ARTICLE IN PRESS
Physica A xxx (xxxx) xxx

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

Stock price forecasting based on LLE-BP neural network model

Zhuoxi Yu [a,*], Lu Qin [b], Yunjing Chen [b], Milan Deepak Parmar [b]

[a] *Department of Statistics, School of Economics, Liaoning University, Shenyang, China*
[b] *School of Management Science and Information Engineering, Jilin University of Finance and Economics, Changchun, China*

ARTICLE INFO

Article history:
Received 27 March 2019
Received in revised form 3 January 2020
Available online xxxx

Keywords:
Stock price
LLE-BP model
PCA-BP model
ARIMA model

ABSTRACT

Most of the factors affecting stock prices have data redundancy and nonlinear characteristics. Classical linear mapping dimensional reduction methods such as principal component analysis (PCA) and linear discriminant analysis (LDA) cannot get good results for nonlinear problems. In this paper, a local linear embedding dimensional reduction algorithm (LLE) is selected to reduce the dimension of the factors affecting the stock price. The data after dimensional reduction is used as the new input variable of Back Propagation (BP) neural network to realize the stock price prediction. The prediction results are compared with the BP neural network model, PCA-BP model, and the traditional ARIMA (3,1,1) model. The results show that LLE-BP neural network model has higher prediction accuracy in stock price prediction, and it is an effective and feasible stock price prediction method.

1. Introduction

High-yield and high-risk stocks have been the focus of many investors since the establishment of the securities market. The changes in the stock market are inseparable from the market economy dynamics of the entire country and have an important impact on the promotion of national economic growth. Stock price volatility is a matter of great concern. Stock prices are difficult to describe, analyze and predict. With the continuous development of the stock market, there have been many methods of stock forecasting and linear dynamic system analysis, such as the traditional time series method. The statistical characteristics of time series are used to reflect the characteristics and changes of linear dynamic systems, thus revealing the inherent changes in stock prices. Linear dynamic system analysis methods mainly include differential autoregressive moving average model, multiple linear regression model, exponential smoothing model, etc. [1–3]; and nonlinear system analysis methods, such as nonlinear systems using neural networks to predict stock prices [4–7].

The stock market is a nonlinear system and is affected by many factors. Stock data has characteristics of high dimensionality, non-linearity and non-stationary [8]. Classic linear mapping dimensional reduction methods such as principal component analysis (PCA), discriminant analysis (LDA) and so on. These methods cannot get good results for nonlinear problems and are not suitable for dimensional reduction, which affects the multidimensional indicators of stock prices. The nonlinear dimensional reduction method (Locally Linear Embedding LLE) can break through the limitations of the principal component analysis method in nonlinear data, for example, can process and analyze nonlinear signals. LLE can also express the intrinsic manifold structure of data well, retaining the essential characteristics of the data. LLE algorithm is able to optimize parameters better because of its own fewer parameters. LLE algorithm is mainly used in face recognition [9], image processing [10], text classification and clustering [11], pattern recognition [12], data visualization [13] and other fields.

* Corresponding author.
E-mail address: yzx8170561@163.com (Z. Yu).

https://doi.org/10.1016/j.physa.2020.124197
0378-4371/© 2020 Published by Elsevier B.V.

This paper proposes to predict the stock price based on the LLE-BP neural network model. Firstly, the LLE algorithm is used to reduce the dimensionality of the high-dimensional financial indicators that affect the stock price. The index after the dimension reduction is used as the new input variable of the Back Propagation (BP) neural network. Then the stock price is predicted by BP neural network, and the prediction results are compared with the prediction results of the BP neural network model, PCA-BP model, and ARIMA model. Finally, the prediction accuracy of the LLE-BP neural network model proposed in this paper is verified.

2. Locally Linear Embedding Algorithm

The LLE algorithm is an unsupervised dimensionality reduction method for nonlinear data proposed by Roweis and Saul [14] in 2000. It can keep the local linear characteristics of samples when the dimensionality is reduced. In view of this, it is widely used in high-dimensional data visualization, image recognition, and other fields. The core idea is that for data in high-dimensional space, assuming that these data are linear in a small region, and a certain data can be expressed linearly by some data samples in its neighborhood, projecting the data to low-dimensional space, and keeping the weight coefficient of linear relationship before and after projection as same as possible, then reconstructing data points in low-dimensional space, and keeping reconstruction error to a minimum [15].

