Research on Stock Trend Analysis and Prediction Based on PCA-LSTM

Jian Zhang

Xi'an University of Science & Technology, Xi'an, Shaanxi, 710054, China

704301853@qq.com

Keywords: Stock price prediction; LSTM; Principal component analysis; Neural network

Abstract: Stock prediction has always been a hot and difficult topic for scholars at home and abroad, and stock research also has important practical and theoretical significance. Stock data is sequential. However, neural networks have performed well in dealing with time series problems. Among them, long short-term memory neural networks are very suitable for processing this time series data with long-term dependence, so this article is mainly based on principal component analysis and performs feature extraction on stock data. The stock trend is analyzed and predicted through the adjustment and optimization of the LSTM neural network model.

1. Introduction

As global information technology continues to innovate and productivity increases, people's material lives have become more enriched, and their lifestyles and concepts have also changed significantly. More and more people have more wealth than they need to live, and more and more of their excess funds are being invested in stocks, and stock investments can yield considerable returns to their investors. However, stocks are risky, and investment needs to be cautious. Because the stock market is not controllable, investors face different investment risks. People desire to be able to accurately predict stock prices, reduce investment risks, and increase their returns.

In the past, fundamental and technical analyses were generally used to evaluate stock prices, stock market fluctuations, etc. These traditional stock investment analysis methods have indeed promoted the development of the stock market, but they are also criticized for several reasons. First, the prediction accuracy is low, and they are not very useful for guiding short-term investors. Second, the scope of consideration is narrow, and it is impossible to judge the market's long-term trend effectively. Third, it is too influenced by psychological factors and cannot be used as a unified standard for guiding investors' decisions.

As information technology has progressed, artificial intelligence systems, such as neural networks, have gradually entered people's daily lives. Neural networks are mainly used to simulate the nervous systems of humans and other organisms. It has been found that neural networks are more efficient than traditional mathematical models, so the number of people studying neural networks has increased, and in the end, there have been relatively impressive results. Moreover, the study also found that neural networks are very suitable for non-stationary, random-like, and other nonlinear complex problems. Therefore, many scholars and experts have begun to use neural networks for stock prediction because stocks are data with characteristics such as nonlinearity and high noise. Compared with traditional stock analysis methods, using neural networks for stock prediction analysis has better predictions. As a result, the accuracy of stock price prediction has been significantly improved, so more and more people are using neural networks to predict stocks, among which fuzzy neural networks, BP neural networks, etc., are widely used. The BP neural network cannot capture the time series information of stock data. Several scholars have found that RNN neural networks are better at predicting the future. RNN neural networks can bring past information to the present, but they have a short-term memory. When the training time is too long, there will be a long-term dependence, eventually leading to the gradient's disappearance or explosion. LSTM neural networks were originally designed to solve this problem, but now LSTM is primarily used for text analysis, speech recognition, and several other applications. There have been relatively few studies on using LSTM in

  DOI: 10.25236/semihs.2024.006

stock price prediction. Therefore, this study will be based on LSTM to predict and study stock price trends.

2. Introduction of Model Structure and Principle

2.1 Principal Component Analysis

Principal Component Analysis (PCA) is a commonly used dimensionality reduction technique, which can be used for exploratory analysis and feature extraction of data. The basic idea is to convert the original high-dimensional data into a low-dimensional linear combination to retain the information of the original data as much as possible.

2.1.1 The Steps of PCA

Data standardization: First, the original data is standardized so that the mean value of each feature is 0 and the variance is 1. PCA is calculated based on the covariance matrix. Therefore, if the scale difference between features is large, it will hurt the results [1]. Calculate the covariance matrix: Calculate the covariance matrix of the standardized data, which reflects the correlation between different features. The elements of the covariance matrix represent the covariance between corresponding features.

Eigenvalue decomposition: Perform eigenvalue decomposition on the covariance matrix to obtain eigenvalues and corresponding eigenvectors. The eigenvalues represent the importance of the corresponding features, and the eigenvectors corresponding to the larger eigenvalues indicate the main direction of the data.

Selection of eigenvectors: Depending on the size of the eigenvalues, select the eigenvectors that correspond to the first k eigenvalues as the principal components, where k represents the dimension after dimensionality reduction.

Data conversion: The original data is projected onto the selected principal component to obtain the data after dimensionality reduction. The principal components are linear combinations of the original features, which can be achieved by multiplying matrices [2].

