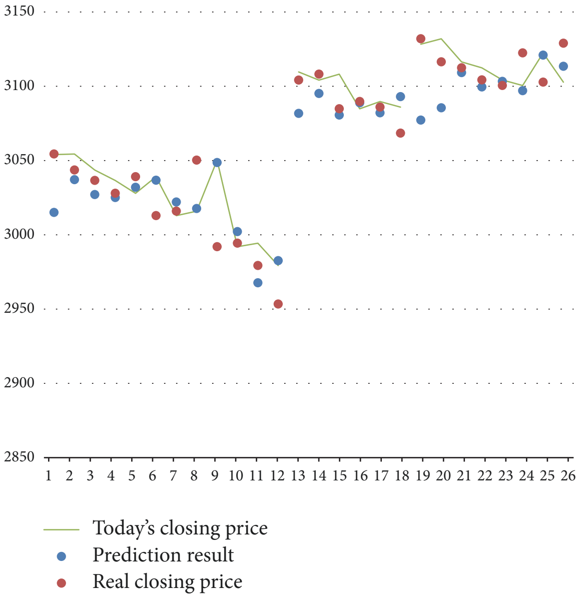

Figure 8: Falling model.

From Tables 2 and 4, we can know that rising model can improve trend prediction accuracy not only in stock index but also in individual share. This model can keep error in a low level.

7.3. Trend Forecast for Falling Model in the Long Period of Index. This paper selected the SSE index from 13 October 2009 to 29 January 2016 (1533 sets of data) as the experimental training set and selected the SSE index from three long periods which match the falling model from 15 July 2016 to 03 November 2016 as test sets. The test sets have 26 groups.

With the same GA-BP algorithm, the result of the falling model (another part of bimodel) method and that of the single model method are shown in Table 5.

In Figures 8 and 9, green lines represent closing prices, blue points represent prediction results, and red points represent real closing prices. For each day, if the red point and the blue point are on the same side, it means that this trend prediction is accurate.

Table 4: The real data and prediction results of the rising model and the single model.

Time	Real closing price	Rising model	Single model
		Prediction result	Prediction result
2017/5/17	4.96	5.072	5.020
2017/5/18	5	5.028	4.982
2017/5/19	4.98	5.021	5.022
2017/5/22	5.03	5.019	5.008
2017/5/23	5.1	5.055	5.058
2017/5/24	5.11	5.103	5.118
2017/5/25	5.16	5.129	5.129
2017/5/26	5.23	5.203	5.184
2017/5/31	5.25	5.240	5.241
2017/6/1	5.31	5.273	5.272
2017/6/2	5.3	5.300	5.318
2017/7/12	5.08	5.171	5.076
2017/7/13	5.16	5.156	5.094
2017/7/14	5.17	5.200	5.184
2017/7/17	5.34	5.220	5.198
2017/7/18	5.28	5.346	5.356
2017/7/19	5.27	5.364	5.295
2017/7/20	5.29	5.363	5.300
2017/7/21	5.33	5.355	5.309
2017/7/24	5.39	5.365	5.347
2017/7/25	5.48	5.413	5.403
2017/7/26	5.6	5.501	5.483
2017/7/27	5.53	5.591	5.586
2017/7/28	5.55	5.612	5.547
2017/7/31	5.51	5.616	5.563
2017/8/1	5.56	5.585	5.532
2017/8/2	5.7	5.599	5.572
2017/8/15	5.61	5.577	5.500
2017/8/16	5.63	5.657	5.617
2017/8/17	5.62	5.682	5.634
2017/8/18	5.62	5.669	5.641
2017/8/21	5.63	5.663	5.637
2017/8/22	5.69	5.660	5.637
2017/8/23	5.85	5.679	5.683
2017/8/24	5.92	5.778	5.810
2017/9/18	5.76	5.848	5.750
2017/9/19	5.78	5.792	5.732
2017/9/20	5.79	5.804	5.755
2017/9/21	5.95	5.819	5.761
2017/9/22	5.98	5.940	5.913
2017/9/25	6.11	5.995	5.930
2017/9/26	6.07	6.067	6.039
2017/9/27	6.02	6.071	6.011
2017/9/28	6.05	6.048	5.976
2017/9/29	6	6.044	5.992
2017/10/9	6.1	6.008	5.956
2017/10/10	6.14	6.110	6.033
2017/10/11	6.2	6.159	6.054
2017/10/12	6.21	6.188	6.123

Table 5: The results of the falling model and the single model.