Suppose there are N sample points in the D-dimensional space of the high-dimensional space: $X = \{X_1, X_2, \ldots, X_N\}$, $X_i \in R^D, i = 1, 2, \ldots, N$, It is necessary to reduce the dimension so that there are N sample points that exist in the d-dimensional space. $Y = \{Y_1, Y_2, \ldots, Y_N\}$, $Y_j \in R^d, j = 1, 2, \ldots, N$, $d \ll D$, $Y = \{Y_1, Y_2, \ldots, Y_N\}$ is the mapped output of $X = \{X_1, X_2, \ldots, X_N\}$ in the d-dimensional space.

The steps of the LLE algorithm are as follows:

Step 1: Find the K nearest neighbors of each sample point X_i in high dimensional space and calculate the Euclidean distance:

$$d_{ij} = \left\| X_i - X_j \right\|, \tag{1}$$

The value of K can be calculated by multiple experiments or by using the *calc_k*() function of the *R* software if the value of d is known.

Step 2: A local reconstruction weight matrix of the sample point is calculated from his nearest neighbors. Define a function to measure the reconstruction error:

$$\begin{gathered} \min \varepsilon(w) = \sum_{i=1}^{N} \left\| X_i - \sum_{j=1}^{K} w_{ij} X_{ij} \right\|^2, \\ s.t. \sum_{j=1}^{K} w_{ij} = 1. \end{gathered} \tag{2}$$

N is the number of sample points, X_{ij} is the ith sample point using the jth sample to express, and w_{ij} is the coefficient (i.e., the weight) when the ith sample point expressed by the jth neighbor. When a data point does not belong to the nearest neighbors of the reconstructed data point (i.e., it is not the closest sample point to the reconstructed data point) $w_{ij} = 0$. $\sum_{j=1}^{K} w_{ij} X_{ij}$ is the result of reconstructing the ith sample point, and its difference from the ith sample point is the reconstruction error, and all the coefficients w_{ij} eventually form the local reconstruction weight $N \times N$ matrix W.

Step 3: Map all sample points X_i from the high dimensional original space to Y_i vectors of the low dimensional space, so the X_{ij} map to Y_{ij}. Define the cost function:

$$\begin{gathered} \min \varepsilon(Y) = \sum_{i=1}^{N} \left\| Y_i - \sum_{j=1}^{K} w_{ij} Y_{ij} \right\|^2 = \min tr(YMY^T), \\ s.t. YY^T = N * I. \end{gathered} \tag{3}$$

$M = (I - W)(I - W)^T$, I is the unit matrix. Finding the value Y that minimizes the cost function of Eq. (3) yields the low-dimensional output vector we need.

Finding the value to minimize Eq. (3) is equivalent to finding the smallest d eigenvalues of the M matrix. The first eigenvalue should be rounded off because it is zero. Finally, the feature vectors corresponding to the 2nd to d+1th eigenvalues of M are the output results after dimensional reduction.

3. BP (Back Propagation) neural network

BP neural network is a concept proposed by scientists represented by Rumelhart and McClelland [16] in 1986. It is a multilayer feedforward neural network based on an error backpropagation algorithm. The learning process of the algorithm is divided into two parts: signal forward propagation and error backpropagation. In the forward propagation

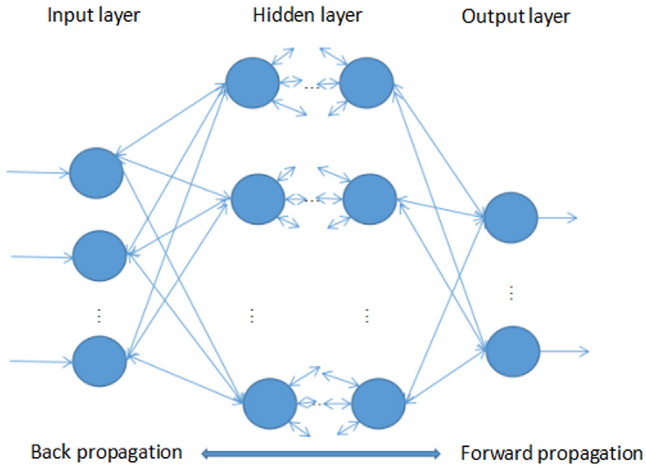

Fig. 1. Topological structure of BP neural network.

process, the input signal passes through the hidden layer to the output layer through nonlinear transformation. If the actual output of the output layer does not conform to the expected output, it will enter the error backpropagation part, the output will be back transmitted to the input layer through the hidden layer in some form, and the error will be allocated to all the units of each layer, so as to obtain the error signal of each layer unit, which is the basis for correcting the weight of each unit. The weights and thresholds of the network are adjusted continuously through backpropagation until the sum of the squares of the errors of the network is minimized [17].