2.2 Long Short-Term Memory

LSTM (Long Short-Term Memory) is an RNN (Recurrent Neural Network) mainly used to process sequence data and has long memory capabilities [3] [4]. Based on the input data and the hidden state of the previous time step, the LSTM can update the state of the cells and pass that state on to the next time step, which enables the LSTM to maintain a long memory ability when dealing with long sequences to better capture long-term dependencies in sequences. The details are shown in Figure 1.

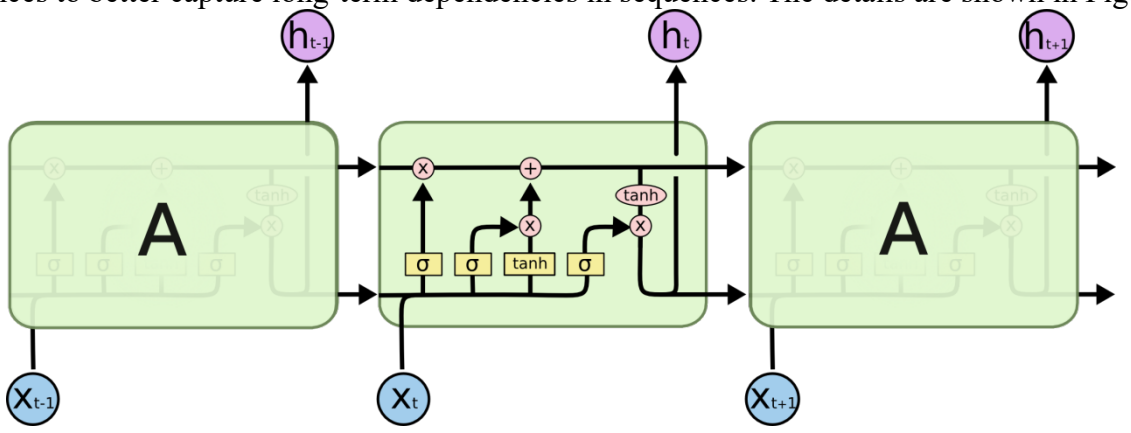
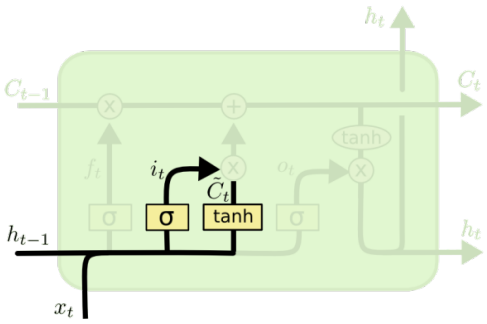


Figure 1 Process sequence data of LSTM

Where, h_{t-1} represents the implicit state of time $t-1$, C_{t-1} represents memory cell of time $t-1$, X_t represents the input of time t, σ represents the sigmoid function.

The working principle of LSTM can be briefly summarized in the following steps and shown in Figure 2:

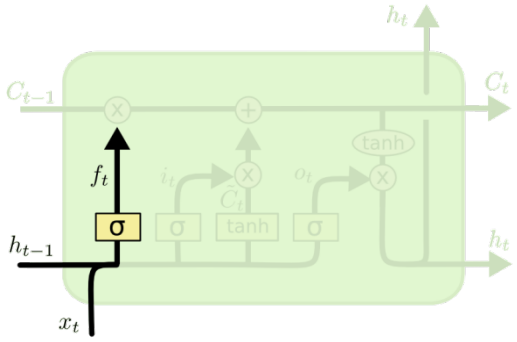


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Figure 2 The working principle of LSTM

Forget Gate: The input data of the current time step and the hidden state of the previous time step are received, and a Sigmoid function generates a value between 0 and 1. （0 means completely abandoning the previous cell state, and 1 means completely retaining the previous cell state.）This value determines how many cellular states of the previous time step are to be forgotten. Forget Gate: $F_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$. The details are shown in Figure 3.

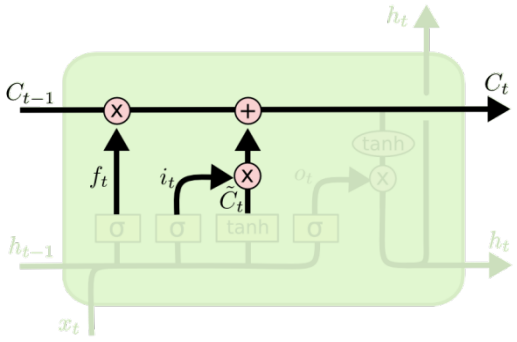


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Figure 3 The process of Forget Gate

Update Cell State: The cell state is updated by inputting the input data of the gate, the forgetting gate, and the current time step [5] [6]. It is determined by the input gate how much new information is added to the cell state and by the forget gate how much old information is removed from the cell state. The details are shown in Figure 4.