	Average error	Maximum error	Trend accuracy
Falling model	0.00575280455171	0.0189167045488	0.769230769231
Single model	0.0066222028267	0.0276437614071	0.461538461538

Table 6: The real data and predict results of the falling model and the single model.

Time	Real closing price	Falling model	Single model
		Prediction result	Prediction result
2016/7/15	3054.3	3015.018	3053.366
2016/7/18	3043.56	3037.105	3070.031
2016/7/19	3036.6	3027.025	3052.860
2016/7/20	3027.9	3024.982	3043.598
2016/7/21	3039.01	3031.914	3032.450
2016/7/22	3012.82	3036.633	3059.587
2016/7/25	3015.83	3022.038	3013.960
2016/7/26	3050.17	3017.658	3029.422
2016/7/27	2992	3048.599	3074.710
2016/7/28	2994.32	3002.108	3006.372
2016/7/29	2979.34	2967.689	2988.813
2016/8/1	2953.39	2982.602	2964.578
2016/8/18	3104.11	3081.673	3127.543
2016/8/19	3108.1	3095.075	3124.608
2016/8/22	3084.81	3080.501	3125.914
2016/8/23	3089.71	3088.842	3089.085
2016/8/24	3085.88	3082.001	3100.461
2016/8/25	3068.33	3092.965	3093.066
2016/10/25	3131.94	3077.205	3153.576
2016/10/26	3116.31	3085.473	3148.972
2016/10/27	3112.35	3109.123	3122.424
2016/10/28	3104.27	3099.440	3114.055
2016/10/31	3100.49	3103.166	3103.392
2016/11/1	3122.44	3096.934	3106.386
2016/11/2	3102.73	3120.894	3148.097
2016/11/3	3128.94	3113.404	3112.531

From Table 6, we can know that falling model not only can keep error in a low level but also can improve the accuracy of trend prediction. According to the experimental results of 7.1 and 7.2, bimodel method, including rising model and falling model, reaches nearly 75% accuracy and keeps the error in a low level.

7.4. Trend Forecast for Falling Model in a Long Period of Individual Share. This paper selected the stock of industrial and commercial bank of China (ICBC) from 31 March 2008 to 31 July 2015 (1768 sets of data) as an experimental training set and selected the stock of ICBC from three long periods which match the falling model from 11 August 2015 to 03 November 2017 as test sets. The test sets have 31 groups.

With the same GA-BP algorithm, the result of the falling model method and that of the single model method are shown in Table 7.

In Figures 10 and 11, green lines represent closing prices, blue points represent prediction results, and red points represent real closing prices. For each day, if the red point and the blue point are on the same side, it means that this trend prediction is accurate.

Form Tables 6 and 8, we can easily know that falling model can improve trend prediction accuracy not only in stock index but also in individual share. From the above

Table 7: Results of the falling model and the single model.

	Average error	Maximum error	Trend accuracy
Falling model	0.0123645507535	0.0893076243902	0.870967741935
Single model	0.0151008394772	0.0921751707317	0.451612903226

Table 8: Real data and prediction results of falling model and single model.

Time	Real closing price	Falling model	Single model
		Prediction result	Prediction result
2015/8/11	4.87	4.847	4.936
2015/8/12	4.8	4.773	4.869
2015/8/13	4.79	4.755	4.810
2015/8/14	4.77	4.768	4.798
2015/8/17	4.71	4.746	4.774
2015/8/18	4.63	4.693	4.706
2015/8/19	4.57	4.608	4.604
2015/8/20	4.55	4.523	4.524
2015/8/21	4.54	4.488	4.508
2015/8/24	4.1	4.466	4.478
2015/8/25	3.89	4.036	4.098
2015/9/7	4.52	4.609	4.835
2015/9/8	4.5	4.343	4.533
2015/9/9	4.5	4.462	4.499
2015/9/10	4.47	4.428	4.477
2015/9/11	4.43	4.419	4.452
2015/9/14	4.52	4.408	4.425
2015/9/15	4.54	4.458	4.491
2015/9/16	4.53	4.464	4.521
2015/9/17	4.52	4.495	4.536
2015/9/18	4.45	4.498	4.528
2015/9/21	4.42	4.436	4.458
2015/10/22	4.65	4.650	4.721
2015/10/23	4.61	4.581	4.685
2015/10/26	4.62	4.640	4.640
2015/10/27	4.63	4.615	4.634
2015/10/28	4.61	4.606	4.641
2015/10/29	4.58	4.590	4.624
2015/10/30	4.57	4.572	4.594
2015/11/2	4.52	4.559	4.576
2015/11/3	4.45	4.509	4.523

experiments, these results prove the certain practical significance of the bimodel method.