The BP neural network is a kind of neural network with three or more layers, including the input layer, hidden layer, and output layer. Each layer is composed of one or more neurons, the neurons of each layer can only accept the input of the neurons of the previous layer, and the input information must be processed by the neurons of each layer to become the output of the output layer, that is, the input layer receives the external input information, the neurons of the hidden layer can only accept the input of the information of input layer, and the neurons of the output layer can only accept the information of the hidden layer. Its structure is shown in Fig. 1.

The BP neural network training process includes 7 steps in Fig. 2:

Step 1: Network initialization.

(1) Initialize the number of input layer n nodes, hidden layer l nodes, and output layer m nodes;

(2) Initialize weight: w_{ij}, w_{jk};

(3) Initialize the thresholds of the hidden layer and output layer a, b;

(4) The learning rate η and the excitation function of neurons f are given.

Step 2: Calculate the output of the hidden layer. The hidden layer output H is calculated according to the input variable X, the input layer and the implicit layer connection weight w_{ij} and the hidden layer threshold a.

$$H_j = f(\sum_{i=1}^{n} w_{ij}x_i - a_j), j = 1, 2, \dots, l \tag{4}$$

l is the number of hidden layer nodes, and f is the hidden layer excitation function. This function has a variety of expressions, such as logsig function, tansig function, purelin function.

Step 3: Calculate the output of the output layer. The BP neural network predictive output O is calculated based on the hidden layer output H, the connection weight w_{jk}, and the threshold b.

$$O_k = \sum_{j=1}^{l} H_j w_{jk} - b_k, k = 1, 2, \dots, m \tag{5}$$

Step 4: Calculate the error. The network prediction error e is calculated based on the network prediction output O and the expected output Y.

$$e_k = Y_k - O_k, k = 1, 2, \dots, m \tag{6}$$

Step 5: Update the weights. The network connection weight w_{ij}, w_{jk} is updated according to the network prediction error e.

$$w_{ij} = w_{ij} + \eta H_j(1 - H_j)x(i) \sum_{k=1}^{m} w_{jk}e_k \quad i = 1, 2, \dots, n; j = 1, 2, \dots, l \tag{7}$$

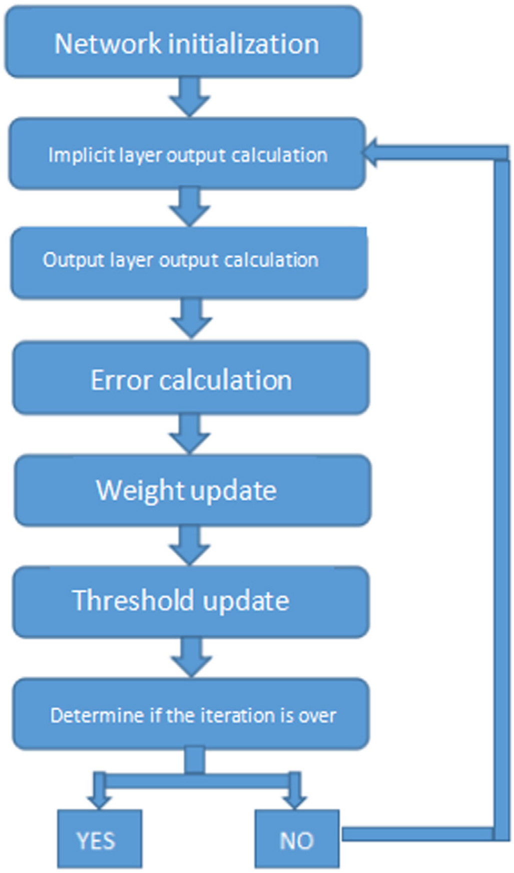

Fig. 2. BP neural network flow chart.

$$w_{jk} = w_{jk} + \eta H_j e_k \quad j = 1, 2, \ldots, l; k = 1, 2, \ldots, m \tag{8}$$

η is the learning rate.

Step 6: Update the thresholds. The network node thresholds a, b are updated according to the network prediction error e.

$$a_j = a_j + \eta H_j (1 - H_j) \sum_{k=1}^{m} w_{jk} e_k \quad j = 1, 2, \ldots, l \tag{9}$$

$$b_k = b_k + e_k \quad k = 1, 2, \ldots, m \tag{10}$$

Step 7: Determine whether the algorithm iteration ends. If not, return to step 2.