Update Cell State: $C_t = f_t * C_{t-1} + I_t * \tilde{C}_t$



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Figure 4 Output gate process

Output Gate: Through a Sigmoid function, receive the input data of the current time step and the hidden state of the previous time step and determine how many cell states will be output. Next, we process the cell state through tanh (getting a value between -1 and 1) and multiply it by the output of the sigmoid gate [7] [8]. Ultimately, we only output the part we are sure to output. Output Gate: $O_t = \sigma(W_O \cdot [h_{t-1}, x_t] + b_O)$. Hidden State: $h_t = O_t * \tanh(C_t)$, the hidden state is calculated by the output gate and cell state. The hidden state can be regarded as the output of the current time step.

3. Feature Extraction and Parameter Tuning Optimization

3.1 Feature Extraction

In the past, trading indicators were mostly used to predict stock prices. However, these indicators do not adequately reflect changes in stock prices. Therefore, in this study, comprehensively considering transaction, technical, and environmental indicators, the research object is cut from multi-dimensional and multi-feature. The specific selected characteristic indexes are shown in Table 1 below:

Table 1 The specific selected characteristic indexes

Technology index	Meaning of technical indicators	Trading index	Meaning of trading index
PE	Pricing/Earning ratio	open	Opening price
EPS	Earnings per share	close	Closing price
MNIF	Net inflow of main funds	high	Maxivalence
MNIFR	Net inflow rate of main funds	low	Bottom price
Environmental index	Environmental index	volume	Turnover
SCFI	Shanghai Composite Financial Index	pUpDownRate	Price limit
SCI	Shanghai Composite Index	*Turnover Rate*	Buying-selling rate

There are a total of 13 feature indicators in this article. The large number of variables means a large amount of calculation [9] [10] [11]. There may also be correlations between variables, and some input variables may have very little to do with the closing price, which will affect the experiment's complexity and prediction accuracy.

In the case of blindly reducing variables, a great deal of useful information will be lost, and the accuracy of calculations will be reduced. Model performance can be improved by selecting appropriate features and performing effective feature engineering. It is possible to improve the model's predictive accuracy by combining different features, transforming them, and selecting different selection methods. This article selects the feature extraction method of principal component analysis 1 to identify patterns by analyzing data and finding a pattern that can reduce data dimensions and information loss.

3.2 Parameter Tuning Optimization

The advantages of using an LSTM model to predict stock closing prices include: (1) Long-term dependencies in time series data can be captured to understand market trends better. (2) The forgetting gate mechanism of LSTM can effectively avoid overfitting problems and improve the model's generalization ability. (3) The LSTM can handle non-stationary time series data with good robustness. (4) LSTM can embed contextual information in the input data, making the model more flexible when processing sequences. Therefore, this article builds a stock price prediction model based on the LSTM neural network.

A parameter adjustment is an essential step in optimizing the performance of an LSTM model and improving prediction accuracy. The LSTM model constructed in this article involves parameter adjustment actions in the following areas:

Network structure adjustment: The number of layers and neurons of the LSTM model can be tried to be adjusted.

Increasing the number of layers and neurons of the model can increase its capacity and improve its fitting ability. However, excessively increasing the model capacity may lead to overfitting, so proper tuning and validation are required.

Adaptation of learning rate: The learning rate is an important hyperparameter that controls the

speed at which model parameters are updated [12]. Different learning rate values can be tried to observe the convergence speed and performance of the model during training. In the case of a significant learning rate, the model may fail to converge, whereas in the case of a small learning rate, the training speed of the model may be slow.

Optimizer selection: The optimizer of the LSTM model can choose Adam, RMSprop SGD, etc. Different optimizers have different performance characteristics and convergence speeds. You can try different optimizers to observe the training effect of the model.

Dropout: By randomly discarding the output of some neurons during training, the risk of overfitting the model can be reduced. The different dropout ratios can be tried to find the right value.

Time step: Modifying the time step may impact the model's performance. A smaller time step can capture the patterns and trends in the sequence more finely, but it also increases the computational complexity [13] [14] [15]. The long sequence length may lead to gradient disappearance or gradient explosion problems, which can be adjusted according to the problem's characteristics and the data's length.