8. Conclusion

Stock prediction is an interesting and difficult question. Existing algorithms, including BP and many current algorithms, still could not provide helpful prediction results for stock investors. An important reason may be that stock data in a long period are too complex and include too many modes. The data-divider, which divides a complex data set into two simple sets, enables two "old" BP models to get satisfactory prediction results for the difficult question. It is interesting that "old" BP neural networks are still potential. An intelligent and self-adaptive data-divider is our future object.

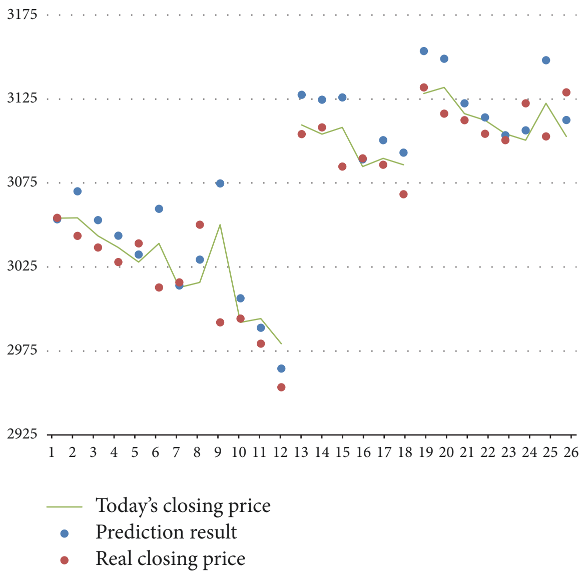

FIGURE 9: Single model.

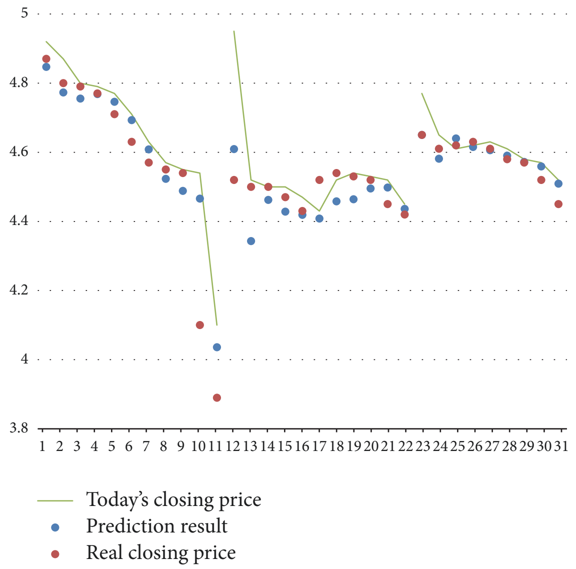

FIGURE 10: Falling model.

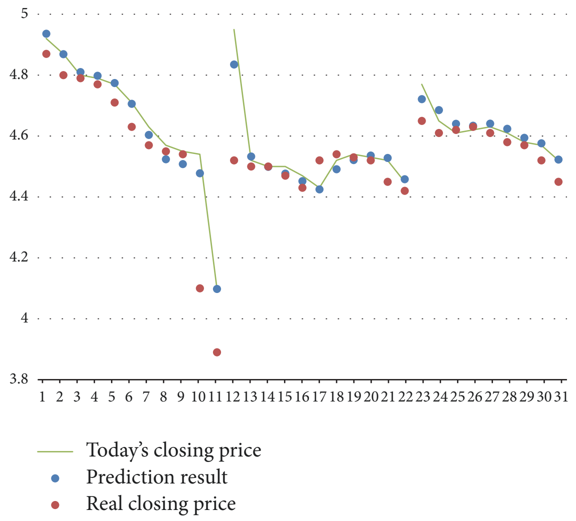

FIGURE 11: Single model.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors would like to thank the Chinese National Natural Science Foundation (61370177), Guangdong industry-university-research (2012B090600017 and 2011B090400204), and Student Research Project of South China University of Technology (2016).