4. Empirical study on LLE-BP neural network model

4.1. Indicator selection

In this paper, the data of the stock Pingtan Development (c000592) in the agricultural, forestry, animal husbandry and fishery industries from January 1, 2016, to August 8, 2017, are selected as the experimental data. A total of 14 indicators were selected as the main factors affecting the stock price, including opening price, highest price, lowest price, closing price, trading volume, trading amount, daily amplitude, circulating shares, earnings per share (diluted) (yuan/share), return on net assets (diluted), accumulation fund per share (yuan/ share), operating profit per share (yuan/share), net assets per share (yuan/share), and operating income per share. At the same time, the closing price is selected as the stock price forecast indicator. Partial data are shown in Table 1.

Table 1
2016–2017 Pingtan Development (c000592) index data (partial data).

Date	Opening price	Highest price	Lowest price	Closing price	Trading volume	Trading amount	Daily amplitude	Circulating shares
2016.1.1	19.99	20	17.98	17.98	13303476	248382664.9	10.1101	965890446
2016.1.2	16.39	17.63	16.39	17.62	29791063	507733880.4	6.8966	965890446
2016.1.3	17.62	18.28	17.3	18.11	16028569	285966559.5	5.5619	965890446
2016.1.4	17.6	17.68	16.3	16.33	5218100	87986980.02	7.6201	965890446
2016.1.5	16.88	17.9	15.3	17.43	25465778	429309940.9	15.9216	965890446
2016.1.6	16.95	17.2	15.69	15.69	25727515	422714703.8	8.6632	965890446
.....								
2017.8.3	5.55	5.61	5.5	5.5	14145900	78321099	1.982	1931780892
2017.8.4	5.5	5.54	5.42	5.44	18426694	100627570.3	2.1818	1931780892
2017.8.5	5.49	5.78	5.44	5.67	52667521	296471159.2	6.25	1931780892
2017.8.6	5.61	5.74	5.57	5.61	34380650	193804309.8	2.9982	1931780892
2017.8.7	5.63	5.82	5.61	5.76	44440389	255746706.7	3.7433	1931780892
2017.8.8	5.77	5.93	5.74	5.75	31048848	180661320.3	3.2986	1931780892

Date	Earnings per share (diluted) (yuan/share)	Return on net assets (diluted)	Accumulation fund per share (yuan/share)	Operating profit per share (yuan/share)	Net assets per share (yuan/share)	Operating income per share
2016.1.1	0.04	2.6537	0.6	0.02	1.38	0.74
2016.1.2	0.04	2.6537	0.6	0.02	1.38	0.74
2016.1.3	0.04	2.6537	0.6	0.02	1.38	0.74
2016.1.4	0.04	2.6537	0.6	0.02	1.38	0.74
2016.1.5	0.04	2.6537	0.6	0.02	1.38	0.74
2016.1.6	0.04	2.6537	0.6	0.02	1.38	0.74
...	...	...	...	...	...	...
2017.8.3	−0.01	−0.4081	0.76	−0.01	1.63	0.06
2017.8.4	−0.01	−0.4081	0.76	−0.01	1.63	0.06
2017.8.5	−0.01	−0.4081	0.76	−0.01	1.63	0.06
2017.8.6	−0.01	−0.4081	0.76	−0.01	1.63	0.06
2017.8.7	−0.01	−0.4081	0.76	−0.01	1.63	0.06
2017.8.8	−0.01	−0.4081	0.76	−0.01	1.63	0.06

Table 2
Partial data after standardization.

Date	Opening price	Highest price	Lowest price	Closing price	Volume	Turnover	Daily amplitude
2016.1.1	3.3049	3.1534	2.87788	2.72721	−0.61392	0.05031	2.67847
2016.1.2	2.2504	2.48051	2.39338	2.62132	0.07731	1.28079	1.32253
2016.1.3	2.61069	2.66506	2.67067	2.76546	−0.49967	0.22862	0.75935
2016.1.4	2.60483	2.4947	2.36595	2.24184	−0.9529	−0.71068	1.62781
...	...	...	...	...	...	...	...
2017.8.5	−0.9424	−0.88395	−0.94329	−0.89394	1.0364	0.27846	1.0497
2017.8.6	−0.90725	−0.8953	−0.90367	−0.91159	0.26973	−0.20864	−0.3224
2017.8.7	−0.9014	−0.87259	−0.89149	−0.86747	0.69148	0.08525	−0.008
2017.8.8	−0.86039	−0.84136	−0.85187	−0.87041	0.13005	−0.27099	−0.19565