4. The Experiment and Results

4.1 Usage of Platforms and Data

This article mainly uses the LSTM neural network to build a stock price prediction model, which selects China Ping An's stock transaction data from October 8, 2013 to September 28, 2023, a total of 10 years. The data comes from Oriental Fortune, and the programming environment is Python3. 7, mainly using Torch, Sklearn, and others to conduct experiments.

4.2 Record of Experimental Results

4.2.1 Reduction Comparison of Principal Component Analysis

The original data is reduced in dimension through principal component analysis, shown in Table 2, Figures 5 and 6.

Table 2 The reduced original data in dimension through principal component analysis

1	2	3	4	5	6	7	8	9	10	11	12
35	61. 6	78. 0	86. 0	90. 4	94. 1	95. 6	96. 6	97. 5	98. 3	98. 9	99. 3

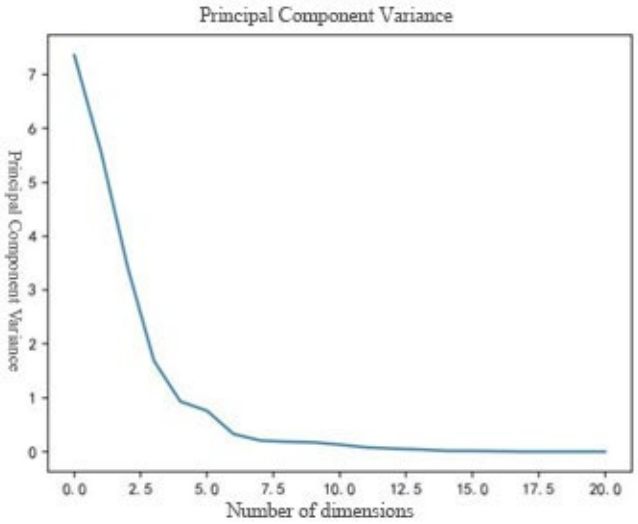


Figure 5 Principal component variance

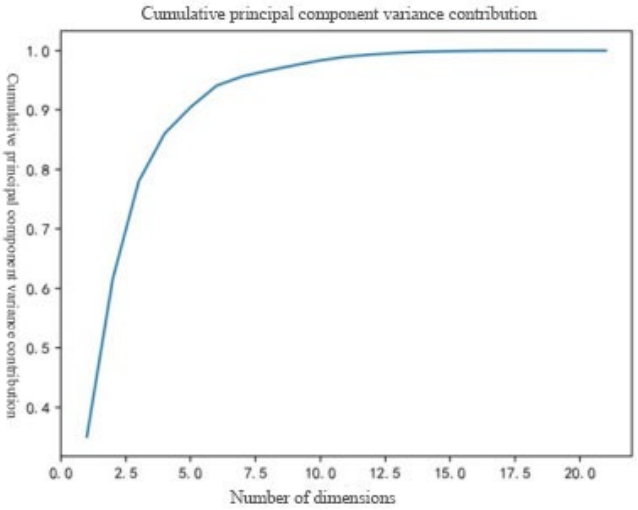


Figure 6 Cumulative principal component variance contribution

4.2.2 Model Operation Comparison

Based on the principal component analysis, different feature dimensions are extracted, the LSTM model is run and compared, and the number of feature dimensions is selected to be 5, as shown in Table 3 and Figure 7:

Table 3 The data of different feature dimensions

Dimensions No.	**5**	**7**	**9**	**11**
Train score RMSE:	0.23	0.69	0.90	0.91
Test score: RMSE	0.17	0.57	0.77	0.75