References

[1] B. Luo, Y. Chen, and W. Jiang, "Stock market forecasting algorithm based on improved neural network," in *Proceedings of the 8th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA '16)*, pp. 628–631, China, March 2016.

[2] M. Qiu, Y. Song, and F. Akagi, "Application of artificial neural network for the prediction of stock market returns: the case of the Japanese stock market," *Chaos, Solitons & Fractals*, vol. 85, pp. 1–7, 2016.

[3] J. Wang and J. Wang, "Forecasting stock market indexes using principle component analysis and stochastic time effective neural networks," *Neurocomputing*, vol. 156, pp. 68–78, 2015.

[4] B. Al-Hnaity and M. Abbod, "A novel hybrid ensemble model to predict FTSE100 index by combining neural network and EEMD," in *Proceedings of the European Control Conference (ECC '15)*, pp. 3021–3028, July 2015.

[5] F. Wang, Y. Zhang, H. Xiao, L. Kuang, and Y. Lai, "Enhancing Stock Price Prediction with a Hybrid Approach Based Extreme Learning Machine," in *Proceedings of the 15th IEEE International Conference on Data Mining Workshop (ICDMW '15)*, pp. 1568–1575, November 2015.

[6] Y. Sun and Y. Gao, "An improved hybrid algorithm based on PSO and BP for stock price forecasting," *Open Cybernetics and Systemics Journal*, vol. 9, no. 1, pp. 2565–2568, 2015.

[7] V. W. Chu, F. Chen, R. K. Wong, I. Ho, and J. Lee, "Enhancing portfolio return based on market-sentiment linked topics," in *Proceedings of the International Conference on Big Data and Smart Computing (BigComp '16)*, pp. 85–92, January 2016.

[8] Y. Shynkevich, T. M. McGinnity, S. Coleman, and A. Belatreche, "Stock price prediction based on stock-specific and sub-industry-specific news articles," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '15)*, July 2015.

[9] M. Wu and X. Diao, "Technical analysis of three stock oscillators testing MACD, RSI and KDJ rules in SH & SZ stock markets," in *Proceedings of the 4th International Conference on Computer Science and Network Technology (ICCSNT '15)*, pp. 320–323, December 2015.

[10] H. Li, J. Bo, N. Tao, and Y. Bo, "A BP Neural Network Predictor Model for Stock Price," in *in proceeding of the Intelligent Computing Methodologies - 10th International Conference (ICIC '14)*, pp. 362–368, August 2014.

[11] W. Ma, Y. Wang, and N. Dong, "Study on stock price prediction based on BP neural network," in *Proceedings of the 2010 IEEE International Conference on Emergency Management and Management Sciences (ICEMMS '10)*, pp. 57–60, August 2010.

[12] P. Liu and Y. Ren, "Bp neural network model for prediction of listing Corporation stock price of Qinghai province," in *Proceedings of the International Conference on Logistics, Informatics and Service Science (LISS '15)*, July 2015.

[13] M.-T. Wu and Y. Yong, "The research on stock price forecast model based on data mining of BP neural networks," in *Proceedings of the 3rd IEEE International Conference on Intelligent System Design and Engineering Applications (ISDEA '13)*, pp. 1526–1529, January 2013.

[14] Q. Mingyue, *A study on prediction of stock market index and portfolio selection*, Fukuoka Institute of Technology, 2014.

[15] L. Zhang, T. Liu, and J. Zhang, "Analysis of Momentum Factor in Neural Network Blind Equalization Algorithm," in *proceeding of the 2009 WRI International Conference on Communications and Mobile Computing (CMC '09)*, vol. 1, pp. 345–348, 2009.

[16] H. Kuang, J. Jin, and Y. Su, "Improving Crossover and Mutation for Adaptive Genetic Algorithm," *Computer Engineering and Applications*, pp. 93–96, 2006.

[17] A. Brindle, *Genetic algorithms for function optimization*, Computer Science Dept., University Of Alberta, 1981.

[18] T. Yalcinoz and H. Altun, "Environmentally constrained economic dispatch via a genetic algorithm with arithmetic crossover," *IEEE AFRICON Conference*, vol. 2, article no. 79, pp. 923–928, 2002.