Date	Circulating shares	Earnings per share (diluted) (yuan/share)	Return on net assets (diluted)	Accumulation fund per share (yuan/share)	Operating profit per share (yuan/share)	Net assets per share (yuan/share)	Operating income per share
2016.1.1	−1.63185	1.82518	1.92103	−0.67883	1.80802	−0.8103	1.68503
2016.1.2	−1.63185	1.82518	1.92103	−0.67883	1.80802	−0.8103	1.68503
2016.1.3	−1.63185	1.82518	1.92103	−0.67883	1.80802	−0.8103	1.68503
2016.1.4	−1.63185	1.82518	1.92103	−0.67883	1.80802	−0.8103	1.68503
...	...	...	...	...	...	...	...
2017.8.5	0.61122	−1.50188	−1.31957	−0.36447	−1.13885	−0.45742	−1.18912
2017.8.6	0.61122	−1.50188	−1.31957	−0.36447	−1.13885	−0.45742	−1.18912
2017.8.7	0.61122	−1.50188	−1.31957	−0.36447	−1.13885	−0.45742	−1.18912
2017.8.8	0.61122	−1.50188	−1.31957	−0.36447	−1.13885	−0.45742	−1.18912

4.2. Data processing

The Z-score method is used to standardize the original 14-dimensional data in this paper, in order to eliminate the influence of the magnitude and numerical size of the dimension and the variable itself. The partial standardized data are shown in Table 2.

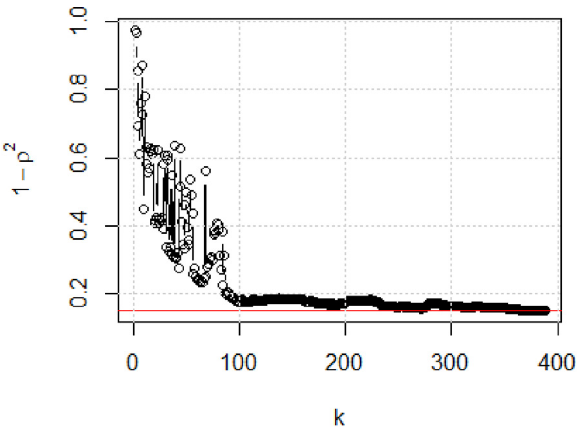

Fig. 3. Choice of the best K value.

Table 3
Partial data after dimensional reduction with LLE.

Date	D1	D2	D3	D4
2016.1.1	0.23415	−0.65290	0.52296	−2.51557
2016.1.2	−0.16952	0.35015	0.38490	−2.34420
2016.1.3	0.04458	−0.28189	0.88151	−2.24160
2016.1.4	0.27330	−0.62843	1.08461	−2.02614
...	...	...	...	...
2017.8.5	2.98347	1.87509	−0.87484	0.29528
2017.8.6	3.11215	1.48199	−0.25774	0.51013
2017.8.7	3.02092	1.72003	−0.50790	0.40627
2017.8.8	3.11519	1.35362	−0.25456	0.50462

4.3. Dimensional reduction with LLE algorithm

The LLE algorithm reduces the dimension by mainly determining two parameters, the number K of neighbors of each sample, the dimension d of the final output and $K > d$. By applying the essential feature dimension estimation function in the MATLAB toolbox, and applying the Maximum Likelihood Estimator (MLE) method, the dimension $d = 4$ of the data is estimated. The distance between the sample points is calculated using the Euclidean distance.

According to the obtained d value, the optimal neighbor number K is found for the data in this paper using the *calc_k*() function in *R*. The result of the operation is shown in Fig. 3.

As can be seen from Fig. 3, the optimum K value is 100.

According to the obtained optimal parameters $K = 100$, $d = 4$, the data is recalculated by using the *lle* packet in *R* software, and the dimension reduced data are obtained as new input of the BP neural network. The partial data after dimensional reduction are shown in Table 3.