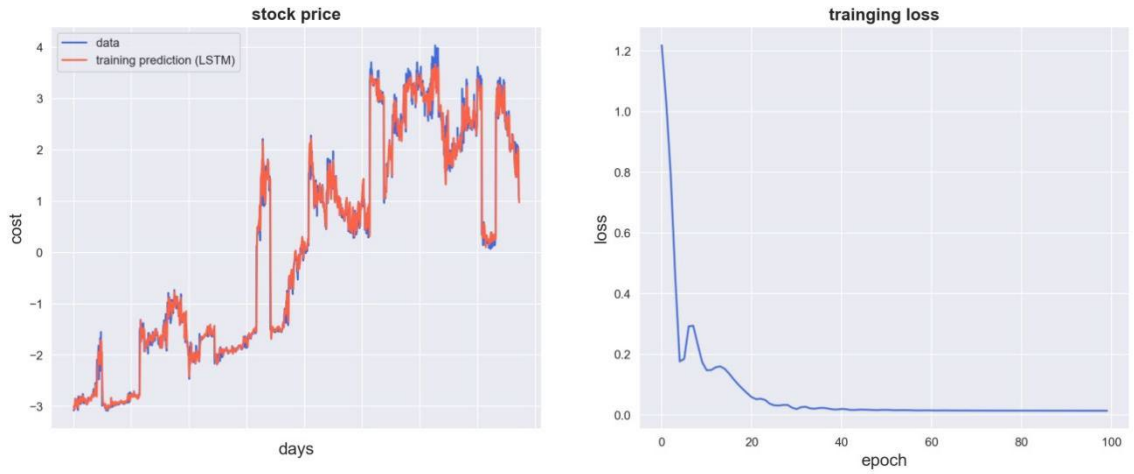


Figure 7 The stock price data and training loss data

If the number of feature dimensions is 5, the number of different hidden layers is set, and the LSTM model is run and compared [16] [17]. Select the number of hidden layers as 2, and the result is shown in Table 4:

Table 4 Data corresponding to different hidden layers No.

Number of hidden layers	1	2	3	4
Train score RMSE:	0.23	0.18	0.25	0.29
Test score: RMSE	0.21	0.17	0.24	0.32

If the number of feature dimensions is 5, the number of neurons in different hidden layers is set, and the LSTM model is compared. The number of hidden layer neurons is selected as 32, and the result is shown in Table 5:

Table 5 Data corresponding to different hidden layer neurons No.

Number of hidden layer neurons	32	64	128	256
Train score RMSE:	0.20	0.26	0.29	0.36
Test score: RMSE	0.19	0.23	0.26	0.33

Based on parameter adjustment and optimization, the basic parameters of the experimental model are selected as follows in Table 6 and Figure 8:

Table 6 Basic settings of model parameters

Basic settings of model parameters			
Hidden layers No.	2	Neurons No. at the input	5
Hidden layer neurons No.	32	Time step	20
Learning rate	0.05	Epochs	100
Optimizer selection	Adam	Dropout	0

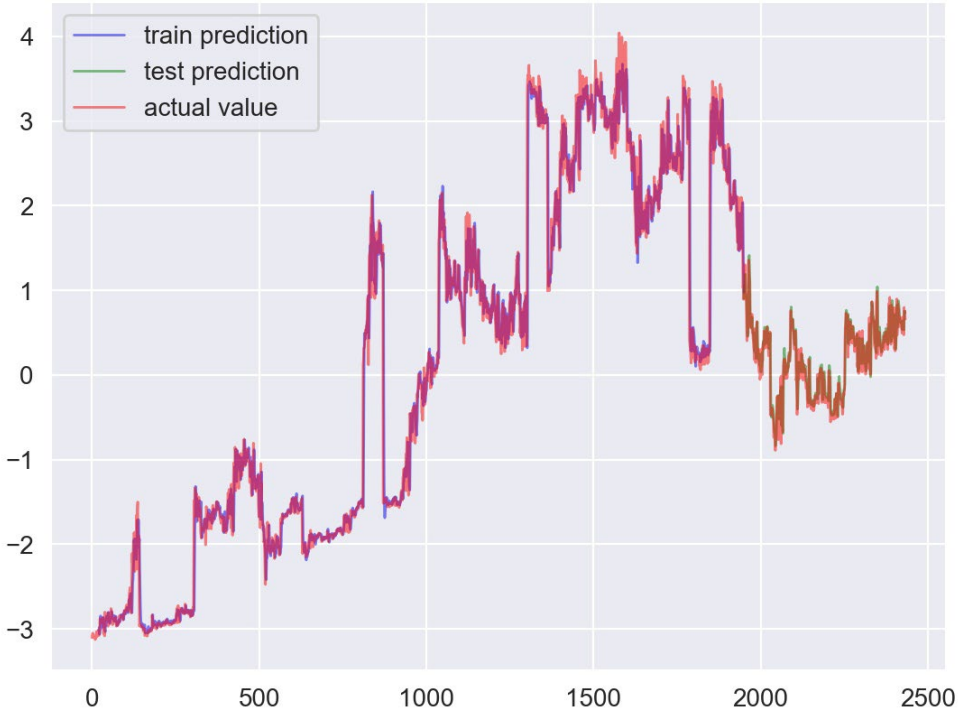


Figure 8 Parameter variation diagram

5. Conclusion

Stock price prediction is significant in the financial field, and an important reference for investors, traders, and market participants. With the advances in the information age, prediction methods based on stock information are also changing. Investors' investment concepts and ideas are gradually becoming more sophisticated. Therefore, a single index can no longer predict stock prices entirely. This article selects multiple features affecting stock prices as input variables to construct an LSTM neural network model. The objective of the study is to optimize the adjustment to stock price prediction through repeated experimentation. The hypothesis is that this will improve the accuracy of prediction.

This article selects indexes from multiple dimensions and normalizes the data to include as many factors as possible that affect stock prices as part of feature selection and preprocessing. Due to the large dimension of the input data, this article chose the feature extraction method of principal component analysis to extract features of the input data. The finally extracted sampling data was used as the input data for the subsequent model. As a result, the prediction error is smaller than before dimensionality reduction through experimental comparison. This article selects the LSTM neural

network model as the selection and construction of the prediction model. At the same time, in terms of LSTM parameter selection, comparisons are made in terms of time steps, number of hidden layers, hidden layer neurons, etc., to explore stock prediction effects of different model structures and parameter settings. Finally, the LSTM neural network prediction model structure in this experiment was determined.

It can be seen that the changing trend is the same, and the price is not much different based on the model verification compared to the experimental results after multiple adjustments and optimization with the original data.

References

[1] Chen Weihua. Research on the accuracy of stock market volatility prediction based on deep learning and stock forum data [J]. Management World, 2018, 34 (1): 180-181.

[2] Feng Yuxu, Li Yumei. Research on the CSI 300 index prediction model based on LSTM neural network [J]. Practice and Understanding of Mathematics, 2019, 49 (7): 308-315.

[3] Han Shanjie, Tan Shizhe. Design and implementation of deep learning model for stock prediction based on TensorFlow [J]. Computer Applications and Software, 2018, 35 (6): 267-291.

[4] Huang Qiuping, Zhou Xia, Gan Yujian, Wei Yu. Research on the application of SVM and neural network models in stock forecasting [J]. Microcomputers and Applications, 2015, 34 (5): 88-90.

[5] Hu Yue. Stock market timing model based on convolutional neural network - taking the Shanghai Composite Index as an example [J]. Financial Economics, 2018 (4): 71-74.

[6] Li Rendong, Rao Jiayi, Yan Yaning. Stock price prediction based on intelligent computing [J]. Science and Technology Bulletin, 2013, 29 (4): 152-154.

[7] Li Jie, Lin Yongfeng. Time series data prediction based on multi-time scale RNN [J]. Computer Applications and Software, 2018, 35 (7): 33-37.

[8] Li Zhenzhen, Wu Qun. Research on stock prediction algorithm based on LSTM neural network [J]. Fujian Computer, 2019, 35(7): 41-43.

[9] Lin Nan. Empirical analysis of Bank of China stock price forecast based on BP neural network and GARCH model [D]. Lanzhou: Lanzhou University, 2014.

[10] Lin Sheng. Research on stock forecasting based on LSTM [D]. Guangzhou: Guangzhou University, 2019.

[11] Hammad A A A, Ali S M A, Hall E L.Forecasting the Jordanian stock prices using artificial neural networks [M].Intelligent Engineering Systems through Artificial Neural Networks, 2007: 502-505.

[12] Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks [J]. science, 2006, 313 (5786) : 504-507.

[13] Huo J, Zheng Y, Chen X.Implementation of Transaction Trend Prediction Model Based on Regression Analysis [J].Journal of Baoshan Teachers'College, 2009, 117(1) : 19-23.

[14] KOLARIK T, RUDORFER G.Time series forecasting using neural networks [J].Aem Sigapl Apl Quote Quad, 1994, 25 (1) : 86-94.

[15] LIUF.WANG J.Fluctuation Prediction of stock marke index by Legendre neural network with random time strength function[J].Neuroc omputing, 2012, 83 (6) : 12-21.

[16] Ozbayoglu M.Neural based techical analysis in stock market forecasting[J].Intelligent Engineering Systems through Artificial Neural Networks, 2008, 18:261-265.

[17] Pang X, Zhou Y, Wang P, et al.An innovative neural network approach for stock market prediction[J]. Journal of Supercomputing, 2018 (1) : 1-21.