4.4. Prediction with BP neural network

The data from January 1, 2016 to July 28, 2017 are selected as training samples to predict the closing price for 11 days from July 29, 2017 to August 8, 2017. According to the dimensional reduction result of the LLE algorithm, the four-dimensional samples are selected as the input of the BP neural network, and the stock closing price of the next day is the output of the neural network. In this paper, a three-layer neural network model with only one hidden layer is adopted. Since the input sample dimension is 4 and the output sample dimension is 1, the structure of the BP neural network is 4-8-1. The input layer has 4 nodes, the hidden layer has 8 nodes, and the output layer has 1 node. The logsig function acts as a hidden layer node transfer function. The BP neural network prediction accuracy is greatly affected by the choice of the functions of the hidden layer and the output layer because the network structure is the same for the weights and thresholds. The prediction errors corresponding to different transfer functions are shown in Table 4. The nonlinear sigmoid function is used as a neuron activation function.

4.5. Comparative analysis with other predictive models

In order to verify the prediction accuracy of the proposed model, the BP neural network model, the PCA-BP neural network model based on linear dimensional reduction [18] and the traditional time series model are compared with the LLE-BP neural network prediction model proposed in this paper.

Table 4
Prediction errors corresponding to different transfer functions.

Implicit layer function	Output layer function	Error percentage	MSE
logsig	tansig	40.63%	0.9025
logsig	purelin	0.08%	0.0001
logsig	logsig	352.65%	181.2511
tansig	tansig	31.90%	1.1733
tansig	logsig	340.90%	162.9698
tansig	purelin	1.70%	0.0107
purelin	logsig	343.36%	143.76334
purelin	tansig	120.08%	113.0281
purelin	purelin	196.49%	99.0121

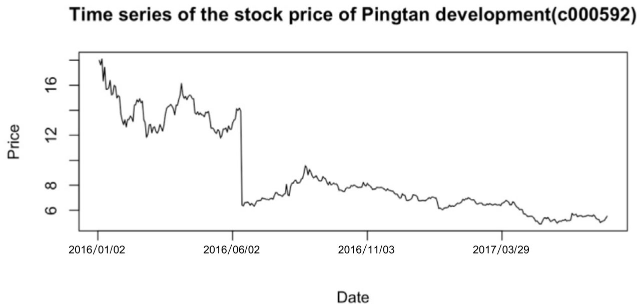

Fig. 4. Time sequence diagram for the stock price of Pingtan Development (c000592).

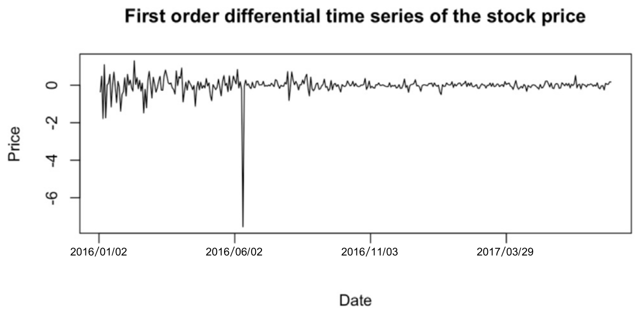

Fig. 5. Time sequence diagram for the first order differential stock price.

4.5.1. ARIMA model construction

ARIMA (p, d, q) model aims at stationary sequence modeling. The time sequence diagram of the original data is shown in Fig. 4. From the time sequence diagram, it can be seen that the overall trend is declining, so it can be judged that the sequence is non-stationary; it needs to be differentiated and smoothed, and the sequence after the first-order difference is shown in Fig. 5. The unit root test is carried out for the sequence after the difference, and the test result shows that the sequence is stationary; therefore, the modeling can be carried out according to the sequences ACF (Fig. 6) and PACF (Fig. 7) are tailed. According to the principle of AIC minimum, the final comprehensive model is ARIMA (3,1,1) (AIC = 561.48), and the goodness of fit of the model is relatively high $R^2 = 0.980$.

4.5.2. PCA-BP model construction

By analyzing all the data of these 14 variables, it can be seen that there are significant correlations between the variables from the KMO and Bartlett test, indicating that there is data redundancy, so it is necessary to carry out principal component analysis on these data. After selecting the index whose eigenvalue is greater than 1 for principal component selection, we can see that the cumulative variance contribution rate of the first three principal components has exceeded 85%, so we need to extract three principal components. The partial data of the three main components after dimensional reduction are shown in Table 5. The data of three principal components are used as input data of the BP neural network.

4.5.3. Prediction effect comparison

The forecast results of the closing price of the BP neural network model, the PCA-BP neural network model the traditional time series model ARIMA (3,1,1) and the LLE-BP neural network prediction model proposed in this paper are presented in Table 6.

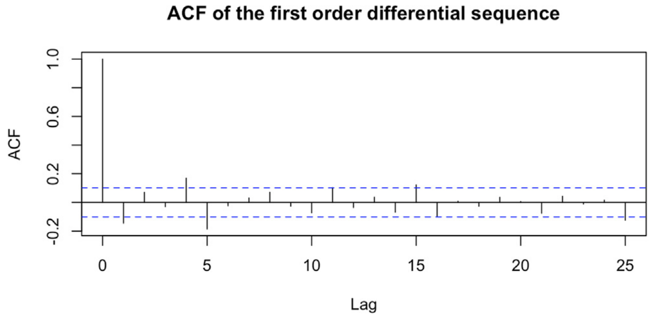

Fig. 6. ACF of the first order differential sequence.

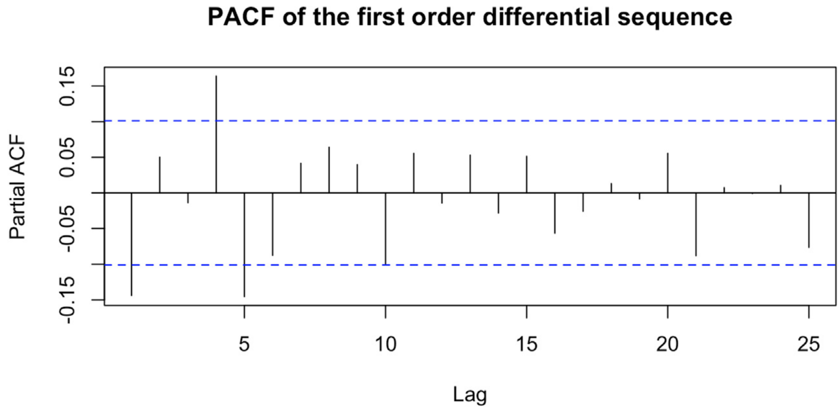

Fig. 7. PACF of the first order differential sequence.

Table 5
Partial data of the three main components.

Date	Component 1	Component 2	Component 3
2016.1.1	2.54803	0.59039	0.51841
2016.1.2	2.12160	0.74544	0.90003
2016.1.3	2.31788	0.61005	0.08362
2016.1.4	0.22547	0.71018	−0.22360
...	...	...	...
2017.8.5	−1.15527	−0.06952	0.85560
2017.8.6	−1.10124	−0.16849	−0.04612
2017.8.7	−1.11947	−0.14424	0.33320
2017.8.8	−1.06068	−0.18275	−0.09140

Table 6
Forecast results of the closing price.

Time	Actual value	BP	LLE-BP	PCA-BP	ARIMA
2017.7.29	5.46	4.4747	5.3254	5.3466	5.57
2017.7.30	5.53	5.4380	5.5591	6.5914	6.32
2017.7.31	5.55	4.5793	5.3785	5.5491	5.03
2017.8.1	5.57	4.5840	5.4293	5.7416	5.16
2017.8.2	5.55	4.5655	5.3662	5.4370	5.80
2017.8.3	5.5	4.3850	5.3822	5.5543	5.50
2017.8.4	5.44	4.4138	5.3688	5.2243	4.43
2017.8.5	5.67	5.8324	5.4302	7.2347	6.03
2017.8.6	5.61	5.0528	5.4801	6.5663	5.33
2017.8.7	5.76	6.0592	5.6167	6.8709	5.23
2017.8.8	5.75	5.1846	5.4806	5.6712	5.83

The prediction results were evaluated using mean absolute error (MAE) and root mean square error (RMSE). The results are shown in Table 7.

Table 7
Error analysis results.

Closing price	BP	LLE-BP	PCA-BP	ARIMA
RMSE	0.7923	0.1619	0.7688	0.4911
MAE	0.7040	0.1483	0.5121	0.3945

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{x}(i) - x(i))^2}, \qquad MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{x}(i) - x(i)|.$$

$\hat{x}(i)$ is the predictive value, and $x(i)$ is the actual value.

From Table 7, the best predictive performance is of the LLE-BP model, followed by the ARIMA model, the PCA-BP model, and the BP model, whether from RMSE or MAE. The prediction performance of the BP model is only slightly weaker than the PCA-BP model, indicating that for the data in this paper, the PCA dimensional reduction effect is not obvious for nonlinear data. The traditional ARIMA (3,1,1) model predicts better than the BP model PCA-BP model because the ARIMA model has a better predictive effect on short-term prediction.

5. Conclusion

For the characteristics of high-dimensional non-linearity of stock data, the non-linear dimensional reduction method LLE algorithm is used to reduce the dimension of high-dimensional stock price influencing factor data in this paper. Combined with the BP neural network in the widely used multilayer feedforward neural network, the closing price of a stock is predicted, and the prediction results are compared with the single BP neural network model, the BP model after PCA linear dimensional reduction and the traditional model ARIMA (3,1,1). The results are compared and analyzed, the simulation results show that the LLE-BP neural network model proposed in this paper has better prediction accuracy, which has certain theoretical guidance significance and practical application value for investors in the stock market to make investment decisions, and provides a certain reference for stock price prediction. Since there is no theoretical guidance for the determination of the number of hidden layers, neuron data and related parameters in the BP neural network, it can only be determined by experience or experiment, which increases the difficulty of learning and time consumption [19].

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The research was supported by the National Social Science Foundation of China (No. 16BTJ020).

References

[1] H.Z. Liu, Y.P. Bai, Analysis of time series model and neural network model in stock forecasting, Math. Pract. Theory 41 (04) (2011) 14–19.
[2] X.B. Huang, C.F. Wang, Z.M. Fang, C.L. Xiong, Detection of Chinese stock information based on hidden Markov model, Syst. Eng.-Theory Pract. 32 (04) (2012) 713–720.
[3] J.J. Shi, T. Song, Analysis of long-term fluctuation trends and influencing factors in stock market—based on spline-GARCH model, J. Appl. Stat. Manag. 34 (01) (2015) 175–182.
[4] A. Kazem, E. Sharifi, F.K. Hussain, et al., Support vector regression with chaos-based firefly algorithm for stock market price forecasting, Appl. Soft Comput. J. 13 (02) (2013) 947–958.
[5] Y. Zuo, E. Kita, Stock price forecast using Bayesian network, Expert Syst. Appl. 39 (08) (2012) 6729–6737.
[6] Mauna, Liu Q.J., Application of neural network method in stock investment, Stat. Inf. Forum 23 (01) (2008) 63–67.
[7] D.R. Xiao, Z.T. Yang, Neural network stock forecasting model based on particle swarm training, Statist. Decisions (12) (2009) 20–22.
[8] H. Cai, R.Y. Chen, Research on stock price prediction based on PCA-BP neural network, Comput. Simul. 28 (03) (2011) 365–368.
[9] J.Q. Wu, X.L. Zhou, K. He, F.N. Lang, Face recognition based on LLE and BP neural network, Laser J. (05) (2006) 71–73.
[10] L. Xie, Y.X. Wei, G.P. Jiang, et al., A small sample combination forecasting method based on local linear embedding, Statist. Decisions (09) (2015) 9–12.
[11] Y. Feng, H.Y. Wang, Q.C. Cheng, A.P. Liu, Chinese text clustering based on LLE-K means method, Comput. Digit. Eng. 38 (11) (2010) 10–12+21.
[12] L. Luo, Y.H. Li, LLE-based classification algorithm and its application in passive millimeter wave target recognition, J. Electron. Inf. Technol. 32 (06) (2010) 1306–1310.
[13] S. Lafon, Y. Keller, R.R. Coifman, Data fusion and multicue data matching by diffusion maps, IEEE Trans. Pattern Anal. Mach. Intell. 28 (11) (2006) 1784–1797.
[14] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.
[15] H.Y. Wang, J. Zheng, Z.G. Yao, L. Li, Application of LLE method based on clustering and improved distance in data dimensional reduction, J. Comput. Res. Dev. (08) (2006) 1485–1490.

[16] David E. Rumelhart, James L. McClelland, et al., Parallel distributed processing: Foundations v. 1: Explorations in the microstructure of cognition, Language 63 (04) (1986) 45–76.

[17] C.P. Su, Y.P. Wang, Information commodity price forecasting model based on BP neural network, Stat. Inf. Forum 22 (01) (2007) 93–95.

[18] X.P. Yang, Stock price forecasting analysis based on principal component and BP neural network, Statist. Decisions (12) (2004) 42–43.

[19] H.D. Long, G.L. Yan, Research on GDP time series prediction based on SARIMA, GM(1, 1) and BP neural network integration model, J. Appl. Stat. Manag. 32 (05) (2013) 814–822.

Please cite this article as: Z. Yu, L. Qin, Y. Chen et al., Stock price forecasting based on LLE-BP neural network model, Physica A (2020) 124197, https://doi.org/10.1016/j.physa.2020.124197.